

Digital Design Professional

Handbuch für die Aus- und Weiterbildung zum
Digital Design Professional
auf Foundation Level
(DDP FL)

Kim Lauenroth, David Gilbert, Michael Kemper, Karsten Lehn,
Norbert Seyff, Melanie Stade und Marcus Trapp

Version 1.0.0
1. März 2022

*Übersetzung basiert auf der englischen Version des Handbuchs
„Education and Training Handbook for the Digital Design Professional at
Foundation Level“ V1.1.0, vom 01.03.2022*

Nutzungsbedingungen

Alle Inhalte dieses Dokuments, insbesondere Texte, Fotos, Grafiken, Diagramme, Tabellen, Definitionen und Vorlagen, sind urheberrechtlich geschützt. Alle (Co-)Autoren dieses Dokuments haben das ausschließliche Nutzungsrecht an den IREB e.V. übertragen.

Jede Nutzung dieses Dokuments oder seiner Bestandteile, insbesondere die Vervielfältigung, Verbreitung (Veröffentlichung), Übersetzung oder Reproduktion, bedarf der vorherigen Zustimmung des IREB e.V.

Jede Person ist berechtigt, die Inhalte dieses Dokuments im Rahmen der urheberrechtlich zulässigen Nutzungshandlungen zu nutzen, und nach den anerkannten Regeln der Wissenschaft korrekt zu zitieren.

Bildungseinrichtungen sind berechtigt, den Inhalt dieses Dokuments unter korrekter Bezugnahme auf das Werk für Lehrzwecke zu verwenden.

Eine Nutzung zu Werbezwecken ist nur mit vorheriger Zustimmung des IREB e.V. gestattet.

Danksagung

Der Inhalt dieses Handbuchs wurde von Michael Burmester, Kurt Schneider und Martin Glinz geprüft. Tracey Duffy hat die englische Version des Handbuchs überarbeitet.

Allen sei für ihr Engagement gedankt. Unser besonderer Dank gilt Bernd Aschauer, Nikola Eger, Thomas Geis, Saskia Hehl, Thomas Immich, Jens Kawelke, Rolf Molich, Knut Polkehn, Lars Sonnabend, Hans-Jörg Steffe, Stefan Tilkov, Marcus Winteroll und vielen anderen für ihre Unterstützung bei der Erstellung dieses Handbuchs.

Das Handbuch wurde am 8. März 2021 vom Council des IREB e.V. auf Empfehlung von Martin Glinz zur Veröffentlichung freigegeben.

Allen sei für ihr Engagement gedankt.

Umfang dieses Handbuchs

Dieses Handbuch bietet eine Einführung in Das Berufsbild Digital Design auf der Grundlage des Lehrplans für den Digital Design Professional (DDP) auf Foundation Level Niveau. Es ergänzt den Lehrplan und richtet sich an vier Lesergruppen:

- Lernende, die sich über Digital Design informieren und die Zertifizierungsprüfung ablegen möchten, können dieses Handbuch als Begleitbuch zu Schulungskursen nutzen, die von Schulungsanbietern angeboten werden, sowie zum Selbststudium und zur individuellen Vorbereitung auf die Zertifizierungsprüfung.
- Praktiker, die sich mit Digital Design vertraut machen wollen, können dieses Handbuch als Nachschlagewerk nutzen, um einen Prozess für die Erstellung digitaler Produkte aufzusetzen und geeignete Werkzeuge und Vorlagen für ihre tägliche Arbeit zu finden.
- Schulungsanbieter, die Ausbildungen für den DDP Foundation Level anbieten, können dieses Handbuch als Ergänzung zum Lehrplan für die Entwicklung ihrer Ausbildungsmaterialien oder als Studententext für die Teilnehmenden ihrer Schulungen verwenden.
- Experte in der Industrie, die bewährte Konzepte und Wissen in ihrer praktischen Arbeit anwenden wollen, finden in diesem Handbuch eine Fülle von nützlichen Informationen und Beispielen.

Dieses Handbuch stellt eine Verbindung zwischen dem Lehrplan, der die Lernziele auflistet und erläutert, und der einschlägigen Literatur über Digital Design her. Die Struktur des Handbuchs entspricht der Struktur des Lehrplans.

Sowohl die Autoren als auch der IREB e.V. haben viel Zeit und Mühe in die Erstellung, das Review und die Veröffentlichung dieses Handbuchs investiert. Wir wünschen Ihnen viel Erfolg beim Studium dieses Handbuchs. Wenn Sie Fehler entdecken oder Verbesserungsvorschläge haben, kontaktieren Sie uns bitte unter www.digitaldesign.org.

Sprache in der deutschen Übersetzung

Die Übersetzung dieses Lehrplan wurde mit Ziel einer geschlechtergerechten Sprache erarbeitet. Gleichzeitig wurde Wert darauf gelegt, die Lesbarkeit, eine möglichst kompakte und visuell ansprechende Darstellung und Effizienz des Textes zu erhalten.

Daher wird grundsätzlich auf die Verwendung von besonderen Zeichen (z.B. *) verzichtet. Stattdessen wird im Lehrplan ein kreativer Umgang mit geschlechtergerechter Sprache gepflegt, der allen Geschlechtern gerecht werden soll. Wichtige Fachbegriffe werden möglichst geschlechtsneutral gewählt (bspw. Benutzende anstatt Benutzer). Bei allgemeinen Aussagen im Text werden sowohl das generische Maskulinum als auch das generische Femininum verwendet.

Weiterhin verwendet die Übersetzung an ausgewählten Stellen englische Fachbegriffe. Diese englischen Begriffe werden immer dann verwendet, wenn der englische Begriff in der deutschen Alltagssprache regelmäßig verwendet wird.

Definitionen, die im DDP-Glossar enthalten sind werden ebenfalls ausschließlich auf Englisch dargestellt um den inhaltlichen Wortlaut nicht zu verzerren.

Versions-Historie

Version	Datum	Kommentar
1.0.0	01.03.2022	Erste Version des DDP-Handbuchs auf Deutsch

Geleitwort

Prof. Dr. Michael Burmester

Ein Paukenschlag. Es war das Jahr 2018. In Deutschland waren wir gerade aus der Startphase des vom Bundesministerium für Wirtschaft und Energie (BMWi) geförderten Mittelstand 4.0-Kompetenzzentrums Usability heraus. Unser Ziel war und ist es, kleinen und mittelständischen Unternehmen bei der Entwicklung und Auswahl von Software zu unterstützen. Software soll einfach nutzbar, also gute Usability aufweisen, aber auch positiv erlebbar sein und damit eine positive User Experience ermöglichen.

Gerade zurück von einer Veranstaltung erzählte mir eine Mitarbeiterin des Kompetenzzentrums völlig begeistert, dass sie von einer neuen Profession gehört habe, die sich Digital Design nennt, und auf menschenzentrierter Gestaltung aufbaut. Das Interessanteste wäre, dass die Digital Designer Verantwortung für die gesamte Gestaltung digitaler Technologien übernehmen und ein Handlungsmandat für die Leitung von Digitalprojekten haben sollen, so wie ein Architekt bei dem Entwurf und Bau von Gebäuden. Was für eine Forderung! Mir ging durch den Kopf, dass es doch dies war, was wir im Kreise von Usability und User Experience Professionals und Forschern öfters andiskutiert hatten, aber es eigentlich nie so auf den Punkt bringen konnten. Ein Handlungsmandat für menschenzentrierte Gestaltung von Technologie – unglaublich. Wir fanden heraus, dass der Branchenverband der Digitalindustrie Bitkom e.V. die Weiterentwicklung von Digital Design unterstützt und der Berufsverband der deutschen Usability und UX Professionals – der German UPA e.V. – bereits ein positives Positionspapier dazu erstellt hatte. Das steigerte unser Interesse weiter. Vor allem auch, weil Digital Design ein klares Bekenntnis zur menschenzentrierte Gestaltung formulierte.

Immerhin ist menschenzentrierte Gestaltung ein international anerkannter und in der Norm DIN EN ISO 9241 (2020) detailliert beschriebener Prozess. Es geht darum, dass Nutzende und weitere Stakeholder, also als Personen oder Organisationen, die ein Interesse an einer Software haben, systematisch am Softwaregestaltungsprozess beteiligt werden. Der Kontext, in dem ein digitales Produkt oder eine digitale Dienstleistung genutzt werden soll – mit all den involvierten Menschen, deren jeweiligen Bedarfen, Bedürfnissen und Aufgaben sowie deren jeweiligen Nutzungsumgebung – muss genau verstanden werden. Daraus leiten sich Gestaltungsideen und -konzepte ab, die als Prototypen oder reduzierte Produkte umgesetzt und anschließend wiederum mit den Nutzenden getestet und in Optimierungszyklen (Iterationen) verbessert werden. Dieses Vorgehen ermöglicht kontinuierliches und kreativitätsförderndes Lernen, wie eine attraktive Lösung gestaltet sein muss. Menschenzentrierte Gestaltung wird mittlerweile nicht nur in weiteren gestaltungsintensiven Bereichen eingesetzt, wie z.B. für Organisationsentwicklung und/oder Service-Design, sondern findet sich auch als Grundidee in Innovationsansätzen wie Lean Start-up oder Design Thinking wieder.

Digital Design legt nun die Verantwortung und das Handlungsmandat beim Entwurf von digitalen Produkten in die Hände des Designers. Dies wird mittlerweile sogar von weiteren Akteuren gefordert. So hat Donald Norman – ein zentraler Vordenker zu Usability, User Experience und menschenzentrierter Gestaltung – das 21st Century Design vorgeschlagen. Auch dieser Ansatz gibt dem Designer eine verantwortliche und koordinierende Funktion und setzt auf Partizipation der Menschen, die von einer Gestaltungslösung betroffen sind. Wie auch 21st Century Design will Digital Design dafür offen sein, was eine angemessene gestalterische Lösung ist. Ja, und diese muss nicht immer digital sein.

Im Rahmen der digitalen Transformation wird heftig darüber diskutiert, welche Auswirkungen der Einsatz digitaler Technologien haben könnte. Dabei werden sehr häufig Dystopien beschrieben: Arbeitsplätze werden vernichtet, die Arbeit wird verdichtet und überfordert die Arbeitenden, die Welt wird virtueller und das Zwischenmenschliche geht verloren. Sicher, digitale Transformation hat schon sehr viele Veränderungen herbeigeführt, die Licht und Schatten aufwies. Wichtig ist aber, dass digitale Lösungen nicht einfach geschehen, sondern bewusst konzipiert werden. Und das sollten sie auch. Digital Design widmet sich genau dieser Mission in umfassender Art und Weise. Im Design geht es ganz wesentlich auch um Haltung (oder neudeutsch das Mindset). Zum Digital Design Mindset gehören selbstverständlich ethische Fragestellungen, Fragen des Datenschutzes und der Datensicherheit, der Nachhaltigkeit von Lösungen etc. sowie selbstverständlich das klare Bekenntnis zur Menschzentrierung und die Perspektive, dass es bei der Konzeption digitaler oder digital erweiterter Lösungen um die Gestaltung – und so wird es auch im vorliegenden Handbuch ganz deutlich gemacht – einer wünschbaren Zukunft geht.

Zu der breiten Perspektive von Digital Design gehört, dass nicht nur die äußere Form eines Produktes gestaltet wird, sondern selbstverständlich auch, das was quasi „unter der Haube“ passiert, nämlich die digitale Technik. Während Nutzende an der Oberfläche Aussehen und Nutzung erleben, liegen darunter u.a. Funktionen und Algorithmen. Beides muss aufeinander abgestimmt sein und prägt so das Gesamterlebnis. Der umfassende Ansatz des Digital Design bezieht zudem explizit den Entwurf des Geschäftsmodells mit ein. Ob Kundinnen und Kunden ein Produkt kaufen möchten, Nutzende ein Produkt weiter nutzen und vielleicht sogar empfehlen möchten, hängt von dem Gesamtangebot mit seinen Kosten und Nutzen ab. Schließlich muss von einem digitalen Produkt auch ein Unternehmen leben können.

Das neue, unter dem Dach von IREB entstandene, Zertifizierungsprogramm für Digital Design Professionals (DDP) stellt einen Meilenstein im Hinblick auf die Etablierung von Digital Design als Profession dar. Das vorliegende Handbuch vermittelt das Basiswissen über Digital Design für die Zertifizierung zum DDP Foundation Level. Es ist – nun ja – umfassend, detailreich und anspruchsvoll, aber es ist auch sehr hilfreich und hochgradig systematisch gestaltet. Digital Design ist eine Profession, welche ein entsprechendes Studium seiner Grundlagen und Vorgehensweisen erfordert. Die Autorinnen und Autoren des Handbuchs ermöglichen dies in ganz hervorragender Form. Alle zentralen Konzepte werden klar definiert, notwendiges Vorwissen wird konsequent eingeführt, das Vorgehen im Digital Design Prozess wird an einem eingängigen Beispielprojekt ausführlich und anschaulich vorgestellt. Nicht nur die dafür notwendigen fachlichen Grundlagen und Methodiken, sondern auch übergreifendes Wissen, wie das Management beteiligter Personen, Human Factors und Geschäftsmodellentwicklung, werden erläutert. Das Handbuch ist ein hervorragender Startpunkt, um verantwortlich und verantwortungsvoll digitale und innovative Lösungen zu konzipieren.

Prof. Dr. Michael Burmester

Inhaltsverzeichnis

1	Eine Einführung in den Digital Design Professional	10
1.1	Digital Design als neues Berufsbild	11
1.1.1	Motivation für ein neues Berufsbild.....	11
1.1.2	Überblick über das Berufsbild des Digital Designers	12
1.2	Das Digitale als Material zum Bauen digitaler Lösungen begreifen	13
1.2.1	Die Idee des Digitalen als Material für den Aufbau digitaler Lösungen	14
1.2.2	Verstehen des Kontextes als Teil des Bauprozesses.....	17
1.2.3	Der Unterschied zwischen Auftraggeber-Rolle, Kunden-Rolle und Benutzer-Rolle.....	18
1.2.4	Die YPRC-Fallstudie als Beispiel für eine digitale Lösung.....	20
1.2.5	Ein Blick auf die Technologie in Bezug auf die Idee des Digitalen als Material	22
1.3	Einführung in den allgemeinen Bauprozess für digitale Lösungen	23
1.3.1	Querschnittliche Tätigkeitsbereiche.....	24
1.3.1.1	Management des Bauprozesses	24
1.3.1.2	Evaluierung der digitalen Lösung.....	24
1.3.2	Kerntätigkeitsbereiche des Bauprozesses.....	25
1.3.2.1	Design einer digitalen Lösung	25
1.3.2.2	Konstruktion einer digitalen Lösung.....	27
1.3.2.3	Realisierung einer digitalen Lösung.....	28
1.3.3	Den Bauprozess als parallelen Prozess begreifen	28
1.3.3.1	Zusammenarbeit zwischen Design und Konstruktion (1)	29
1.3.3.2	Zusammenarbeit zwischen Design und Realisierung (2)	31
1.3.3.3	Zusammenarbeit zwischen Konstruktion und Realisierung (3).....	33
1.3.3.4	Zusammenarbeit zwischen Design, Konstruktion, Realisierung und Management (4).....	34
1.3.4	Management des Bauprozesses auf Lösungs-, System- und Elementebene	36
1.3.4.1	Die Fähigkeit, den Bauprozess auf der Lösungs- und Systemebene zu verstehen	37
1.3.4.2	Der Unterschied zwischen einem konzeptorientierten und einem realisierungsorientierten Bauprozess	40
1.4	Kompetenzprofil eines Digital Design Professionals.....	41
1.4.1	Der Digital Design Professional als Ausbildungsprogramm.....	41
1.4.2	Zehn Prinzipien für gutes Digital Design.....	43
1.4.3	Digital Design Professional ist keine neue Rolle	45
2	Design-Kompetenz.....	47
2.1	Integration von Digital Design in den Bauprozess	47
2.1.1	Grundlagen des Designprozesses	48
2.1.1.1	Der Design-Squiggle als Designprozessmodell	48
2.1.1.2	Das Dual-Mode-Modell des Designs	50
2.1.2	Die drei wesentlichen Schritte des Bauprozesses für eine digitale Lösung	52
2.1.2.1	Der Scoping-Schritt (Auftragsklärung)	53
2.1.2.2	Der konzeptuelle Schritt	56
2.1.2.3	Der Entwicklungs- und Betriebsschritt	59
2.1.2.4	Zur Gleichwertigkeit von Scoping/Konzeptions- und Entwicklungsarbeit.....	62
2.1.3	Qualität als Querschnittsaspekt im Bauprozess	63
2.1.3.1	Ganzheitliche Betrachtung der Qualität im Bauprozess	63
2.1.3.2	Der Unterschied zwischen der Qualität einer digitalen Lösung und der Qualität eines digitalen Systems	64
2.1.3.3	Wahrnehmbare und zugrundeliegende Qualitätsattribute	65
2.1.3.4	Ein Qualitätsmodell für digitale Systeme.....	66

2.1.3.5	Beschreibung eines Qualitätsmodells für digitale Lösungen	67
2.1.4	Zusätzliche Ressourcen für den Bauprozess	68
2.1.5	Zusammenfassung: Ein idealisiertes Modell des Bauprozesses	68
2.2	Konzeptarbeit im Digital Design	72
2.2.1	Grundlagen der Konzeptarbeit	73
2.2.1.1	Konzepte sind Ideen im Denken oder in der Kommunikation	73
2.2.1.2	Nutzen und Grenzen von Konzepten	73
2.2.2	Pragmatische Dokumentvorlagen für die verschiedenen Abstraktionsebenen	75
2.2.2.1	Digital Design Brief	75
2.2.2.2	Lösungsdesignkonzept	77
2.2.2.3	Systemdesignkonzept	78
2.2.2.4	Softwaredesignkonzept	80
2.2.2.5	Gerätedesignkonzept	82
2.2.3	Dokumentationstechniken für die Lösungsebene	83
2.2.3.1	Pressemitteilung aus der Zukunft (Future Press Release)	84
2.2.3.2	Persona zur Charakterisierung von Kunden- und Benutzergruppen	84
2.2.3.3	Stakeholderliste	86
2.2.3.4	Value Proposition Canvas	86
2.2.3.5	Customer Journey Map	87
2.2.3.6	Business Model Canvas	89
2.2.4	Allgemeine Überlegungen zu Designkonzepten auf System- und Elementebene	90
2.2.4.1	Der Wert der Nicht-Redundanz und Instrumente zur Vermeidung von Redundanz	91
2.2.4.2	Modularität der Information	92
2.2.4.3	Beziehungen zwischen den Bausteinen	93
2.2.4.4	Lesbarkeit versus strukturierte Dokumentation	94
2.2.5	Dokumentationstechniken für die System- und Elementebene	94
2.2.5.1	Eine Anmerkung zur Idee der perfekten Technologie	94
2.2.5.2	Eine allgemeine Bausteinvorlage	95
2.2.5.3	Zielvorlage	95
2.2.5.4	Randbedingungs-vorlage	97
2.2.5.5	Form und Funktion auf der Systemebene	98
2.2.5.6	Form und Funktion auf der Elementebene	105
2.2.5.7	Qualität: Qualitätsanforderungsvorlage	116
2.2.5.8	Übersichtsbilder zum Zweck der Visualisierung	117
2.2.6	Alles für das große Ganze wieder zusammenfügen	118
2.2.6.1	Beziehungen zwischen Bausteinen auf Elementebene	119
2.2.6.2	Beziehungen zwischen Bausteinen auf Systemebene	122
2.2.6.3	Beziehung zwischen Elementebene und Systemebene	123
2.2.6.4	Beziehung zwischen Lösungsebene und System-/Elementebene	124
2.2.6.5	Beziehung zwischen Designkonzepten, Tätigkeitsbereichen und Schritten des Bauprozesses	126
2.2.7	Schlussfolgerung: Konzeptarbeit lernen	129
2.3	Anwendung von Prototypen im Digital Design	129
2.3.1	Definitionen für Prototypen	129
2.3.2	Ziele von Prototypen	131
2.3.2.1	Erkundung des Problems, der Benutzerbedürfnisse und der Anforderungen	131
2.3.2.2	Kommunikation von Lösungsideen und Konzepten	133
2.3.2.3	Testen und Verbessern von Konzepten und Lösungsvorschlägen	134
2.3.2.4	Empfehlung für eine Lösung oder eine Lösungsidee	134
2.3.3	Beispiele für die Verwendung von Prototypen in unterschiedlichen Disziplinen	135
2.3.4	Kriterien für die Kategorisierung von Prototypen	136
2.3.4.1	Interaktionsebene	136
2.3.4.2	Ziel des Prototypings	137
2.3.4.3	Detailgrad (Level of Fidelity)	138
2.3.4.4	Detailgrad für jede Dimension eines Prototyps (Mixed-Fidelity-Prototypen)	140

2.3.4.5	Fidelity-Profile von Prototypen, die typischerweise während des Bauprozesses verwendet werden ...	143
2.3.4.6	Prototyp-Kategorien in bestimmten Phasen des Bauprozesses	147
2.3.4.7	Grad der Immersion in Bezug auf das Fidelity-Profil von Prototypen	148
2.3.5	Werkzeuge zur Erstellung von Prototypen	150
2.3.5.1	Software-Design- und Entwicklungswerkzeuge und Technologien für die Erstellung von Prototypen..	150
2.3.5.2	Industriedesign-Werkzeuge für die Erstellung von Prototypen	151
2.3.5.3	Interaktionsdesign-Werkzeuge für die Erstellung von Prototypen	153
2.3.5.4	Andere Werkzeuge.....	154
2.3.6	Erstellung und Verwendung einfacher Low-Fidelity-Prototypen	154
2.3.7	Fazit zum Prototyping.....	158
3	Digitales Material.....	160
3.1	Technologie verstehen.....	160
3.2	Wahrnehmbare Technologie.....	165
3.2.1	Endgeräte.....	165
3.2.2	Interaktionstechnologie.....	166
3.2.3	Software-User-Interface-Technologie.....	175
3.3	Zugrundeliegende Technologie	179
3.3.1	Technologien zur Programmierung	180
3.3.2	Technologien zum Betreiben von Software	182
3.3.3	Technologien für die Digitale Kommunikation	184
3.4	Technologie-orientierte Wissensgebiete	186
3.4.1	Software-Architektur.....	186
3.4.2	Berechnungskomplexität	187
3.4.3	Mensch-Computer-Interaktion	188
3.5	Die Digital Design Perspektive auf Technologie	189
4	Querschnittskompetenzen.....	191
4.1	Human Factors.....	191
4.1.1	Grundlagen der menschlichen Wahrnehmung und Aufmerksamkeit.....	191
4.1.1.1	Visuelle Aufmerksamkeit	192
4.1.1.2	Auditive Aufmerksamkeit	193
4.1.1.3	Lenkung der Aufmerksamkeit.....	194
4.1.2	Grundlagen der menschlichen Leistungsfähigkeit	194
4.1.3	Emotionen in der Mensch-System-Interaktion.....	196
4.1.4	Die Rolle von Prototypen.....	197
4.2	Geschäftsmodelle für digitale Lösungen.....	198
4.2.1	Geschäftsmodellmuster.....	200
4.2.2	Digitale Geschäftsmodelle.....	201
4.2.3	Nachdenken über zukünftige Digital Business Potenziale.....	202
4.3	People Management	204
4.3.1	Den Bauprozess als sozialen Prozess begreifen	205
4.3.2	Menschen durch Persönlichkeitsmodelle verstehen.....	207
4.3.2.1	Der innere Ring: Abstrakt versus konkret (Verstehen, Wahrnehmen und Lernen)	211
4.3.2.2	Der zweite Ring: Kooperativ versus pragmatisch (Aktionsschwerpunkt)	212
4.3.2.3	Der dritte Ring: Direktiv versus informativ (Kommunikationsstil)	212
4.3.2.4	Der vierte Ring: Ausdrucksstark versus reflektiert (Energie, Impulsivität).....	213

4.3.2.5	Schlussfolgerung zur Arbeit mit Persönlichkeitsmodellen	213
4.3.3	Der Bauprozess aus einer gruppendynamischen Perspektive	214
4.3.3.1	Steuerung der Wahrnehmungen und des Lernpotenzials während des Bauprozesses	215
4.3.3.2	Zukunftsvorstellungen führen	216
4.3.3.3	Management von Arbeitsstil, Schlüsselkompetenzen und Rollenzuweisung im Bauprozess.....	218
4.3.3.4	Führung während des Bauprozesses steuern.....	222
4.3.3.5	Berücksichtigung des Arbeitsumfelds und Verständnis des Kontextes.....	223
4.3.4	Fazit.....	224
5	Ein Bauprozess für Einsteiger.....	226
5.1	Leitlinien für den Scoping-Schritt	226
5.1.1	Scoping eines Wicked Problems	227
5.1.1.1	Phase 1: Zusammenstellung eines Design-Thinking-Teams.....	228
5.1.1.2	Phase 2: Durchführung des Design-Thinking-Prozesses	228
5.1.1.3	Phase 3: Dokumentieren der Ergebnisse und ggf. Iteration	229
5.1.2	Scoping eines Tame Problems.....	230
5.1.2.1	Phase 1: Befragung wichtiger Stakeholder	230
5.1.2.2	Phase 2: Scoping-Workshop	231
5.1.2.3	Phase 3: Dokumentieren Sie die Ergebnisse und iterieren Sie, falls erforderlich.....	232
5.1.3	Definition der allgemeinen Bedingungen für den Bau einer digitalen Lösung.....	232
5.1.3.1	Leitlinien für die Festlegung des Zeitplans für den Aufbau einer digitalen Lösung.....	232
5.1.3.2	Leitlinien für die Festlegung der Art der Zusammenarbeit für den Aufbau einer digitalen Lösung.....	233
5.1.3.3	Leitlinien für die Festlegung des Budgets für den Aufbau einer digitalen Lösung	234
5.1.3.4	Leitlinien für die Definition potenzieller Einnahmequellen für den Aufbau einer digitalen Lösung	234
5.1.3.5	Leitlinien für die Festlegung der verfügbaren Ressourcen für den Aufbau einer digitalen Lösung	235
5.2	Leitlinien für den konzeptuellen Schritt	235
5.2.1	Phase 1: Erkundung des Lösungsraums für die digitale Lösung aus der Kundenperspektive	236
5.2.2	Phase 2: Erarbeitung und Bewertung von Lösungskandidaten aus wirtschaftlicher Sicht.....	240
5.2.3	Phase 3: Annäherung an einen vielversprechenden Lösungskandidaten aus der Perspektive der Machbarkeit	244
5.2.4	Phase 4: Abschließende Evaluierung des Lösungskandidaten mit dem Auftraggebenden	244
5.3	Leitlinien für den Entwicklungs- und Betriebsschritt.....	245
5.3.1	Überblick über die Struktur des Prozesses.....	246
5.3.2	Management der Arbeit auf allen Abstraktionsebenen	248
5.3.2.1	Management der Arbeit auf der Elementebene	249
5.3.2.2	Verwaltung der Arbeit auf der Systemebene.....	251
5.3.2.3	Management der Arbeit auf der Lösungsebene	255
5.3.2.4	Das Gesamtbild zum Management von Arbeit auf allen Abstraktionsebenen.....	256
5.3.3	Leitlinien für die Definition von Arbeitsaufträgen im Bauprozess	257
5.3.3.1	Überblick über die Beziehungen zwischen Arbeitsprodukten und Designkonzepten	258
5.3.3.2	Allgemeine Vorlage für Arbeitsaufträge	259
5.3.3.3	Lösungs-Workitems.....	261
5.3.3.4	Epics in Bezug auf Lösungs- und Systemdesignkonzept	261
5.3.3.5	User Storys in Bezug auf Elementdesignkonzepte	264
5.3.3.6	Konzept-Workitems für die Arbeit an Designkonzepten	268
5.3.3.7	Prototyp-Workitem.....	270
5.3.3.8	Evaluations-Workitems.....	271
5.3.3.9	Technische Workitems	273
5.3.3.10	Niederschreiben von Defekten	274
5.3.4	Phase 1: Erste Release-Planung und Backlog-Vorbereitung	275
5.3.4.1	Erarbeitung eines Element-Design-Canvases für jedes Element	276
5.3.4.2	Erstellen einer Story Map zur Definition von User Storys.....	279
5.3.4.3	Definieren von Arbeitsaufträgen sowie Schätzen, Priorisieren und Verwalten des System-Backlogs..	279

5.3.5	Phase 2: Entwicklung des ersten Releases der digitalen Lösung	282
5.3.6	Phase 3: Weitere Entwicklung während des Betriebs	283
5.3.7	Phase 4: Außerbetriebnahme.....	284
5.4	Lean Startup als alternativer Ansatz	285
5.5	Schlussfolgerungen zum Bauprozess für Einsteiger	286
6	Gutes Digital Design verwirklichen.....	287
6.1	Beiträge des Digital Design Professional.....	287
6.2	Die Bedeutung von praktischer Erfahrung und Heuristiken.....	290
6.3	Die Wichtigkeit von Teamarbeit	290
I.	Literaturverzeichnis.....	292
II.	Abbildungsverzeichnis.....	298
III.	Tabellenverzeichnis.....	301

1 Eine Einführung in den Digital Design Professional

In diesem Kapitel wird der Digital Design Professional (DDP) als neues Ausbildungsprogramm für die Gestaltung digitaler Lösungen vorgestellt. In diesem Handbuch verwenden wir die Abkürzung DDP. Die Einführung beginnt mit dem Abschnitt 0, in dem die Bedeutung eines neuen Berufsbildes namens Digital Design erläutert wird, um den künftigen Ansprüchen und Herausforderungen in einer digitalen Gesellschaft gerecht zu werden. Der Kerngedanke von Digital Design wird in Abschnitt 1.2 vorgestellt: Das Digitale als Material zu verstehen, das zum Aufbau digitaler Lösungen verwendet werden kann.

Mit dem Verständnis des Digitalen als Material wird in Abschnitt 1.3 der Bauprozess für eine digitale Lösung vorgestellt. Das Verständnis dieses Bauprozesses ist notwendig, um den DDP-Umfang zu verstehen. Abschnitt 1.4 schließt die Einführung ab und stellt das Kompetenzprofil eines DDP bezogen auf den Bauprozess einer digitalen Lösung vor.

Eine umfassende Fallstudie, die den Bauprozess für eine fiktive digitale Lösung beschreibt, wird mit diesem Handbuch geliefert und dient als Grundlage für Beispiele. Die Lösung wird YPRC (Your Personal Running Coach) genannt. YPRC bietet einen ganzheitlichen Trainingsservice für Laufeinsteiger und besteht aus einer speziellen Smartwatch, einer Smartphone-App und einem Serviceportal. Ausführliche Angaben zur Fallstudie finden Sie in den ergänzenden Materialien zum Handbuch.

Weitere Einzelheiten zu der Fallstudie ziehen sich durch das gesamte Handbuch. Es wird nicht davon ausgegangen, dass die Fallstudie im Voraus bekannt ist. Es ist sogar besser, die Fallstudie nicht im Voraus zu lesen, da die Details der Fallstudie im Laufe des Handbuchs ausgearbeitet werden.

Um den Leser mit diesem Handbuch zu unterstützen, verwenden wir die folgende Formatierung:

Definition eines wichtigen Begriffs

Beispiel aus der YPRC-Fallstudie

Darüber hinaus soll die Überschrift **Praxistipp** betonen, dass der Text unter dieser Überschrift praktische Ratschläge für die tägliche Arbeit enthält und über den Inhalt des Zertifikats hinausgeht.

1.1 Digital Design als neues Berufsbild

1.1.1 Motivation für ein neues Berufsbild

Die Entwicklung der digitalen Technologie verändert die Art der digitalen Lösungen und kann durch die folgenden Ebenen charakterisiert werden¹:

- *Digitization* (Elektrifizierung) ist der Einsatz digitaler Technologie, um mit digitalen Daten Probleme zu lösen, die zuvor mit nicht-digitalen Mitteln gelöst wurden.
- *Digitalisierung* ist der Einsatz digitaler Technologie, um Lösungen und Geschäftsprozesse zu erschaffen, die nur mit digitalen Mitteln realisierbar sind.
- *Digitale Transformation* findet statt, wenn digitale Lösungen Einfluss auf Menschen und Gesellschaft haben, indem sie die Gewohnheiten und das Leben der Menschen mit digitalen Mitteln verändern.

Die Anfänge der digitalen Datenverarbeitung standen ganz im Zeichen der Digitalisierung. Als die Computerhardware immer leistungsfähiger, billiger und weniger voluminös wurde, entdeckten Unternehmen und Ingenieure, dass die digitale Technologie Lösungen ermöglichte, die zuvor nicht realisierbar gewesen waren: Digitalisierungslösungen kamen auf. Seitdem hat die Digitalisierung mehrere Wellen der digitalen Transformation ermöglicht. Heute gibt es eine Koexistenz von Digitization, Digitalisierung und digitaler Transformation.

So wurden beispielsweise analoge Verzeichnisse wie Telefonbücher zunächst als elektronische Verzeichnisse digitalisiert. Diese boten die gleichen Suchfunktionen wie analoge Verzeichnisse, waren aber billiger, schneller und einfacher zu verteilen. Dies ermöglichte die Entwicklung neuer Funktionen - wie z. B. die Suche nach bestimmten Kriterien oder die Rückwärtssuche -, die mit analogen Verzeichnissen nicht durchführbar waren: Die Digitalisierung fand statt. Das Aufkommen des World Wide Web und effizienter Suchtechnologien hat die Gewohnheiten der Menschen von der Suche nach Informationen in speziellen Verzeichnissen hin zur On-Demand-Suche verändert - dies ist ein frühes Beispiel für den digitalen Wandel.

Beim Bau einer Lösung mit digitaler Technologie besteht die erste Herausforderung darin, herauszufinden, was gebaut werden soll. Wenn es darum geht, herauszufinden, was gebaut werden soll, ist „*Stakeholder*“ ein wichtiger Oberbegriff für alle Arten von Personen oder Organisationen, die am Bauprozess beteiligt sind. Der Begriff ist wie folgt definiert [Glin2020]:

Stakeholder: A person or organization who influences a system's requirements or who is impacted by that system.

Eine wesentliche Stakeholder-Rolle ist die Auftraggeber-Rolle (Client):

Client: A person or organization who orders a system or a solution to be built.

Benutzer- und Kunden-Rolle als weitere wesentliche Stakeholder-Rollen werden in 1.2 definiert. Die Rolle des Umsetzungsteam-Mitglieds als Stakeholder wird in Abschnitt 2.1 vorgestellt.

Für die Begriffe Digitalisierung und digitale Transformation gibt es mehrere Definitionen (vgl. [Bloo2018]). Wir bevorzugen die Definition, bei der diese Begriffe als voneinander abhängige Ebenen verstanden werden.

In Projekten, in denen es Auftraggeber gibt, die wissen, was sie bestellen wollen, und Stakeholder, die ihre Bedürfnisse kennen und wissen, welche Probleme eine digitale Lösung lösen soll, wird die Herausforderung, was gebaut werden soll, durch *Requirements Engineering* angegangen: Identifizierung der richtigen Stakeholder, Erhebung der Anforderungen von den Stakeholdern, sowie Konsolidierung, Dokumentation, Validierung und Verwaltung dieser Anforderungen. Ausgehend von den Anforderungen ist eine systematische und effiziente Umsetzung einer digitalen Lösung möglich. Solche Ansätze zum Aufbau einer digitalen Lösung werden als *anforderungsgetriebene* Ansätze bezeichnet.

In der Produktentwicklung gibt es jedoch oft weder Personen in einer Auftraggeber-Rolle, die eine eindeutige Lösung in Auftrag geben, noch leicht verfügbare Stakeholder, die ihre Bedürfnisse und Probleme kennen. Eine ähnliche Situation entsteht, wenn eine Person in einer Auftraggeber-Rolle eine innovative Lösung fordert, aber niemand eine klare Vorstellung davon hat, was eine solche Lösung leisten und wie sie aussehen soll. In diesen Situationen muss die Frage, was gebaut werden soll, auf andere Weise gelöst werden. Die zu entwickelnden digitalen Lösungen werden durch digitale Technologie angetrieben; das Ziel solcher Lösungen ist ein digitalisiertes Unternehmen, das die digitale Transformation vorantreibt [Kell2016]. Diese Situationen erfordern einen *gestaltungsorientierten* Ansatz für den Bau einer digitalen Lösung: *Designer* erkunden, was getan werden könnte, entwerfen Visionen für digitale Lösungen und gestalten schließlich die Form, Funktion und Qualität der digitalen Lösung. Da diese Art der kreativen Gestaltung analog zur Tätigkeit im Industriedesign ist, wo physische Produkte gestaltet werden, wird die Gestaltung von digitalen Lösungen als *Digital Design* bezeichnet.

1.1.2 Überblick über das Berufsbild des Digital Designers

Aufgrund des innovativen Charakters der Digitalisierung und der digitalen Transformation muss eine anforderungsgetriebene Entwicklung durch eine gestaltungsorientierte Entwicklung digitaler Lösungen ergänzt werden, die die Nutzung neuer technischer Möglichkeiten umfasst. Digital Design [Bitk2017] ist ein Berufsbild, das eine Verlagerung des Schwerpunkts in diese Richtung darstellt. Des Weiteren ist Digital Design ein Berufsbild, das darauf abzielt, die heutzutage notwendigen Fähigkeiten zur Gestaltung und zum Bau digitaler Lösungen klar zu formulieren und zu stärken [LBGH2018]. Digital Design ist wie folgt definiert:

Digital Design: The creative design of digital solutions.

Digital Design versteht das Digitale als gestaltbares Material (siehe Abschnitt 1.2). Dieses Verständnis geht über ein rein technisches Verständnis der digitalen Technologie hinaus und zielt auf eine Kombination von gestalterischen und technischen Fähigkeiten ab, ähnlich einem Verständnis, das im Industriedesign und der Bauarchitektur gelebt wird.

Digital Design bedeutet das kreative Design im Sinne des Entwurfs und der Gestaltung digitaler Lösungen durch eine ganzheitliche Betrachtung der technischen Möglichkeiten des digitalen Materials, der wirtschaftlichen Aspekte und der aktuellen oder zukünftigen Bedürfnisse der Menschen.

Der Begriff Design hat mehrere Bedeutungen und kann sowohl für Design als Tätigkeit als auch für Design als Ergebnis verwendet werden. Digital Design, mit der Idee Digital als gestaltbares Material zu verstehen, begreift Design in all seinen Facetten, d.h. sowohl gestalterisch als auch technisch, künstlerisch, kreativ und explorativ. Auch wenn Design im Deutschen in Gestaltung und Entwurf unterschieden werden kann, verzichten wir in diesem Handbuch auf diese

Unterscheidung. Für eine gute Balance zwischen Lesbarkeit und präziser Formulierung unterscheiden wir in diesem Handbuch zwischen Design als Tätigkeit und Design als Ergebnis. Wenn wir von Design als Tätigkeit sprechen, dann verwenden wir den Begriff Design als Substantiv und gestalten als Verb. Wenn wir von den Ergebnissen der Design-Tätigkeit sprechen, verwenden wir entweder den Begriff Design als Oberbegriff für diese Ergebnisse oder ihre speziellen Bezeichnungen (z. B. Prototyp, Designkonzept oder visuelles Design), sofern die Bedeutung aus dem Kontext nicht klar ersichtlich ist.

Digital Design gestaltet neue und optimiert bestehende digitale Lösungen, dies bedeutet:

- Ziele, Nutzen und Mittel einer digitalen Lösung gemeinsam gestalten – dies spiegelt die ganzheitliche Sicht auf die Lösung und das System (siehe Abschnitt 1.3.3) und die Fähigkeit zur Zusammenarbeit mit allen anderen Tätigkeitsbereichen wider
- Sowohl die kleinen als auch die großen Aspekte gestalten, wobei sich die großen Aspekte auf die Sichtweise einer digitalen Lösung auf Lösungs- und Systemebene und die kleinen Aspekte auf das Design der Elemente einer digitalen Lösung beziehen (siehe Abschnitt 1.2.1)
- Wahrnehmbare und zugrundeliegende Aspekte einer digitalen Lösung gemeinsam gestalten – dies bezieht sich auf die Tatsache, dass das Design der wahrnehmbaren Form, Funktion und Qualität einer digitalen Lösung ein tiefes Verständnis der zugrundeliegenden Form, Funktion und Qualität erfordert, die die wahrnehmbaren Aspekte ermöglichen (siehe Abschnitt 1.2.1)
- Materielle und immaterielle Aspekte einer digitalen Lösung gestalten – dies bezieht sich auf die Tatsache, dass eine digitale Lösung oft nicht nur aus Software, sondern auch aus physischen Teilen besteht (siehe Abschnitt 1.2.1)

Digital Design bedeutet die Verantwortung für das Design (im Sinne der eigentlichen Tätigkeit und des Ergebnisses) einer digitalen Lösung zu übernehmen und den Bauprozess einer digitalen Lösung aus der Perspektive des Designs zu leiten. Dazu gehört die Mitgestaltung und die Optimierung des Designs als Teil des Bauprozesses ebenso wie die intensive Zusammenarbeit mit allen anderen Tätigkeitsbereichen des Bauprozesses (1.3).

Nachfolgend werden wichtige grundlegende Informationen zum Verständnis von Digital Design als Berufsbild gegeben: Das Verständnis des Digitalen als Material (Abschnitt 1.2) und der allgemeine Bauprozess einer digitalen Lösung (Abschnitt 1.3).

Zum Abschluss dieses Abschnitts stellen wir DDP vor, das Ausbildungsprogramm, das wir entwickelt haben, um Teil der Digital Design Profession zu werden (Abschnitt 1.4).

1.2 Das Digitale als Material zum Bauen digitaler Lösungen begreifen

Bevor die verschiedenen Begriffe definiert werden, soll die unter Tabelle 1 vorgestellte Analogie zur Gebäudearchitektur als Überblick dienen und die Gesamtidee des digitalen Materials erläutern.

Das Wort *digital* wird oft als technisches Adjektiv verwendet, um die Darstellung von Daten in einem binären Format zu bezeichnen. Wie in Abschnitt 0 dargestellt, bilden binäre Daten die Grundlage zur Gestaltung neuer Geschäftsmodelle, sozialer Netzwerke sowie innovativer Produkte und Dienstleistungen.

Das bedeutet, dass die Tragweite von Binärdaten weit über den Transport und die Umwandlung von Informationen hinausgeht: Wir nutzen Binärdaten zur Gestaltung von Lebensprozessen in demselben Sinne, wie Gebäudearchitekten Lebensprozesse durch die Gestaltung von Räumen gestalten. Daher kann das Digitale auch als Substantiv betrachtet werden - etwas, das sich in der Idee des Digitalen als Material widerspiegelt.

Tabelle 1 - Terminologie in der Gebäudearchitektur und im Digital Design

Referenzpunkt	Berufsbild	
	Gebäudearchitektur	Digital Design
Gestaltungsgegenstand ²	Direkt: Raum Indirekt: Lebensprozesse	Direkt: Fluss von Binärdaten Indirekt: Lebensprozesse
Material	Baumaterial (Beton, Stahl, Holz, Fenster, Türen)	Digitales Material (Software, Hardware, Algorithmen)
Abstraktes Ergebnis	Gebäude	Digitales System
Spezifisches Ergebnis in einem konkreten Kontext mit einem definierten Ziel	Wohnhaus, Bürogebäude, Klinik, Gartenhaus	Fitness-Tracking-App, Online-Shop für Bücher, Unternehmenssoftware, soziales Netzwerk

1.2.1 Die Idee des Digitalen als Material für den Aufbau digitaler Lösungen

Die Idee, das Digitale als Material zu verstehen, soll die Bedeutung des Digitalen für unsere Wirtschaft und Gesellschaft zum Ausdruck bringen. Sie soll auch verdeutlichen, dass das Digitale genauso wie andere Materialien zu innovativen digitalen Lösungen geformt werden kann. Die Kernbegriffe dieses Konzepts werden nachfolgend definiert.

Auf den ersten Blick erscheinen diese Definitionen theoretisch und abstrakt. Wir werden diese Konzepte anschließend anhand der YPRC-Fallstudie erläutern.

Wir definieren Digital wie folgt:

Digital (noun): The structure, flow, and transformation of binary data.

Dieses Verständnis soll verdeutlichen, dass Digitales genauso wie andere Materialien gestaltet werden kann, um digitale Lösungen zu erschaffen. Dieses Verständnis ist die Grundlage für die Schwerpunktverlagerung weg von einer reaktiven technischen und anforderungsorientierten Entwicklung digitaler Lösungen hin zu einer proaktiven designorientierten Entwicklung digitaler Lösungen.

Eine Voraussetzung für das Verständnis von digitalem Material ist die Definition des Begriffs *System*, der wie folgt definiert ist.

**System: In general: A principle for ordering and structuring.
In engineering: A coherent, delimitable set of elements that—by coordinated action—achieve some purpose.**

Nach Walter Gropius [Grop1930] bedeutet Architektur die Gestaltung von Lebensvorgängen („Bauen bedeutet Gestaltung von Lebensvorgängen“). Wir glauben, dass dies auch für das digitale Design gilt.

Die folgenden drei Begriffe sind nützlich, um über Systeme zu kommunizieren:

- *Form*: Die Elemente und die Beziehungen zwischen den Elementen, die die Systemstruktur bilden
- *Funktion*: Die Fähigkeiten eines Elementes, einer Kombination von Elementen oder die Fähigkeiten des Systems als Ganzes
- *Qualität*: Der Grad, in dem ein Element, eine Beziehung zwischen Elementen, oder eine Fähigkeit eines Systems, definierte Qualitätsmerkmale erfüllt

Um gute Qualität zu erreichen, über Qualität zu kommunizieren und Qualität zu bewerten, bedarf es explizit definierter Qualitätsmerkmale (vgl. [ErMa2008]). Qualitätsmerkmale können auf verschiedene Weise definiert werden. In Abschnitt 2.1.3 stellen wir beispielhafte Ansätze zur Definition von Qualitätsmerkmalen vor.

Wie jede andere Art von Daten, benötigen auch binäre Daten ein Medium, um sie zu Tragen. Ein klassisches Trägermedium für geschriebenen Text ist beispielsweise Papier. Für binäre Daten gibt es zahlreiche Trägermedien, für die wir den Oberbegriff digitales Material verwenden und diesen wie folgt definieren:

Digital material: The technological means that enable the digital, that is, the structure, flow, and transformation of binary data.

Um digitales Material zu verstehen, ist es wichtig, die vier wichtigsten Eigenschaften von digitalem Material zu kennen:

1. Digitales Material hat kein Ziel.
2. Digitales Material hat eine zugrundeliegende und eine wahrnehmbare Ebene.
3. Digitales Material hat technologieneutrale Aspekte.
4. Digitales Material kann in Grenzen auch ohne Programmierkenntnisse gestaltet werden.

Das digitale Material ermöglicht jedoch nur den Fluss und die Umwandlung von Daten. Ein Datenfluss existiert nicht ohne ein System, das den Datenfluss verarbeitet, transportiert und speichert. Bedeutung und Wert dieser Daten (d. h. Informationen) entstehen nur, wenn der Datenfluss zwischen Benutzern und einem System stattfindet, das den Datenfluss produzieren, transportieren und konsumieren kann.

Ein solches System wird als digitales System bezeichnet:

Digital system: A technical system that realizes a digital solution in a given context with digital means, that is, by processing, transporting, and storing binary data.

Die Benutzer-Rolle (User) ist eine wichtige Stakeholder-Rolle für ein digitales System und wird wie folgt definiert:

User: A person who uses the functionality provided by a system.

Neben menschlichen Benutzern können digitale Systeme auch von Tieren genutzt werden (z. B. in der digitalisierten Landwirtschaft).

Elemente eines Systems können ebenso als (Sub-) Systeme verstanden werden. Dies ermöglicht die Definition von Systemen, die aus einer mehrstufigen Hierarchie von Systemen bestehen.

Mit dem Begriff des digitalen Systems wird eine digitale Lösung wie folgt definiert:

Digital solution: A socio-technical system that solves a real-world problem with digital means.

Das Verständnis, dass ein soziotechnisches System ein System ist, das Software, Hardware, Menschen und organisatorische Aspekte umfasst, folgt dem Verständnis von Systemen aus der allgemeinen Systemtheorie. Das bedeutet, dass es beim Digital Design um die Gestaltung von technischen (digitalen) Systemen und um die Gestaltung von soziotechnischen Systemen (der digitalen Lösung) mit digitalem Material geht.

Das Problem, das eine digitale Lösung löst, kann auch als Ziel oder Wertversprechen bezeichnet werden, das einem Kunden in einem bestimmten Kontext angeboten wird (siehe Abschnitt 2.1.2). Die Kunden-Rolle (Customer) ist eine weitere wesentliche Stakeholder-Rolle für die digitale Lösung und wird wie folgt definiert:

Customer: A person or organization who receives a system, a product, or a service.

Der Begriff *Empfangen* (receive) umfasst sowohl den Kauf einer Lösung als auch den kostenlosen Erhalt einer Lösung. Die Definition ist sehr weit gefasst, um verschiedene Situationen und Geschäftsmodelle abzudecken. Typische Situationen sind:

- Die Kunden-Rolle kann das digitale System ohne weitere Dienstleistungen erhalten. Zum Beispiel: Kauf einer Bürosoftware.
- Die Kunden-Rolle kann ein Produkt erhalten, das in die digitale Lösung eingebettet ist. Zum Beispiel: Kauf einer Spielekonsole, die den Kauf von Spielen über das Internet ermöglicht.
- Die Kunden-Rolle kann einen Service erhalten, den die digitale Lösung bietet. Zum Beispiel kann ein Kunde die digitale Lösung nutzen, um ein Hotelzimmer zu buchen.

Neben den eigentlichen Kunden kann eine digitale Lösung auch indirekte Kunden haben. Dies ist zum Beispiel der Fall, wenn Kunden eine digitale Lösung einsetzen, um nicht-digitale Dienstleistungen zu verbessern, die sie ihren Kunden anbieten (z. B. eine Arztpraxis, die ihren Patienten einen Online-Terminbuchungsdienst anbietet).

Die zwei Ebenen des digitalen Materials manifestieren sich in zwei Ebenen einer digitalen Lösung:

- Wahrnehmbare Ebene: Form, Funktion und Qualität, die von Stakeholdern wahrgenommen werden kann
- Zugrundeliegende Ebene: Form, Funktion und Qualität, die der Wahrnehmung von Stakeholdern verborgen ist und die die wahrnehmbare Ebene ermöglicht

1.2.2 Verstehen des Kontextes als Teil des Bauprozesses

Im Allgemeinen ist ein Kontext ein Netzwerk von Gedanken und Bedeutungen, das zum Verstehen von Phänomenen oder Äußerungen benötigt wird. Im Digital Design ist der Kontext wie folgt definiert:

Context: The part of the environment of a digital solution or digital system that is relevant for understanding and building a digital solution.

Zum Kontext gehören alle wesentlichen Stakeholder und vor allem potenzielle Kunden und Benutzer der digitalen Lösung.

Problem und Kontext sind untrennbar

Ein gelöstes Problem und sein Kontext sind untrennbar miteinander verbunden. Das bedeutet, dass eine digitale Lösung, die in einem Kontext funktioniert, nicht unbedingt auch in einem anderen Kontext funktionieren muss.

Der wichtige Unterschied zwischen einer digitalen Lösung und einem digitalen System ist, dass das digitale System das technische Mittel darstellt, um einen Zweck in einem definierten Kontext zu erreichen (die digitale Lösung). Aus einer theoretischen Perspektive sind zwei Aspekte wichtig.

Erstens kann die Beziehung zwischen Mittel und Zweck kompliziert oder komplex sein (vgl. [Snow2005]): Eine komplizierte Beziehung ist durch eine deterministische Ursache-Wirkung-Beziehung zwischen Mittel und Zweck gekennzeichnet. Eine komplexe Beziehung hat einen nicht-deterministischen Anteil, der es schwierig oder sogar unmöglich macht, sie im Voraus zu analysieren.

Menschen ohne richtiges Verständnis für digitales Material empfinden digitale Lösungen oft als komplex. Mit dem richtigen Verständnis für digitales Material und einer Ausbildung in Digital Design ist es jedoch möglich, die komplizierten von den komplexen Mittel-Zweck-Beziehungen zu trennen und mit beiden angemessen umzugehen.

Zweitens sind Mittel und Zweck unabhängig voneinander: Ein Zweck kann mit verschiedenen Mitteln erreicht werden und ein Mittel kann verwendet werden, um verschiedene Zwecke zu erreichen. Dies führt oft zu dem Eindruck, dass der Zweck (die digitale Lösung) klar definiert werden sollte, bevor über die Mittel (das digitale System) nachgedacht wird.

In der Praxis beeinflussen sich Mittel und Zweck jedoch erheblich gegenseitig. Natürlich ist es wichtig, mit dem Verständnis des Zwecks zu beginnen. Das Verständnis der Mittel zum Zweck verbessert jedoch auch das Verständnis des Zwecks. Design ist deshalb lösungsorientiert und betont die Bedeutung von Prototyping, um das Verständnis für Mittel und Zweck gemeinsam zu verbessern (vgl. [Cros2006]).

Die gemeinsame Betrachtung von Mittel und Zweck ist beim Gestalten digitaler Lösungen besonders wichtig. Digitales Material bietet neue Mittel, mit denen wiederum innovative Zwecke erreicht werden können. Digital Design bedeutet daher insbesondere, die digitale Lösung und das digitale System parallel zu gestalten.

1.2.3 Der Unterschied zwischen Auftraggeber-Rolle, Kunden-Rolle und Benutzer-Rolle

Aus der Sicht eines Umsetzungsteams müssen verschiedene Interessengruppen berücksichtigt werden, um eine gute digitale Lösung zu gestalten. Insbesondere die folgenden Stakeholder-Rollen verdienen besondere Aufmerksamkeit

- **Auftraggeber:** Stakeholder-Rolle, die eine digitale Lösung (Produkt/Service) beauftragt, um ein gewisses Kunden- und Benutzersegment anzusprechen. Die Auftraggeber-Rolle stellt erste Hypothesen auf, welchen Mehrwert sich Kunden wünschen und trägt die Verantwortung für den Gesamterfolg der digitalen Lösung.
- **Kunde:** Stakeholder-Rolle, die ein Produkt/eine Dienstleistung bezieht/nutzt, um einen (Mehr)Wert für sich und/oder andere zu schaffen. Diese Stakeholder-Rolle hat Erwartungen an Produkt/Service sowie an die gesamte Customer Experience.
- **Benutzer:** Stakeholder Rolle, die ein Produkt/Service aus eigener Motivation heraus oder von außen motiviert benutzt bzw. verwendet. Die Nutzerrolle hat Erwartungen an Produkt/Service sowie an die User Experience.

Je nach Perspektive ist es oft nicht einfach oder eindeutig zu entscheiden, welche Person welche Rolle einnimmt. Der Auftraggeber der digitalen Lösung kann sich beispielsweise selbst als Kunde des Lieferanten sehen und hat sicherlich auch einen Mehrwert aus dem beauftragten Produkt/Service.

Dies ist allerdings nicht die Perspektive, die dem Digital Designer bei der Gestaltung der digitalen Lösung hilft! Als Digital Designer ist es wichtig, die Rollen aus der Perspektive der digitalen Lösung selbst zu unterscheiden: Wer gibt die digitale Lösung in Auftrag (Auftraggeber), für wen soll die digitale Lösung einen Mehrwert schaffen (Kundensegmente) und wer wird die digitale Lösung benutzen/bedienen (Benutzersegmente).

Im Rahmen der Lösungsfindung sind alle Rollen gleichermaßen relevant. Die Stakeholder in der **Kunden-Rolle** geben an, welchen Mehrwert sie von der digitalen Lösung erwarten (z. B. Funktionsumfang, Qualität) und wie ein gutes Kundenerlebnis (z. B. Kaufprozess, Wartung, Service) aussehen sollte. Stakeholder in der **Benutzer-Rolle** geben ebenfalls Impulse zur digitalen Lösung in Bezug auf eine gute User Experience (z. B. Usability, Identifikation mit dem Produkt/Service) - insbesondere dann, wenn sie die digitale Lösung nicht aus eigener Motivation heraus nutzen, sondern von außen "motiviert" werden. Am Ende müssen allerdings Stakeholder in der **Auftraggeber-Rolle** entscheiden, wie viel Customer Experience und User Experience umgesetzt wird, da der Auftraggebende am Ende für den Erfolg der digitalen Lösung verantwortlich ist. Dies hat auch einen großen Einfluss darauf, ob die digitale Lösung von den Kunden und Benutzern akzeptiert wird, oder nicht.

Aus der Sicht des Umsetzungsteams sollte bereits im Scoping festgelegt werden, wer die Auftraggeber-, wer die Kunden-, und wer die Benutzer-Rolle einnimmt. Dies gilt sowohl für digitale Lösungen, bei denen alle Rollen in der eigenen Organisation zu finden sind, als auch für digitale Lösungen, bei denen potentielle Kunden und Benutzer auch außerhalb der Organisation zu finden sind.

Beispiel: Der Hersteller eines Batterieladegeräts für ein Elektroauto bietet eine Smartphone-App an, mit der das Ladegerät gesteuert werden kann (Start/Stopp des Ladevorgangs, Einstellung der Ladeleistung, Timer usw.). Diese App lässt er von einem Lieferanten erstellen. Der Hersteller des Ladegerätes ist hier in der Auftraggeber-Rolle, da er die App beim Lieferanten bestellt und den Funktionsumfang verantwortet.

Die **Rolle des Kunden** übernimmt diejenige, die das Ladegerät erwirbt - wahrscheinlich die Leasingnehmerin des Elektroautos, denn ohne aufgeladene Batterie ist das Auto nicht viel wert. Aus Kundensicht erfährt sie nicht nur das Produkt, sondern auch den Prozess der Kaufanbahnung und Kaufabwicklung sowie den Service und die Wartung des Ladegerätes (Customer Experience).

Die **Rolle des Benutzers** wird von der Person eingenommen, die das Ladegerät zum Aufladen des Elektroautos bedient. Aus Benutzersicht erfährt sie die User Experience des Ladegeräts und der Lade-App, im täglichen Gebrauch.

Benutzer und Kunde können in diesem Beispiel dieselbe Person sein - müssen aber nicht.

Eine klare Ausrichtung auf Auftraggeber-, Kunden- und Benutzer-Rolle verbessert das Innovationspotenzial

Ein klarer Fokus ist aus zwei Gründen wichtig. Der erste Grund ist, dass ein klares Bild davon bestehen muss, was aus der Sicht des Kunden und des Benutzers entworfen werden soll. Im Zusammenhang mit digitalen Lösungen stoßen wir häufig auf die implizite Annahme, dass der Benutzer und der Kunde ein und dieselbe Person sind. Eine Person, die beispielsweise ein Hotelzimmer auf der Website des Hotels bestellt, ist ein Kunde des Hotels und gleichzeitig ein Benutzer der Website.

Diese implizite Annahme schränkt den Lösungsraum digitaler Lösungen unnötig ein, weil sie davon ausgeht, dass der Mehrwert einer digitalen Lösung nur durch die direkte Interaktion mit der Lösung entsteht. Eine gute digitale Lösung kann auch durch indirekte Interaktion einen Mehrwert schaffen. Nehmen wir wieder das Hotelbeispiel: Der Kunde im Hotel könnte auch an der Rezeption anrufen und mit einem Hotelangestellten sprechen, der wiederum mit der Website interagiert, um das Zimmer als Benutzer zu buchen. Der Mehrwert für den Kunden ist in beiden Fällen der gleiche, aber die Interaktion ist völlig unterschiedlich. Für gutes Digital Design (siehe Abschnitt 1.4.2) ist es wichtig, diese Perspektiven zu trennen, um wertvolle nicht-digitale Aspekte einer digitalen Lösung zu erkennen, die mit digitalen Mitteln verbessert oder unterstützt werden können.

Der zweite Grund, warum ein klarer Fokus wichtig ist, besteht darin, dass die verschiedenen Perspektiven einer bestimmten Person auf die Stakeholder-Rollen nicht verwechselt werden dürfen. Erstens kann bei bestimmten digitalen Lösungen eine Person gleichzeitig in der Kunden-Rolle und in der Benutzer-Rolle sein. Eine solche Person hat vielleicht eine bestimmte Vorstellung vom Wert (die Kundenperspektive) und davon, wie der Wert geschaffen wird (Benutzerperspektive). Zweitens überschätzen Personen in einer Auftraggeber-Rolle oft ihr Wissen und ihr Verständnis für die Kunden- bzw. die Benutzerperspektive. Dies kann zu falschen Annahmen und oft zu suboptimalen oder sogar schwachen digitalen Lösungen führen. Um ein gutes Digital Design zu erzielen, ist es daher notwendig, die Vorgaben der Auftraggeber-Rolle sorgfältig auszuwerten und zu klären.

Im Alltag verwenden wir die Begriffe Benutzer und Kunde oft synonym. Es ist jedoch wichtig zu erkennen, dass diese Unterscheidung entscheidend ist, wenn sich die Kunden und Benutzer einer digitalen Lösung stark unterscheiden. Um die Lesbarkeit des Handbuchs zu verbessern, haben wir uns entschlossen, die Begriffe Benutzer und Kunde nur in den Situationen zu unterscheiden, in denen die Unterscheidung wichtig ist. In Situationen, in denen die Unterscheidung wichtig ist, empfehlen wir Ihnen daher ebenfalls auf eine präzise Verwendung dieser Begriffe zu achten.

1.2.4 Die YPRC-Fallstudie als Beispiel für eine digitale Lösung

Wir werden nun die eingeführte Terminologie anhand der YPRC-Fallstudie erläutern.

In unserer Fallstudie ist der Auftraggeber das Startup-Unternehmen, das die Finanzierung für den Aufbau der YPRC-Lösung anbieten könnte. Das Startup-Unternehmen glaubt, dass es einen potenziellen Markt für eine innovative digitale Lösung gibt, die Läufern Coaching-Dienste anbietet. Diese Überzeugung bringt uns zur Kundenperspektive.

Aus Kundensicht ist es das Ziel von YPRC, eine Rundumlösung für Einsteiger im Langstreckenlauf zu bieten, die keinen persönlichen Trainer haben oder sich diesen nicht leisten können, aber eine Betreuung wünschen, die über eine bestehende einfache Fitness-App hinausgeht. Im Mittelpunkt der YPRC-Lösung steht die Idee, ein Coaching-Erlebnis anzubieten, das mit dem eines erfahrenen persönlichen Trainers vergleichbar ist, der Seite an Seite mit dem Kunden läuft. Obwohl dies nur eine grobe Charakterisierung der YPRC-Lösung ist, schafft sie ein erstes Verständnis dafür, worum es bei YPRC geht.

Nun wollen wir uns die Systemebene ansehen. In der Storyline von YPRC bewertete das Team verschiedene Alternativen für die Gestaltung des digitalen Systems von YPRC. Im Wesentlichen wurden zwei alternative Systemformen diskutiert. Alternative A ist ein digitales System, bei dem der persönliche Trainer durch künstliche Intelligenz (KI) ersetzt wird, die dem Läufer Ratschläge erteilt. Alternative B ist ein digitales System, bei dem der Personal Coach über eine digitale Datenverbindung mit dem Läufer verbunden ist. Diese Datenverbindung dient dazu, die Gesundheitsdaten des Läufers an den Trainer zu übertragen und eine Sprachverbindung zwischen Läufer und Trainer herzustellen.

Auf den ersten Blick und ohne weitere Bewertung erscheinen beide Ideen vernünftig. Sie stellen jedoch völlig unterschiedliche Strategien dar, um das Ziel der potenziellen Kunden zu erreichen, das wir auf der Lösungsebene definiert haben. Daher wird die Form des digitalen Systems in jedem Fall anders aussehen. Mit diesem Beispiel wollen wir den wichtigen terminologischen Unterschied zwischen einem digitalen System und einer digitalen Lösung hervorheben. Das digitale System ist ein Werkzeug, und die digitale Lösung fügt die Perspektive einer Transformation dieses Werkzeugs in Richtung definierter *Ziele* hinzu. Der Aufbau einer digitalen Lösung erfordert die Berücksichtigung der Ziele zusammen mit dem digitalen System, das diese Ziele erreichen soll. Das Beispiel zeigt auch, dass das digitale System und die dahinterstehende Technologie (in diesem Fall die künstliche Intelligenz) die Ziele der digitalen Lösung erheblich beeinflussen können. Ohne die Idee, dass digitale Technologie ein Coaching-Erlebnis bieten kann, das mit dem eines echten Coachs vergleichbar ist, wäre die Lösungsidee von YPRC nicht realisierbar.

Wenn Alternative B realisiert wird, erfordert diese Form von YPRC eine Reihe von Elementen: den Läufer, der die YPRC-Smartwatch trägt und die YPRC-Smartphone-App nutzt, sowie ein Webportal, das mit der App verbunden ist und von dem Remote-Coach des Läufers genutzt wird. Die Beziehungen zwischen diesen Elementen lassen sich anhand einer einfachen Abbildung veranschaulichen:

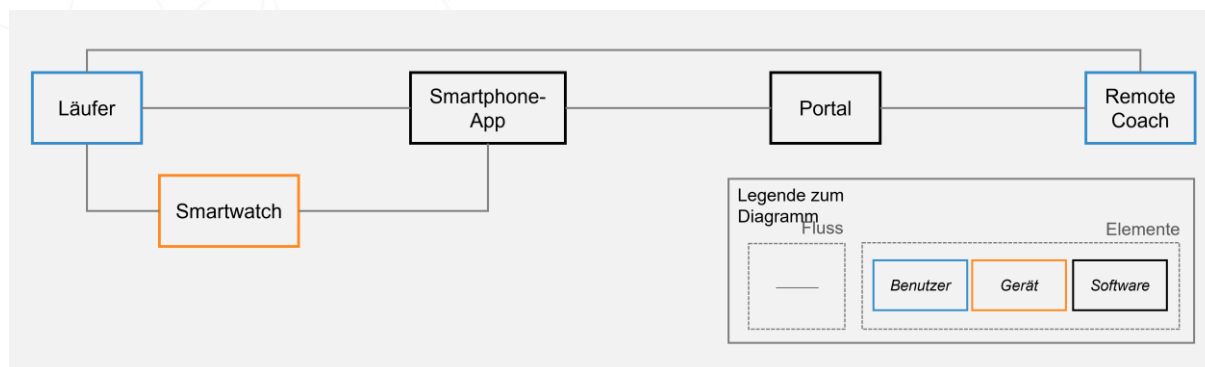


Abbildung 1 - Vereinfachte Darstellung der digitalen Lösung YPRC

Aus dieser Abbildung geht nicht viel hervor, außer dass es fünf Elemente gibt, die in einer gewissen Beziehung zueinander stehen. Dennoch vermitteln die Elemente von YPRC eine wichtige Botschaft: Ein digitales System innerhalb einer digitalen Lösung ist nicht auf Software beschränkt. Heutzutage kann eine digitale Lösung aus dedizierten Hardware-Geräten bestehen, die speziell als Teil einer digitalen Lösung gebaut wurden. Beispiele für solche Geräte sind Smartwatches, intelligente Lautsprecher und Geräte für die Hausautomatisierung. Wir werden uns nun die Funktionen von YPRC ansehen, um auf weitere Einzelheiten von YPRC einzugehen.

Die Funktion der YPRC-Smartwatch ist es, den Puls des Läufers zu messen und dem Läufer den aktuellen Puls anzuzeigen. Die Pulsdaten werden kontinuierlich von der Smartwatch an die Smartphone-App übertragen. Beachten Sie, dass sich der Begriff „kontinuierlich“ auf eine Eigenschaft dieser Funktion bezieht. Der Läufer kann die Smartphone-App verwenden, um die Pulsdaten nach dem Training einzusehen. Bis jetzt ist dies eine gewöhnliche Fitness-App.

Die charakteristischen Funktionen von YPRC werden durch das Webportal ermöglicht. Die Smartphone-App überträgt die Trainingsdaten des Läufers an das Portal. Das Portal nutzt die Technologie der künstlichen Intelligenz, um die Daten des Läufers zu analysieren und dem Läufer Trainingstipps zu geben. Der Läufer kann diese Trainingsunterstützung durch künstliche Intelligenz abonnieren, indem er eine monatliche Gebühr bezahlt. Außerdem kann der Läufer einen persönlichen Remote-Coach für eine Trainingseinheit buchen. Während einer Trainingseinheit kann der Remote-Coach die Daten des Läufers über das Portal in Echtzeit einsehen (man beachte die Qualität *Echtzeit*) und dem Läufer über eine Sprachverbindung sofortige Lauf Tipps geben (ebenfalls eine Qualität). Mit diesen Details über die Funktion von YPRC können wir eine aussagekräftigere Abbildung erstellen:

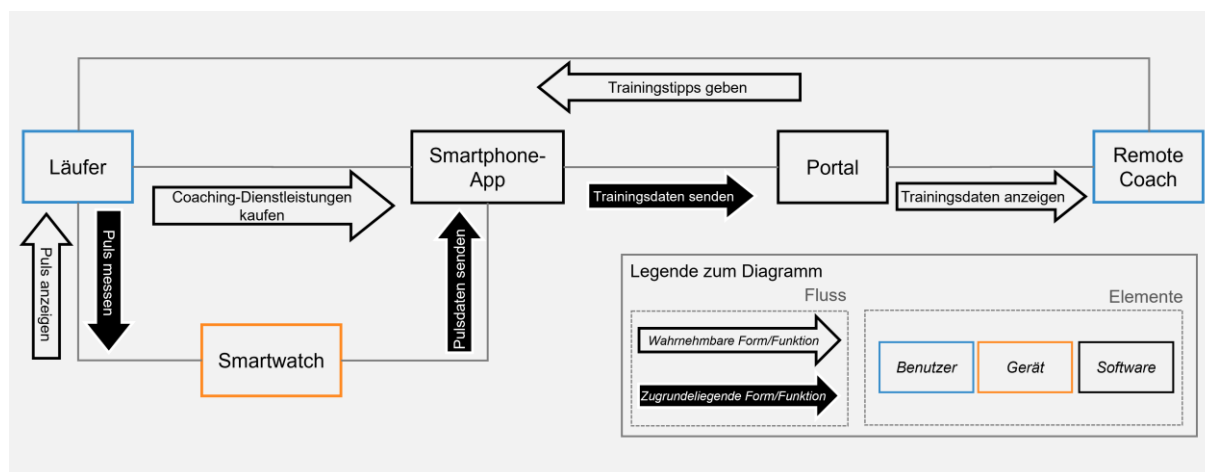


Abbildung 2 - Vereinfachte Form und Funktion der digitalen Lösung YPRC

Obwohl YPRC eine einfache digitale Lösung zu sein scheint, zeigt Abbildung 2 - Vereinfachte Form und Funktion der digitalen Lösung YPRC bereits ein kompliziertes System mit mehreren Beziehungen, Funktionen und Eigenschaften.

Die Idee der wahrnehmbaren und zugrundeliegenden Ebene ist die zweite Eigenschaft des Digitalen als Material und kann eine praktischere Sicht auf diese digitale Lösung schaffen. So gehört beispielsweise der Datentransfer zwischen Smartwatch, App und Portal zur zugrundeliegenden Form und Funktion des YPRC. Eine schnelle Berechnung und Datenübertragung (die zugrundeliegende Qualität) sind für die Lösung wichtig, weil sie notwendig sind, um den Remote-Coaching-Service in Echtzeit zu ermöglichen, aber aus der Perspektive des Läufers (und des Coachs) nicht wahrnehmbar sind. Der Läufer kann nur die aktuellen Trainingstipps wahrnehmen und der Remote-Coach kann nur die Echtzeit-Visualisierung der Trainingsdaten wahrnehmen. Um zu verdeutlichen, dass diese Aspekte zur zugrundeliegenden Form und Funktion gehören, ist die Beschreibung in Abbildung 2 in schwarzer Schrift gehalten.

Die Unterscheidung zwischen wahrnehmbarer und zugrundeliegender Form, Funktion und Qualität ist nicht nur ein Mittel zur einfacheren Strukturierung komplizierter digitaler Systeme: Sie ist in erster Linie eine zentrale Eigenschaft digitalen Materials. Relevante wahrnehmbare Funktionen einer digitalen Lösung, die einen echten Mehrwert schaffen, werden erst durch die zugrundeliegende Form, Funktion und Qualität ermöglicht.

Um digitale Lösungen ganzheitlich zu gestalten, bedeutet dies, dass die wahrnehmbare und zugrundeliegende Form, Funktion und Qualität immer gemeinsam betrachtet und gestaltet werden müssen. Digital Design bedeutet weder nur visuelles Design der Lösung noch Interaktionsdesign oder ausschließlich technisches Design des Systems. Eine einseitige Fokussierung auf wahrnehmbare (visuelle oder interaktive) oder zugrundeliegende (technische Datenflüsse) Aspekte wird immer zu suboptimalen oder gar nicht realisierbaren Lösungen führen.

Nehmen wir zum Beispiel an, YPRC würde sich für den Ansatz der künstlichen Intelligenz (KI) entscheiden und könnte eine wirklich starke KI entwickeln, die nützliche Trainingstipps für einen Läufer geben kann (zugrundeliegende Funktion). Nehmen wir weiter an, dass diese Tipps dann auf einer schlecht gestalteten Smartphone-App User Interface (wahrnehmbare Form) visualisiert werden: Es ist sehr wahrscheinlich, dass die Benutzer diesen Dienst nicht annehmen werden. Umgekehrt, wenn ein tolles Smartphone-User Interface schwache Trainingstipps einer schlechten künstlichen Intelligenz liefert, werden die Benutzer auch diesen Dienst nicht annehmen.

1.2.5 Ein Blick auf die Technologie in Bezug auf die Idee des Digitalen als Material

Um diese Einführung in die Idee des Digitalen als Material abzuschließen, ist ein Blick auf die Technologie notwendig. Vielleicht haben Sie bemerkt, dass in den Beschreibungen der YPRC-Fallstudie Fachbegriffe und die Benennung von Technologien im Wesentlichen weggelassen wurden. Es wurden nur die Begriffe App, Portal und künstliche Intelligenz (KI) verwendet.

Diese technologieneutrale Perspektive ist die dritte Eigenschaft der Idee des Digitalen als Material: Wir können auf einer sehr detaillierten und konkreten Ebene mit einem Minimum an technischem Wissen über eine digitale Lösung sprechen. Obwohl Software selbst eine wichtige Technologie ist, taucht der Begriff Software in unserer Beschreibung auch nicht auf.

Das Fehlen von Software spiegelt die vierte Eigenschaft der Idee des Digitalen als Material wider: Es ist möglich, eine digitale Lösung zu schaffen ohne Software zu programmieren. Wir können diese vierte Eigenschaft anhand eines sehr einfachen Beispiels veranschaulichen.

Nehmen wir an, wir wollen eine simple digitale Lösung für eine Pizzabestellung entwickeln. Unsere Pizzeria nutzt einen beliebten Messaging-Dienst als digitalen Kommunikationskanal (z. B. WhatsApp). Sie können Ihre Bestellung und Lieferadresse als Nachricht an das Restaurant senden. Das Restaurant bestätigt Ihre Bestellung mit einer Nachricht, die auch den zu zahlenden Preis enthält. Wenn Sie mit dem Preis einverstanden sind, können Sie Ihre Bestellung mit einem bestehenden Zahlungsdienst (z. B. PayPal) bezahlen. Sobald die Zahlung bestätigt wurde (das Restaurant erhält eine E-Mail vom Zahlungsanbieter), beginnt das Restaurant mit der Vorbereitung der Bestellung. Wenn die Bestellung zur Auslieferung bereit ist, sendet das Restaurant eine weitere Nachricht, um Sie darüber zu informieren, dass Ihre Pizza auf dem Weg ist. Diese digitale Lösung ist natürlich sehr primitiv, aber sie ist eine funktionierende Lösung für die Online-Pizzabestellung.

Dieses einfache Beispiel sollte jedoch nicht den Eindruck erwecken, dass die Entwicklung digitaler Lösungen keinerlei technisches Know-how erfordert - es ist sehr wohl erforderlich. Nur durch technischen Sachverstand ist es möglich, die Möglichkeiten der Technologie zu nutzen und digitale Lösungen auf einer technologieneutralen Ebene zu gestalten (vgl. dritte Eigenschaft des digitalen Materials), die dann auch tatsächlich realisiert werden können. Außerdem muss die verwendete Software natürlich im Voraus von jemandem entwickelt und bereitgestellt werden.

1.3 Einführung in den allgemeinen Bauprozess für digitale Lösungen

Mit dem Verständnis für digitales Material können wir nun den Bauprozess für eine digitale Lösung betrachten. Ein Prozess ist im Allgemeinen wie folgt definiert:

**Process: A set of interrelated activities performed
in a given order to process information or materials.**

Im Zusammenhang mit digitalen Lösungen kann das Wort *Bauen* auf den ersten Blick ungewöhnlich erscheinen. Wir haben uns bewusst für diesen allgemeinen Begriff entschieden, da er gut zu der Idee passt, das Digitale als Material zu begreifen. Bauen heißt für uns, eine digitale Lösung mit digitalem Material anzufertigen.

Im Folgenden führen wir eine Reihe von Begriffen ein, um den Bauprozess zu beschreiben. Diese Begriffe beziehen sich auf Tätigkeitsbereiche und mögliche Ergebnisse der Aktivitäten. In Abschnitt 1.4 verwenden wir diese Tätigkeitsbereiche, um das Kompetenzprofil eines DDP zu definieren.

Die Tätigkeitsbereiche dürfen nicht mit den Rollen innerhalb einer Projektstruktur verwechselt werden. Rollen können aus diesen Tätigkeitsbereichen definiert werden, hängen aber vom jeweiligen Prozessmodell oder der Projektsituation ab.

Bestehende Disziplinen für die Entwicklung von Elementen einer digitalen Lösung (z. B. Software-Engineering, Industriedesign, Usability Engineering, Produktmanagement, Softwaretests usw.) können mit einem oder mehreren der vorgestellten Tätigkeitsbereiche in Einklang gebracht werden. Wo nötig, werden in diesem Handbuch kurze Verweise auf bestehende Disziplinen

gegeben. Weitere Einzelheiten über die Beziehung zu bestehenden Disziplinen sind jedoch nicht Teil dieses Handbuchs.

Das bedeutet nicht, dass ein DDP diese Disziplinen ignorieren oder vernachlässigen sollte. Tatsächlich ist das genaue Gegenteil der Fall. Mit diesem allgemeinen Verständnis des Bauprozesses ist ein DDP in der Lage, mit einer Vielzahl von bestehenden Disziplinen zusammenzuarbeiten. In diesem Abschnitt wird der Bauprozess für eine digitale Lösung schematisch beschrieben. Wir unterscheiden drei Kerntätigkeitsbereiche und zwei übergreifende Tätigkeitsbereiche eines Bauprozesses:

- Haupttätigkeitsbereiche
 - Design
 - Konstruktion
 - Realisierung
- Querschnittliche Tätigkeitsbereiche
 - Management
 - Evaluierung

Dieses Grundverständnis des Bauprozesses ist wichtig, um seine verschiedenen Aspekte zu verstehen und um die Integration von Digital Design in den Bauprozess zu begreifen. Dieses Verständnis ist jedoch nicht ausreichend, um einen konkreten Bauprozess zu definieren und durchzuführen. Ein konkreter Prozess für Einsteiger ist in Kapitel 5 beschrieben.

1.3.1 Querschnittliche Tätigkeitsbereiche

1.3.1.1 Management des Bauprozesses

Der Bau einer digitalen Lösung ist so komplex, dass ein eigener Tätigkeitsbereich für das Management (die Steuerung) erforderlich ist. Dieser ist wie folgt definiert:

Management: Leading the building process in cooperation with all other activities.

Bei der Steuerung des Bauprozesses unterscheiden wir zwischen drei Perspektiven:

- *Projektmanagement-Perspektive*: Koordination von Aktivitäten, Zeit und Budget
- *People-Management-Perspektive*: Management der Erwartungen der Stakeholder, Management des Erkenntnisprozesses der Stakeholder, Beschaffung der richtigen Leute und Fähigkeiten für die jeweilige Aktivität
- *Produktmanagement-Perspektive*: Entwicklung einer kurz- und langfristigen Strategie für die Entwicklung der digitalen Lösung

Die Perspektiven des Projekt- und des People-Management werden in Abschnitt 1.3.4 näher erläutert. Das Produktmanagement geht über das Grundlagenniveau hinaus und ist daher nicht ausdrücklich Bestandteil dieses Handbuchs.

1.3.1.2 Evaluierung der digitalen Lösung

Die Evaluierung des Tätigkeitsbereichs soll sich auf die Qualität der Arbeitsprodukte konzentrieren. Die realisierte digitale Lösung wird ebenfalls als Arbeitsprodukt betrachtet. Wir definieren Evaluierung wie folgt:

Evaluation: A systematic process for determining the value, quality, or appropriateness of something.

Im Digital Design bestimmt die Evaluierung, ob eine digitale Lösung oder ein Arbeitsprodukt, das zur Erstellung einer digitalen Lösung verwendet wird, tatsächlich die Qualitäten und Eigenschaften besitzt, die es gemäß den Designkonzepten und den Bedürfnissen der Stakeholder haben sollte. Das bedeutet, dass die Evaluierung im Bauprozess immer mit einem Arbeitsprodukt und somit mit einem der Kerntätigkeitsbereiche verbunden ist. Daher betrachten wir die Evaluierung als untrennbaren Bestandteil der Kerntätigkeitsbereiche. Die jeweilige Perspektive auf die Evaluierung wird folglich stets als Teil der Kerntätigkeitsbereiche beschrieben (siehe 1.3.2). Die eigenständige Definition von Evaluierung soll aber die Bedeutung und die allgemeine Anwendbarkeit von Evaluierung im Bauprozess betonen.

Um die Arbeit der Evaluierung zu erfassen und zu strukturieren, definieren wir einen eigenen Konzepttyp:

Evaluation concept: A description of the evaluation approach for a work product.

Evaluationskonzept ist ein Oberbegriff, um den breiten Umfang der Qualitätssicherung während des Bauprozesses zu erfassen. Im Gegensatz zu den Konzepten, die durch die Kerntätigkeitsbereiche erstellt werden (siehe Abschnitt 1.3.2), beschreibt das Evaluationskonzept nicht die digitale Lösung. Ein Evaluationskonzept beschreibt den Ansatz zur Evaluierung eines bestimmten Arbeitsproduktes des Bauprozesses. Der Grund dafür ist, dass der Evaluierungsansatz explizit gemacht werden soll.

1.3.2 Kerntätigkeitsbereiche des Bauprozesses

1.3.2.1 Design einer digitalen Lösung

Design ist ein Tätigkeitsbereich, der sich mit der Zukunft beschäftigt und damit, welche (digitalen) Lösungen eine gegenwärtige in eine gewünschte Situation verwandeln können. Wie bereits in 1.1.2 erwähnt, ist Design ein besonderer Begriff, mit zahlreichen Bedeutungen (vgl. [ErMa2008]).

Design: 1. A plan or drawing produced to show how something will look, function, or be structured before it is made.

2. The activity of creating a design.

Der erste Teil der Definition verwendet den Begriff Design als Ergebnis. Dieses Ergebnis ist definiert als ein Plan oder eine Zeichnung, die erstellt wird, um zu zeigen, wie etwas aussehen, funktionieren oder aufgebaut sein wird, bevor es hergestellt wird. Der zweite Teil der Definition bezieht sich auf die Aktivität zum Erstellen dieses Ergebnis also Design im Sinne des Gestaltens von etwas.

Design bedeutet also auch, sich eine angestrebte Zukunft vorzustellen und diese mit Hilfe von Designkonzepten (siehe unten) zu beschreiben. Dieses Verständnis von Design hat drei Aspekte:

1. Erarbeiten und verstehen der angestrebten Zukunft
2. Definieren und Gestalten einer digitalen Lösung, die diese Zukunft mittels eines Designkonzepts schaffen soll
3. Evaluierung der Qualität des Designkonzepts

Der erste Aspekt erfordert Einfühlungsvermögen, Vorstellungskraft und Kreativität. Der zweite Aspekt erfordert Kompetenzen in digitaler Technologie und Konzeptarbeit. Der dritte Aspekt erfordert Kompetenzen in verschiedenen Bereichen der Qualitätssicherung (z. B. technische Machbarkeit, Funktionstauglichkeit, Benutzbarkeit).

Über eine gewünschte und angestrebte Zukunft zu sprechen, ist eine ziemlich abstrakte und weitreichende Überlegung. In der Designliteratur werden häufig alternative Begriffe wie *Designprobleme* oder *Designziele* verwendet. Wir bevorzugen jedoch „Zukunft“ anstelle von „Problemen“ oder „Zielen“, weil dies beschreibt, worum es beim Design wirklich geht: sich eine Zukunft vorzustellen und sie zu gestalten.

Die Vision einer angestrebten Zukunft, gemeinsam mit allen relevanten Stakeholdern zu entwerfen, ist eine echte Herausforderung, die keinesfalls unterschätzt werden darf.

Um die angestrebte Zukunft zu beschreiben, werden während des Designs Konzepte erstellt, die die digitale Lösung beschreiben, mit der die gewünschte Zukunft erreicht werden soll. Diese Konzepte werden als Designkonzepte bezeichnet und sind wie folgt definiert:

Design concept: A description of the design of a digital solution, of a digital system, or of an element of a digital solution under the assumption of perfect technology.

Designkonzept ist ein allgemeiner Begriff. Es kann als Bauplan für die digitale Lösung in verschiedenen Abstraktionsebenen betrachtet werden, der alle wichtigen Elemente (z. B. spezielle Geräte, User Interfaces, Funktionen usw.) enthält. Denken Sie daran, dass eine digitale Lösung mehr ist als nur Software. Das bedeutet, dass der Begriff Designkonzept auch die Beschreibung von Geräten umfasst, die speziell für die vorliegende digitale Lösung entworfen wurden. Infolgedessen kann die Erstellung des Designkonzepts die Einbeziehung verschiedener Disziplinen erfordern (z. B. Requirements Engineering, Interaktionsdesign, Industriedesign und Service-Design).

Die Annahme perfekter Technologie [WaMe1986] ist eine wichtige Vereinfachung für das Design digitaler Lösungen und bedeutet im Wesentlichen fehlerfreie Technologie sowie unendliche Rechenkapazität, Speicherkapazität und unendliche Kommunikationskapazität. Sie vereinfacht die Entwicklung von Designkonzepten in mehrfacher Hinsicht. Wir werden auf die Details der Designkonzepte und die Annahme einer perfekten Technologie in Abschnitt 2.2.5.1 zurückkommen, wo wir eine detaillierte Einführung in die Konzeptarbeit geben.

Die Verantwortung für das Design endet nicht mit dem Designkonzept. Die obige Definition umfasst auch zwei wichtige Qualitätssicherungsaspekte des Designs.

Die *Vision einer angestrebten Zukunft vor Augen zu haben* bedeutet, dass im Design sichergestellt werden muss, dass die angestrebte Zukunft für alle relevanten Stakeholder wünschenswert ist. Bei diesem Aspekt geht es darum, alle relevanten Stakeholder einzubeziehen, um ihre Akzeptanz für die festgelegte Zukunft sicherzustellen.

Die *Erstellung eines Designkonzepts für eine digitale Lösung, die diese Zukunft schaffen soll*, bedeutet, dass im Design sichergestellt werden muss, dass das Designkonzept das Potenzial hat, die angestrebte Zukunft zu schaffen, und dass die definierte digitale Lösung die angestrebte Zukunft hervorbringt. Dieser Aspekt hat zwei Dimensionen.

In der ersten Dimension steht das Designkonzept selbst im Mittelpunkt der Qualitätssicherung, d. h. die definierte Form, Funktion und Qualität müssen validiert werden, um festzustellen, ob sie geeignet und in der Lage sind, die angestrebte Zukunft zu schaffen (*Bauen wir die richtige digitale Lösung?*). Dieser Aspekt erfordert auch die Einbeziehung von Stakeholdern, um das Designkonzept zu validieren. Neben dem Designkonzept ist die Erstellung von Prototypen eine wichtige Technik zur Validierung bestimmter Aspekte eines Designkonzepts. In Abschnitt 2.3 finden Sie weitere Einzelheiten über die Anwendung von Prototyping im Design.

Der zweite Aspekt betrifft die ordnungsgemäße Konstruktion und Umsetzung der digitalen Lösung gemäß dem Designkonzept (*Bauen wir die digitale Lösung richtig?*). Dieser Aspekt erfordert eine enge Zusammenarbeit zwischen den Tätigkeitsbereichen Konstruktion und Realisierung. Einzelheiten zu dieser Zusammenarbeit werden in Abschnitt 1.3.3 unmittelbar nach der Einführung dieser Begriffe erörtert.

1.3.2.2 Konstruktion einer digitalen Lösung

Die Konstruktion ist ein Tätigkeitsbereich, der sich mit den technischen Details einer digitalen Lösung befasst, um ihre Realisierung vorzubereiten. Wir definieren Konstruktion wie folgt:

Construction: The creation of the realization concept of a digital solution that will create the desired future.

Dieses Verständnis von Konstruktion hat zwei Aspekte: Erstens die Erstellung des Realisierungskonzepts der digitalen Lösung und zweitens die Evaluierung, dass die digitale Lösung, die durch dieses Konzept beschrieben wird, die angestrebte Zukunft schafft, die durch die Gestaltungsaktivität angestrebt wird.

Realisierungskonzept ist wiederum ein Oberbegriff, er ergänzt das Designkonzept und ist wie folgt definiert:

Realization concept: A description of a digital solution with real technology.

Das Realisierungskonzept muss reale Technologie verwenden und sich mit allen technischen Details befassen, die notwendig sind, um die digitale Lösung und ihre Elemente zu realisieren³. Dies kann die Definition der physikalischen Struktur (z. B. physikalische Komponenten und Materialien) und der technischen Struktur (z. B. Mikroprozessor und Platine) von dedizierten Geräten, die Definition der Softwarestruktur (z. B. Softwarekomponenten), die Definition der Realisierungstechnologie (z. B. Programmiersprachen, Frameworks, technische Sensoren usw.) und die Definition der technischen Infrastruktur (z. B. die Definition von geeigneten Rechenzentren) umfassen.

Dieses breite Spektrum zeigt deutlich, dass die Konstruktion einer digitalen Lösung die Einbeziehung verschiedener Disziplinen erfordern kann (z. B. Software Engineering, Industriedesign und Produktionstechnik) und dass Realisierungskonzepte verschiedene Instanzen haben (z. B. Software-Architekturkonzepte, physische Gebäudepläne, elektronische Layouts). Auf die Herausforderung, verschiedene Disziplinen einzubeziehen, werden wir in Abschnitt 1.3.4 zurückkommen.

³ Dieses Handbuch konzentriert sich auf Designkonzepte und bietet keine weiteren Anleitungen für die Arbeit an Realisierungskonzepten.

Natürlich hängen das Realisierungskonzept und das Designkonzept voneinander ab. Wir erörtern die Beziehung zwischen beiden Konzepten in Abschnitt 1.3.3.1, wenn wir die Zusammenarbeit zwischen den Tätigkeitsbereichen Design und Konstruktion diskutieren.

Bei der Qualitätssicherung in der Konstruktion geht es um das Realisierungskonzept. Wie das Designkonzept hat auch die Qualitätssicherung des Realisierungskonzepts zwei Dimensionen.

Erstens muss das Realisierungskonzept die notwendigen technischen Fähigkeiten beschreiben, um die anvisierte Zukunft zu schaffen. Dazu gehört insbesondere der Aspekt, dass die definierten Technologien bestimmte Qualitäten erreichen müssen (z. B. Zuverlässigkeit der digitalen Lösung). In Abschnitt 2.1 gehen wir näher auf die Beziehung zwischen Qualität und Technologie ein.

Zweitens muss die Konstruktion sicherstellen, dass die definierten Realisierungskonzepte auch wirklich umgesetzt werden. Hier ist eine enge Zusammenarbeit zwischen den Tätigkeitsbereichen Design und Realisierung erforderlich (siehe Abschnitt 1.3.3).

1.3.2.3 Realisierung einer digitalen Lösung

Der Tätigkeitsbereich Realisierung befasst sich mit der faktischen Umsetzung der digitalen Lösung und ist wie folgt definiert:

Realization: The implementation of the digital solution according to the defined design concepts and realization concepts.

Wie bei der Konstruktion hat dieses Verständnis von Realisierung zwei Aspekte: Erstens die Umsetzung der digitalen Lösung gemäß dem Designkonzept und dem Realisierungskonzept und zweitens die Sicherstellung, dass die implementierte digitale Lösung die angestrebte Zukunft schafft, die durch die Design- und Konstruktionsaktivitäten angestrebt wird.

Das Realisieren einer digitalen Lösung ist keineswegs ein triviales Unterfangen. Wie bei der Konstruktion kann das Realisieren einer digitalen Lösung die Einbeziehung verschiedener Disziplinen erfordern. Je nach Art der Elemente einer digitalen Lösung müssen verschiedene Disziplinen beteiligt sein:

- Spezielle Geräte: Planung und Durchführung der Herstellung und Lieferung der Geräte, einschließlich der Entwicklung der Softwareteile des Geräts
- Softwareelemente: Einrichtung einer Entwicklungsumgebung für die eigentliche Implementierung, Einrichtung einer Produktionsumgebung für den Betrieb der Softwareelemente und Rollout der Software

Die Evaluierung während der Realisierung ist für jeden Bauprozess von grundlegender Bedeutung. Beim Realisieren muss sichergestellt werden, dass die digitale Lösung entsprechend dem Designkonzept und dem Realisierungskonzept umgesetzt wird, sodass die digitale Lösung die angestrebte Zukunft erschafft. Diese Qualitätssicherung wird in Zusammenarbeit der Tätigkeitsbereiche Design und Konstruktion durchgeführt.

1.3.3 Den Bauprozess als parallelen Prozess begreifen

Aus einer unreflektierten Sichtweise können die Kerntätigkeitsbereiche als Folgeaktivitäten betrachtet werden: Die am Design Beteiligten erstellen das Designkonzept, woraus die an der Konstruktion Beteiligten ein Realisationskonzept erstellen, das von den an der Realisierung

Beteiligten umgesetzt wird. Ein solcher Ansatz wird oft als Wasserfall-Ansatz bezeichnet, wird aber seit den Anfängen der Softwareentwicklung als unpraktisch angesehen (vgl. [Royce1970]).

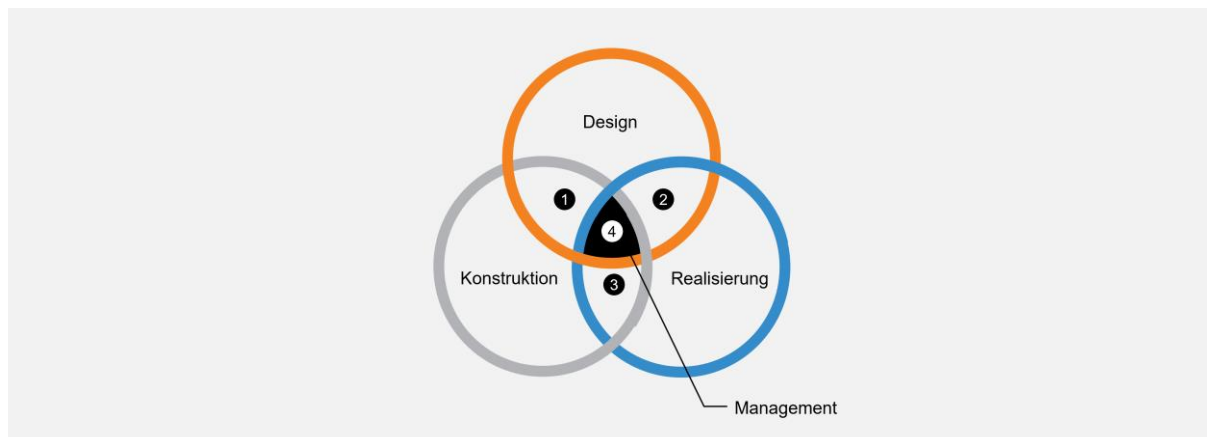


Abbildung 3 - Überschneidungen zwischen den Kerntätigkeitsbereichen des Bauprozesses

Eine realistischere Sicht des Bauprozesses ist das Verständnis aller drei Kerntätigkeitsbereiche als fortlaufende Aktivitäten, die parallel durchgeführt werden. Abbildung 3 zeigt die Aktivitäten des Bauprozesses in Form eines Venn-Diagramms. Die vier sich überschneidenden Bereiche sind wichtig, weil sie den Bereich definieren, in dem die verschiedenen Aktivitäten während des Bauprozesses zusammenwirken.

Im Folgenden wird das Zusammenwirken von Design, Konstruktion und Realisierung einschließlich der Bewertung detailliert erörtert (die Überschrift jedes Unterabschnitts bezieht sich auf die Nummer des betrachteten Bereichs). Leser, die bereits Erfahrung mit der Entwicklung digitaler Lösungen haben, werden mit den dargestellten Inhalten bereits vertraut sein. Die umfassende Einführung richtet sich vor allem an Einsteiger. Die Verwaltung des gesamten Bauprozesses wird in Abschnitt 1.3.4 behandelt.

1.3.3.1 Zusammenarbeit zwischen Design und Konstruktion (1)

Der Unterschied zwischen Design und Konstruktion ist für das Verständnis des Bauprozesses für eine digitale Lösung sehr wichtig. Beide Tätigkeitsbereiche arbeiten auf einer konzeptuellen Ebene und verwenden häufig dieselben Sprachen für die Kommunikation und Dokumentation (z.B. Diagramme oder technische Zeichnungen). Daher ist es recht einfach, beide Tätigkeitsbereiche durcheinander zu werfen.

Bei Design und Konstruktion geht es um die Perspektive

Der wichtigste Unterschied zwischen Design und Konstruktion ist die Perspektive. Design betrachtet das Äußere einer digitalen Lösung mit dem Ziel, die angestrebte Zukunft zu verstehen und auszuarbeiten. Konstruktion blickt auf das Innere einer digitalen Lösung und beschäftigt sich mit der Definition der technischen Umsetzung der angestrebten Zukunft.

Um diesen Unterschied zu verdeutlichen, haben wir bewusst eigene Ergebniskategorien definiert - Designkonzept für das Design und Realisierungskonzept für die Konstruktion - und den Begriff der perfekten Technologie eingeführt. Idealerweise definieren Designkonzepte und Realisierungskonzepte zusammen die gesamte digitale Lösung ohne jegliche Redundanz. Weitere Einzelheiten zur Erstellung von Designkonzepten, einschließlich der Annahme einer perfekten Technologie, werden in Abschnitt 2.2.5.1 erörtert.

Beispiel YPRC. Tabelle 2 zeigt beispielhaft die Beziehungen zwischen einem Designkonzept und einem Realisierungskonzept anhand der YPRC-Fallstudie.

Die Beispiele zeigen, dass die technische Machbarkeit und der Freiheitsgrad eine wichtige gemeinsame Aufgabe von Design und Konstruktion sind. Jede Entscheidung während des Designs oder der Konstruktion kann Auswirkungen auf den anderen Tätigkeitsbereich und auf die endgültige digitale Lösung haben.

Tabelle 2 - Beispielhafte Beziehungen zwischen einem Designkonzept und einem Realisierungskonzept

Designkonzept	Realisierungskonzept
Mock-up einer Benutzer-Schnittstelle (User Interface) der YPRC-Smartphone-App	Technische Komponenten und Bibliotheken für das User Interface: HTML, CSS und Angular-Java-Skript
Datenmodell, das die Gesundheitsdaten des Läufers beschreibt	Tabellen in einer SQL-Datenbank zur Speicherung der Gesundheitsdaten
Kommunikation zwischen der YPRC-Smartphone-App und dem Portal	Webdienstbeschreibung der Schnittstellen
Form des Gehäuses der YPRC Smartwatch	Technische Form des Gehäuses, einschließlich der Fertigungspläne

Auch wenn Design und Konstruktion unterschiedliche Tätigkeitsbereiche sind, müssen sie zusammenarbeiten und sich gegenseitig ergänzen. Für jede Form, Funktion oder Qualität, die durch eine Gestaltungsaktivität definiert wird, ist eine entsprechende technische Umsetzung notwendig, um sie zu realisieren.

Technische Entscheidungen bedeuten eine gemeinsame Verantwortung, die Innovationspotenziale schafft

Aus praktischer Sicht muss es möglich sein, eine während des Designs entworfene Form, Funktion oder Qualität mit der verfügbaren Technologie zu realisieren⁴. Personen, die Designentscheidungen treffen, müssen über genügend technisches Fachwissen verfügen, um die Machbarkeit ihrer Entscheidungen zu beurteilen, oder sie müssen erkennen, wann sie Fachwissen aus der Konstruktion hinzuziehen müssen.

Beispiel YPRC. Angenommen, jemand definiert, dass die künstliche Intelligenz auf der Smartphone-App läuft und dem Läufer über einen künstlichen Sprachassistenten Trainingstipps in Echtzeit gibt. Eine solche Funktion erfordert eine erhebliche Rechenleistung, die auf einem Smartphone möglicherweise nicht zur Verfügung steht. In dieser Situation ist ein Konstruktionsexperte mit Kenntnissen über Smartphones erforderlich, um die technische Machbarkeit dieser Funktion zu bewerten.

⁴ Wenn eine angestrebte Form, Funktion oder Qualität derzeit nicht realisierbar ist, kann dies den Anstoß zu Forschungs- und Entwicklungsaktivitäten geben, die zu Technologien führen könnten, die sie realisierbar machen. Dies ist jedoch nicht Gegenstand dieses Handbuchs.

Umgekehrt müssen Personen, die Konstruktionsentscheidungen treffen, über ausreichende Designkenntnisse verfügen, um die Auswirkungen ihrer Entscheidung auf das Design einer digitalen Lösung beurteilen zu können, und sie müssen erkennen, wann sie Fachleute aus dem Tätigkeitsbereich Design hinzuziehen müssen. Nehmen wir zum Beispiel an, dass während des Aufbaus der YPRC-Fallstudie jemand beschließt, eine Open-Source-SQL-Datenbank für die Speicherung der Benutzer- und Gesundheitsdaten im Portal zu wählen. Nehmen wir weiter an, dass diese Datenbanktechnologie nur bis zu einer bestimmten Datenmenge und einer bestimmten Anzahl von Transaktionen pro Sekunde skaliert werden kann. Eine solche Entscheidung kann sich auf die Skalierbarkeit und die Anzahl der Benutzer auswirken, die die Software verarbeiten kann. In diesem Fall muss diese technische Einschränkung mit Personen aus dem Design besprochen werden, um festzustellen, ob die Grenze für die endgültige Realisierung akzeptabel ist.

In der Zusammenarbeit zwischen Design und Konstruktion geht es nicht nur um Grenzen und Beschränkungen - Design- und Konstruktions-Entscheidungen können auch zusätzliche Möglichkeiten schaffen. Wenn beispielsweise bei der Konstruktion eine Technologie ausgewählt wird, die zusätzliche Funktionen ermöglicht, an die im Design nicht gedacht wurde, sollte die Konstruktion den Tätigkeitsbereich Design über diese Möglichkeiten informieren, um sie eventuell in die digitale Lösung zu integrieren.

1.3.3.2 Zusammenarbeit zwischen Design und Realisierung (2)

Die Zusammenarbeit zwischen Design und Realisierung muss je nach Art des zu realisierenden Elements unterschieden werden. Bevor wir die speziellen Eigenschaften von Software und physischen Teilen im Detail diskutieren, wollen wir zunächst die allgemeinen Aspekte der Zusammenarbeit zwischen den Tätigkeitsbereichen Design und Realisierung betrachten:

- *Klärung konzeptueller Details für die Realisierung:* Das Design muss die konzeptuellen Details der digitalen Lösung liefern, die für die Umsetzung der Lösung erforderlich sind. Gleichzeitig hat die Realisierung die Aufgabe, auf Ungenauigkeiten und Lücken im Designkonzept der Lösung hinzuweisen, damit das Design diese korrigieren und verbessern kann.
- *Qualitätssicherung der wahrnehmbaren Form, Funktion und Qualität:* Die wahrnehmbare Form, Funktion und Qualität bilden die sichtbare Ebene der digitalen Lösung für die Benutzer und liegen in der Verantwortung des Designs. Daher ist das Design dafür verantwortlich, dass die wahrnehmbare Form, Funktion und Qualität der digitalen Lösung die angestrebte Zukunft schafft und gemäß dem Designkonzept realisiert wird. Der konkrete Ansatz zur Erfüllung dieser Verantwortung ist abhängig von dem jeweiligen Bauprozess.

Design und Realisierung sollten so früh wie möglich zusammenarbeiten

Die Zusammenarbeit zwischen den Tätigkeitsbereichen Design und Realisierung kann bereits in einem sehr frühen Stadium des Bauprozesses mit der Erstellung von Prototypen zu Evaluationszwecken beginnen, die nicht allein durch das Design erstellt werden können. Beispiele für solche Prototypen sind interaktive Prototypen, die eine tatsächliche Implementierung von Software erfordern (siehe unten). In einer solchen Situation sollte ein Evaluationskonzept erstellt werden, das das Ziel und die Maßnahmen der Evaluationsaktivität beschreibt. Beispielsweise wird ein HTML-User Interface-Prototyp für eine App implementiert, um die Benutzbarkeit des gesamten visuellen Designs der App zu bewerten. Das Evaluationskonzept

sollte die konkrete Form des Prototyps und das Verfahren zur Durchführung der Evaluierung festhalten.

Bei digitalen Hardware-Geräten endet die Entwicklung eines physischen Geräts mit einem Modell des Produkts (auch Prototyp genannt), das in einem teuren und anspruchsvollen Massenproduktionsprozess reproduziert wird. Im Gegensatz dazu ist das Design eines Softwareelements eine fortlaufende Aktivität während des gesamten Realisierungsprozesses.

Der Grund für diesen Unterschied liegt in der inhärenten Komplexität der Implementierung des Softwareteils digitaler Lösungen. Die Implementierung (auch Programmierung genannt) von Software wird als intellektuelle Herausforderung angesehen, da selbst die einfachsten Programme den menschlichen Verstand an seine Grenzen bringen können (vgl. [Glas2006], [Wein1971]). Wichtige Fragen zu den Details einer Software-Implementierung stellen sich während der eigentlichen Implementierung der Software, weil der Akt des Programmierens uns zwingt, in logisch präzisen Strukturen der Programmiersprache zu denken.

Späte Designentscheidungen sind ein wichtiger Aspekt im Digital Design

Für die Zusammenarbeit zwischen den Tätigkeitsbereichen Design und Realisierung bedeutet dies, dass verschiedene detaillierte Designentscheidungen (z. B. das Verhalten einer Software in Ausnahmesituationen) während der Realisierung getroffen werden müssen, um die Software richtig zu implementieren. In der Praxis wird diese Detailarbeit an der Software oft als unbefriedigend empfunden, da die Entwicklung von Software in allen Details oft sehr langwierig ist.

Dennoch ist die Möglichkeit von späten Designentscheidungen die besondere Stärke von Software. Richtig genutzt, kann diese Möglichkeit die Geschwindigkeit und Effizienz des gesamten Bauprozesses erheblich steigern. Bei einem guten Bauprozess werden späte Designentscheidungen bewusst eingeplant. Das bedeutet, dass nur diejenigen Designentscheidungen getroffen werden, die im Vorfeld der Realisierung getroffen werden müssen.

Ein gutes Beispiel für die Vorteile später Designentscheidungen ist die Benutzer-Schnittstelle (User Interface) einer Software. User Interfaces, die für die wichtigsten Use Cases einer Software relevant sind, sollten so früh wie möglich entworfen werden, um die Benutzerakzeptanz der Entwürfe zu bewerten. Das Design von User Interfaces mit geringerer Bedeutung kann kurz vor der eigentlichen Realisierung erstellt werden.

Ein zweiter Vorteil von späten Designentscheidungen bei der Softwarerealisierung kann bei der Qualitätssicherung erzielt werden. Bestimmte Aspekte einer Software lassen sich am besten im Echtbetrieb validieren. Ein gutes Beispiel hierfür ist das Verhalten von Software im Falle von Fehlern. Die Benutzbarkeit der Visualisierung von Fehlermeldungen in Ausnahmefällen hängt stark von der realen Situation des Benutzers ab. Die Validierung und Verbesserung des Designs einer Software lässt sich in solchen Situationen am besten durch die reale Implementierung erreichen.

Die Möglichkeit der späten Designentscheidungen richtig zu nutzen, ist eine echte Herausforderung und erfordert eine sorgfältige Vorbereitung (z. B. die Auswahl der richtigen Technologie, um Flexibilität bei der Umsetzung zu ermöglichen) und Erfahrung (z. B. die Identifizierung derjenigen Entscheidungen beim Design der Software, die verschoben werden können).

Das Design eines physischen Produkts erfordert eine andere Zusammenarbeit

Die Realisierung eines physischen Produkts erfordert einen anderen Ansatz der Zusammenarbeit. Industriedesign und Produktentwicklung sind Disziplinen, die sich mit Massenprodukten befassen. Ein umfassender Überblick über dieses Thema würde den Rahmen dieses Handbuchs sprengen. Dennoch geben wir eine kurze Einführung, um wichtige Herausforderungen hervorzuheben.

Alle relevanten Designentscheidungen und Qualitätssicherungsmaßnahmen müssen getroffen werden, bevor die Massenproduktion des physischen Produkts beginnt. Die Zusammenarbeit zwischen den Tätigkeitsbereichen Design und der Realisierung physischer Produkte stützt sich daher in hohem Maße auf verschiedene Arten von Prototypen. In Abschnitt 2.3 werden weitere Einzelheiten zu den Prototypen erläutert. Im Folgenden werden drei Beispiele für solche Prototypen vorgestellt:

- Der *Funktionsprototyp* hat nicht die endgültige Form des Produkts, sondern enthält alle relevanten Funktionselemente. Er wird erstellt, um die technische Funktion eines Produkts zu validieren.
- Der *Erscheinungsprototyp* bietet keine Funktionalität, sondern stellt die endgültige Form des Produkts dar. Er wird verwendet, um das visuelle Erscheinungsbild des Produkts bei den Benutzern zu validieren.
- Der *Vorserienprototyp* ist ein vollständig realisiertes Muster des Produkts für endgültige und umfassende Qualitätssicherungsmaßnahmen.

Der letzte Aspekt der Zusammenarbeit zwischen den Tätigkeitsbereichen Design und Realisierung ist die Qualitätssicherung der implementierten Lösung. Typische Beispiele für diesen Aspekt sind Benutzer-/Kundenakzeptanztests und Usability-Tests. Zu diesem Zweck können verschiedene Arten von Evaluationskonzepten definiert werden. Sie umfassen typischerweise Testfälle, die die konkrete Anwendung der digitalen Lösung beschreiben, einschließlich Testdaten, anhand derer die Umstände, unter denen die digitale Lösung verwendet wird, definiert werden. Zu den anspruchsvolleren Evaluierungsansätzen gehören automatisierte Tests und A/B-Tests. Wir betrachten solche Ansätze jedoch als fortgeschrittene Ansätze.

1.3.3.3 Zusammenarbeit zwischen Konstruktion und Realisierung (3)

Wie bei der Zusammenarbeit zwischen Design und Realisierung muss auch bei der Zusammenarbeit zwischen Konstruktion und Realisierung nach der Art des zu realisierenden Elements unterschieden werden. Bevor wir die speziellen Eigenschaften von Software und physischen Teilen im Detail diskutieren, wollen wir zunächst die allgemeinen Aspekte der Zusammenarbeit zwischen Konstruktion und Realisierung betrachten:

- *Klärung technischer Details für die Realisierung:* Die Konstruktion muss die technischen Details der digitalen Lösung bereitstellen, die für die Umsetzung der Lösung erforderlich sind. Gleichzeitig hat die Realisierung die Aufgabe, auf Ungenauigkeiten und Lücken im Realisierungskonzept der Lösung hinzuweisen, damit die Konstruktion diese korrigieren und verbessern kann.

- *Qualitätssicherung der zugrundeliegenden Form, Funktion und Qualität:* Die zugrundeliegende Form, Funktion und Qualität bilden das Fundament der digitalen Lösung und liegen in der Verantwortung der Konstruktion. Die Konstruktion ist daher dafür verantwortlich, dass die zugrundeliegende Form, Funktion und Qualität der digitalen Lösung entsprechend dem Realisierungskonzept realisiert werden. Der konkrete Ansatz zur Erfüllung dieser Verantwortung ist abhängig von dem jeweiligen Bauprozess.

Gute Beispiele für das Zusammenwirken von Konstruktion und Realisierung im Sinne der Qualitätssicherung sind automatisierte Tests (z. B. Unit-Tests) einzelner Elemente der digitalen Lösung und statische Analysen des Quellcodes (z. B. SonarQube) zur Sicherstellung eines gewissen Qualitätsniveaus im Quellcode. Auch hier sollte der detaillierte Ansatz in einem eigenen Evaluationskonzept festgehalten werden.

Die Konstruktion eines physischen Geräts endet, gemeinsam mit dem Designprozess, mit einem Modell des Produkts, das in einem Massenproduktionsprozess reproduziert werden soll. Die Konstruktion eines Softwareelements ist eine fortlaufende Aktivität zusammen mit dem Design während des gesamten Realisierungsprozesses.

Konstruktion und Realisierung in der Softwareentwicklung können von verzögerten Entscheidungen profitieren

Wichtige technische Entscheidungen für ein Softwareelement müssen getroffen werden, bevor die eigentliche Realisierung beginnt (siehe Abschnitt 1.3.1). Dennoch kann die Zusammenarbeit zwischen Konstruktion und Realisierung auch die Idee der verzögerten Entscheidungen nutzen. Wenn zum Beispiel eine bestimmte Funktionalität die Auswahl einer bestehenden Open-Source-Komponente erfordert und die Entscheidung über diese Komponente Auswirkungen auf andere Teile der Lösung hat, kann diese Entscheidung bis kurz vor Beginn der eigentlichen Implementierung dieser Komponente verschoben werden. Der Vorteil dieser verzögerten Entscheidung ist ein zweifacher: Zum einen werden die Details der jeweiligen Funktion (z. B. Datenstrukturen oder Schnittstellen zu anderen Komponenten) besser verstanden, was zu einer fundierteren Entscheidung über die Komponente führt; zum anderen können im Laufe des Bauprozesses neue Komponenten auftauchen, die bei der Planung der digitalen Lösung noch nicht verfügbar waren. Eine solche Situation ist nicht ungewöhnlich, da die Software-Community kontinuierlich an der Weiterentwicklung ihrer Technologien arbeitet.

Bei einem physischen Gerät ist der Ansatz der Zusammenarbeit ähnlich wie die Zusammenarbeit zwischen Design und Realisierung. Auch hier müssen alle relevanten Entscheidungen und Qualitätssicherungsmaßnahmen für das Produkt getroffen werden, bevor die Massenproduktion beginnt. Daher stützt sich die Zusammenarbeit zwischen Konstruktion und Realisierung auch auf Prototypen (siehe Abschnitt 2.3). Die Konstruktion nutzt diese Prototypen jedoch, um an der zugrundeliegenden Form, Funktion und Qualität der digitalen Lösung zu arbeiten. Anhand eines Prototyps kann zum Beispiel beurteilt werden, ob die technischen Komponenten in das Gehäuse des Geräts passen.

1.3.3.4 Zusammenarbeit zwischen Design, Konstruktion, Realisierung und Management (4)

Im Folgenden wird kurz auf die Herausforderungen bei der Steuerung des Bauprozesses eingegangen. Diese Einführung soll vor allem ein Bewusstsein für die Herausforderungen schaffen und ist nicht als Leitfaden für Planung und Steuerung gedacht.

In den drei vorangegangenen Unterabschnitten wurden verschiedene Paare von Überschneidungen zwischen Design, Konstruktion und Realisierung aufgezeigt. Diese Überschneidungen konzentrieren sich vor allem auf die inhaltlichen Details des Designs, der Konstruktion und der Realisierung von Form, Funktion und Qualität der digitalen Lösung, einschließlich ihrer Bewertung von Form, Funktion und Qualität. Das Management und die Planung des Bauprozesses erfordern die Zusammenarbeit zwischen allen drei Tätigkeitsbereichen, da die Kompetenzen aller drei Bereiche erforderlich sind, um eine kohärente Perspektive für den Bauprozess zu schaffen.

Iteratives und inkrementelles Arbeiten ist der Kern der Zusammenarbeit

Die Definition und Abstimmung der Details des Bauprozesses sind keineswegs Top-Down-Aktivitäten vom Design über die Konstruktion bis zur Realisierung. Ein solcher Ansatz führt in der Regel zu suboptimalen Ergebnissen. Insbesondere bei der Entwicklung von Software hat die agile Entwicklungsbewegung gezeigt, dass ein iterativer und inkrementeller Managementansatz viel besser geeignet ist.

Ein solch iterativer und inkrementeller Ansatz erfordert eine enge Zusammenarbeit zwischen den Fachleuten aller drei Tätigkeitsbereiche, da jeder einzelne Tätigkeitsbereich einen bestimmten Input für einen anderen Bereich liefert. Die schlechte Nachricht ist, dass es gegenseitige Abhängigkeiten zwischen den Tätigkeitsbereichen gibt und dass diese komplexen Abhängigkeiten es unmöglich machen, einen allgemeinen Planungsansatz für den Bauprozess einer digitalen Lösung zu definieren. Im Folgenden werden diese Abhängigkeiten anhand einiger Beispiele veranschaulicht.

- Das Aussehen und Verhalten des User Interface (als Teil des Designs) ist die Grundlage für die Auswahl der technischen Komponenten zur Umsetzung des User Interface (als Teil der Konstruktion). Eine bestimmte Technologie kann Funktionen für die Implementierung des User Interface bieten, an die das Design nicht gedacht hat (Realisierung). Wenn dieses Potenzial nicht genutzt wird, entsteht eine digitale Lösung, die nicht das gesamte Potenzial der Technologie ausschöpft. Ein Plan für den Bau einer digitalen Lösung mit einer großen Anzahl von User Interfaces sollte einen umfassenden Austausch zwischen Fachleuten für Design, Konstruktion und Realisierung vorsehen, um das gesamte Potenzial der Technologie auszuschöpfen.
- Die Entwicklungsumgebung (die Umgebung zum Implementieren und Testen eines Elements einer digitalen Lösung) kann als reiner Realisierungsaspekt betrachtet werden. Moderne Entwicklungsumgebungen bieten jedoch wichtige Funktionen, die auch für Design und Konstruktion nützlich sind, z. B. Werkzeuge zur Modellierung von Datenstrukturen. Bestimmte Entwicklungsumgebungen unterstützen insbesondere das Rapid Prototyping, d. h., detaillierte Designaspekte einer digitalen Lösung können während der Realisierung definiert und sofort gemeinsam mit den Stakeholdern validiert und verbessert werden.
- Die Planung der Evaluierung bestimmter Aspekte einer digitalen Lösung (z. B. die Benutzerakzeptanz der definierten Prozesse) muss mit anderen Aktivitäten abgestimmt werden. Zum einen muss der Teil der Lösung, der bewertet werden soll, richtig umgesetzt werden. Zum anderen muss die richtige Umgebung für die Evaluierung (z. B. ein Usability-Labor) eingerichtet werden, wozu auch die Einladung potenzieller Testkandidaten gehört. Die Planung solcher Evaluationsaktivitäten erfordert eine eher langfristige Planung, die mit anderen Aktivitäten in Konflikt geraten kann.

- Der Ablauf der Implementierung einer digitalen Lösung kann durch verschiedene Aspekte bestimmt werden. Ein typischer Ansatz sind *User Storys* (eine Beschreibung der Lösung, die einen Nutzen für einen Benutzer schafft, siehe Abschnitt 5.3.4.2) oder ein *Minimal Viable Product* (MVP, siehe Abschnitt 5.4). Solche Implementierungsansätze betreffen in der Regel die gesamte Struktur einer digitalen Lösung, da eine Funktionalität durch die gesamte Lösung hindurch realisiert werden muss. Ein technisch getriebener Ansatz implementiert zunächst alle wichtigen Grundfunktionen und beginnt dann mit der Implementierung der für den Benutzer relevanten Funktionen. Ein solcher Ansatz erfordert mehr Zeit, bis ein Benutzer ein Feedback geben kann.
- Die Definition von Zeitplänen wird häufig von den Realisierungskosten und der Markteinführungszeit bestimmt. Ein solcher Ansatz vernachlässigt in der Regel den Aufwand für Design und Konstruktion, einschließlich der Evaluierung der Design- und Realisierungskonzepte. Eine bessere Aufwandsabschätzung und Budgetzuweisung kann erreicht werden, wenn Design und Konstruktion ausdrücklich im Budget berücksichtigt werden. Vor allem bei Projekten mit festem Budget ist die Design- und Konstruktionskompetenz von großer Bedeutung, um die bestmögliche digitale Lösung für das verfügbare Budget zu erhalten.

1.3.4 Management des Bauprozesses auf Lösungs-, System- und Elementebene

Der Bauprozess für eine digitale Lösung kann entsprechend der in Abbildung 4 dargestellten hierarchischen Struktur der Abstraktionsebenen strukturiert werden:

- *Lösungsebene*: Verstehen und Gestalten der Ziele und des Geschäftsmodells (falls zutreffend) in Bezug auf Kunden, Stakeholder, bestehende Systeme und das digitale System als Blackbox
- *Systemebene*: Verständnis und Definition des gesamten digitalen Systems in Bezug auf die Ziele der digitalen Lösung als Zusammenspiel des bestehenden Systems, der Menschen als Benutzer und der zu entwickelnden Elemente
- *Elementebene*: Verstehen, Definieren und Realisieren eines bestimmten Elements des digitalen Systems, das die digitale Lösung in Bezug auf Benutzer, bestehende Systeme und andere Elemente umsetzt

Dieses Verständnis von Lösung und System stellt einen wichtigen terminologischen Unterschied zu anderen Bereichen dar. Einige Disziplinen betrachten die Kunden, die Benutzer und die bestehenden Systeme als Systemkontext und nur die Elemente als System. Dieses Verständnis eignete sich für Digitization (Überführung bestehender Dinge in die digitale Welt). Digitalisierung und digitale Transformation erfordern jedoch das oben skizzierte breitere Verständnis, da Neues entsteht und die Lösung und das System parallel gestaltet werden. Die YPRC-Fallstudie ist ein Beispiel dafür. Während der eigentlichen Entwicklung von YPRC kamen neue Ideen für den Verkauf zusätzlicher Dienstleistungen auf, die Teil des Geschäftsmodells wurden.

Die Trennung zwischen Lösungs-, System- und Elementebene ist für das Verständnis und die Verwaltung des Bauprozesses einer digitalen Lösung von großer Bedeutung. Selbst sehr einfache digitale Lösungen bestehen in der Regel aus mehr als einem Element. Sobald eine digitale Lösung nicht nur auf einem einzelnen Gerät läuft, sind zwei Elemente beteiligt: der Front-End-Teil und der Back-End-Teil der Lösung. Die YPRC-Fallstudie ist ebenfalls eine einfache digitale Lösung und besteht bereits aus drei Elementen.

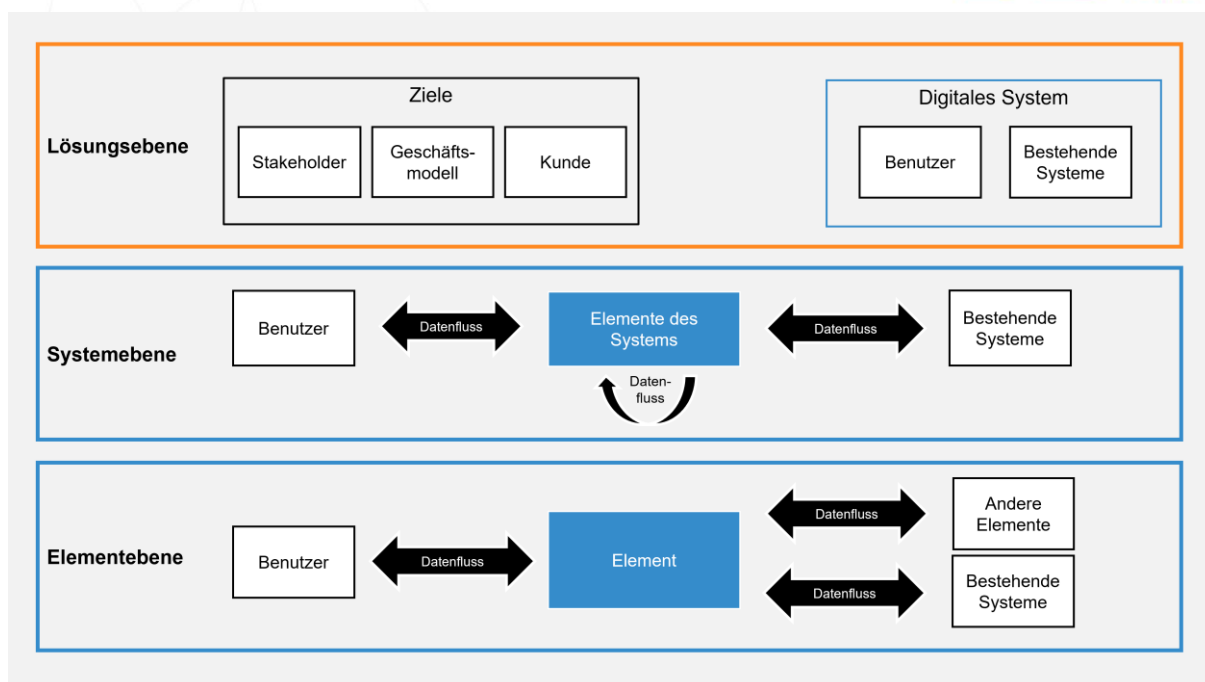


Abbildung 4 - Überblick über die Abstraktionsebenen Lösung, System und Element

Die drei Abstraktionsebenen sollten nicht als drei aufeinander folgende Prozessschritte verstanden werden. Sie dienen dazu, die Sicht auf eine digitale Lösung zu strukturieren. Mit einem ersten Verständnis der angestrebten Zukunft beginnt ein iterativer Prozess, der auf allen drei Abstraktionsebenen parallel abläuft.

1.3.4.1 Die Fähigkeit, den Bauprozess auf der Lösungs- und Systemebene zu verstehen

Der Bauprozess für eine digitale Lösung beginnt immer auf der Lösungsebene und mit der angestrebten Zukunft. Auf dieser Ebene werden die wichtigsten Entscheidungen getroffen:

- Die Hauptziele einer digitalen Lösung, einschließlich der Kunden
- Das Wertversprechen (Value Proposition) und das Geschäftsmodell hinter der digitalen Lösung (falls zutreffend)

Neben der Lösungsebene ist die Systemebene wichtig für die Festlegung von Form, Funktion und Qualität des digitalen Systems insgesamt, um die festgelegten Ziele zu erzielen. Mit einem angemessenen Grundverständnis der Lösungs- und Systemebene können die folgenden Aktivitäten durchgeführt werden:

- Planung der erforderlichen Mitarbeiter, um die digitale Lösung zu bauen
- Planung der langfristigen Entwicklung einer digitalen Lösung
- Planung und Steuerung des nachgelagerten Bauprozesses für die Elemente
- Fortlaufende Abstimmung der Bauprozesse für die Elemente untereinander und mit der Systemebene
- Treffen grundlegender Entscheidungen über die Technologien, die zur Realisierung der digitalen Lösung eingesetzt werden sollen
- Evaluierung und Weiterentwicklung des Hauptziels der digitalen Lösung
- Evaluierung und Weiterentwicklung der Gesamtidee der digitalen Lösung
- Evaluierung und Weiterentwicklung des Geschäftsmodells (falls zutreffend)

Planung der erforderlichen Mitarbeiter, um die digitale Lösung zu bauen

Die Hauptideen einer digitalen Lösung sind ein Ausgangspunkt für die Planung der Fähigkeiten, die zum Aufbau der digitalen Lösung erforderlich sind. Je nach Art der beteiligten Elemente sind unterschiedliche Fähigkeiten und eine spezifische Teamvielfalt erforderlich, um ein effektives Umsetzungsteam zu bilden⁵. Vor allem, wenn die digitale Lösung aus einem oder mehreren speziellen Geräten besteht, sind Fähigkeiten in den Bereichen Industriedesign, Produktionstechnik und Entwicklung eingebetteter Systeme erforderlich, um das spezielle Gerät zu bauen. Darüber hinaus erfordert der Bau eines speziellen Geräts andere Managementfähigkeiten als eine rein softwarebasierte digitale Lösung. Auch für den Softwareteil bietet die Systemebene eine erste Orientierungshilfe für die erforderlichen Fähigkeiten, insbesondere wenn eine spezielle Software benötigt wird.

Beispiel YPRC. Für das YPRC-Gehäuse benötigen wir auf jeden Fall Kenntnisse in Industriedesign und Produktionstechnik, wenn wir die Smartwatch selbst herstellen wollen (der Kauf eines White-Label-Produkts ist ebenfalls eine Option). Für den Software-Teil brauchen wir Fachleute für die Entwicklung von Smartphone-Apps, und für den Coaching-Teil mit künstlicher Intelligenz brauchen wir Fachleute für die Entwicklung von Software mit künstlicher Intelligenz.

Planung der langfristigen Entwicklung einer digitalen Lösung

Digitale Lösungen können sich im Laufe der Zeit weiterentwickeln, z. B. werden neue Funktionen angeboten oder bestehende Funktionen verbessert. Das Verständnis einer digitalen Lösung auf Systemebene kann als Ausgangspunkt für die Planung dieser Entwicklung betrachtet werden.

Beispiel YPRC. Die Beschreibung der YPRC-Fallstudie legt eine natürliche Reihenfolge für diese Entwicklung nahe: Zunächst wird die Smartwatch zusammen mit der App und dem kostenlosen Portal zur Datenspeicherung angeboten. Sobald es genügend Benutzer der App gibt, startet die Entwicklung des künstlichen Intelligenz-Coachings und der Verkauf dieses Dienstes. Zu einem späteren Zeitpunkt beginnt die Entwicklung des persönlichen Remote-Coachings und der Verkauf dieser Dienstleistung. Dieser einfache, längerfristige Plan ist bereits ein erster Bauplan für die digitale Lösung. Dies ist jedoch nicht der einzige mögliche Ansatz für den Aufbau von YPRC. In Abschnitt 2.1 werden wir auf diesen Plan zurückkommen und mögliche Alternativen diskutieren.

Planung und Steuerung des nachgelagerten Bauprozesses für die Elemente

Eine digitale Lösung besteht in der Regel aus mehr als einem Element, die parallel gebaut werden können und sollten. Ein solch paralleler Bauprozess erfordert Planung und Steuerung. Die im vorangegangenen Abschnitt skizzierte langfristige Entwicklung zeigt, dass wir mit dem Smartphone einschließlich der App und dem Portal zur Datenspeicherung beginnen werden. Der Bauprozess für alle drei Elemente kann parallel beginnen und kann auch aus der Perspektive der Systemebene verwaltet werden. Das wichtigste Instrument, um parallele Bauprozesse zu ermöglichen, ist eine genaue Definition der Funktionalitäten und technischen Schnittstellen zwischen den verschiedenen Elementen.

⁵ Siehe Abschnitt 4.3 für weitere Einzelheiten zum People Management.

Beispiel YPRC. Für den YPRC-Fall müssen wir die Schnittstelle zwischen der Smartwatch und der App sowie die Schnittstelle zwischen der App und dem Portal für die Datenspeicherung definieren. Die Definition dieser Schnittstellen ist eine gemeinsame Aufgabe von Design und Konstruktion. Eine klare Definition der Schnittstellen zwischen den Elementen ermöglicht einen strukturierten Bauprozess für jedes Element, da sich jedes Element auf einen klaren Satz von Daten und Funktionen verlassen kann, die es von anderen Elementen erwarten kann.

Fortlaufende Abstimmung der Bauprozesse für die Elemente untereinander und mit der Systemebene

Der Aufbau einer digitalen Lösung ist kein Top-Down-Prozess. Stattdessen findet ein umfassender Informationsaustausch zwischen der Systemebene und der Elementebene statt, da der Bau der einzelnen Elemente natürlich zu neuen Erkenntnissen und Ideen über die digitale Lösung führt. Die wichtigsten Punkte für Erkenntnisse sind die Schnittstellen zwischen den Elementen. Jede Änderung der Schnittstelle muss dem betroffenen Element unverzüglich mitgeteilt werden, damit der Bauprozess für diese Elemente die Änderung berücksichtigen kann. Die Verfügbarkeit neuer Technologien ist eine weitere Quelle für Änderungen.

Beispiel YPRC. Die Verfügbarkeit der neuen Technologie der künstlichen Intelligenz in der YPRC-Fallstudie könnte das Angebot neuer Coaching-Dienste ermöglichen. Diese Erkenntnisse müssen zwischen den Bauprozessen ausgetauscht werden, um die neue Technologie für die jeweiligen Elemente nutzbar zu machen.

Treffen grundlegender Entscheidungen über die Technologien, die zur Realisierung der digitalen Lösung eingesetzt werden sollen

Mit einem Verständnis der Lösungs- und Systemebene ist es möglich, verschiedene Technologien für die Realisierung der digitalen Lösung zu untersuchen.

Beispiel YPRC. Die verschiedenen Elemente der YPRC-Fallstudie können mit verschiedenen Technologien (z. B. Java für das Portal) realisiert werden. Wenn innovative Technologien in Betracht gezogen werden, ist ein frühzeitiges Verständnis der Systemebene sehr nützlich, um die Anwendbarkeit der innovativen Technologie zu diskutieren. So wird zum Beispiel künstliche Intelligenz (KI) für die Realisierung eines KI-basierten Coachs in Betracht gezogen. Eine KI zu entwickeln und zu trainieren, die in der Lage ist, Läufer während des Trainings zu coachen, ist eine große Herausforderung. Die wichtigsten Herausforderungen (z. B. Trainingsdaten, ausreichende Rechenleistung) für einen solchen Ansatz können bereits mit einem ersten Verständnis der Systemebene diskutiert werden.

Evaluierung und Weiterentwicklung des Hauptziels der digitalen Lösung

Wenn eine digitale Lösung vom Markt getrieben und nicht für eine geschlossene Organisation geschaffen wird, ist die Akzeptanz der digitalen Lösung auf dem Markt sehr wichtig.

Beispiel YPRC. Das Hauptziel von YPRC ist es, eine erschwingliche „Rundum-Coaching-Lösung für Einsteiger im Langstreckenlauf“ anzubieten. Auf den ersten Blick scheint dieses Ziel vernünftig zu sein, da Langstreckenläufe ein beliebter Sport sind. Dennoch kann eine Evaluierung des definierten Ziels - beispielsweise durch eine Markt- oder Kundenbefragung - von Vorteil sein, um die Akzeptanz dieser Idee auf dem Markt besser zu verstehen. Eine typische Frage kann lauten: Gibt es auf dem Markt eine ausreichende Anzahl potenzieller Kunden, die eine solche Lösung kaufen würden?

In Abschnitt 2.1 finden Sie weitere Einzelheiten zu möglichen Ansätzen für die Evaluierung und Weiterentwicklung der Hauptziele.

Evaluierung und Weiterentwicklung der Gesamtidee der digitalen Lösung

Die Lösungsebene fasst bereits die Grundidee einer digitalen Lösung zusammen. Diese Grundidee kann auch zu Bewertungszwecken verwendet werden.

Beispiel YPRC. Die Grundidee von YPRC besteht aus drei Elementen: dem Verkauf einer Smartwatch zusammen mit einer kostenlosen Smartphone-App und einem Portal zur Speicherung von Gesundheitsdaten. Über die App kann der Besitzer der Smartwatch zusätzliche Coaching-Dienste erwerben. Selbst auf dieser abstrakten Ebene kann die Beschreibung dieser Ideen genutzt werden, um mit potenziellen Kunden über YPRC zu diskutieren und weitere Erkenntnisse über die Akzeptanz dieser Lösung zu gewinnen.

In Abschnitt 2.1 werden weitere Einzelheiten zu möglichen Ansätzen für die Validierung und Weiterentwicklung der gesamten Systemebene einer digitalen Lösung beschrieben.

Evaluierung und Weiterentwicklung des Geschäftsmodells

Mit einem ersten Verständnis der Lösungs- und Systemebene ist es möglich, erste Geschäftsmodellideen zu bewerten. In Abschnitt 4.2 gehen wir näher auf die verschiedenen Arten von Geschäftsmodellen für digitale Lösungen ein.

Beispiel YPRC. In der Beschreibung der Grundidee von YPRC im vorherigen Abschnitt wurden bereits bestimmte Aspekte des Geschäftsmodells von YPRC (Verkauf der Smartwatch, Verkauf von Premium-Diensten) und die Kostentreiber von YPRC (Kosten für die Entwicklung und Massenproduktion einer Smartwatch, Kosten für die Entwicklung und Pflege der Smartphone-App und des Portals sowie die Betriebskosten für das Portal) skizziert. Diese Informationen können genutzt werden, um einen ersten Entwurf des Geschäftsmodells von YPRC auszuarbeiten und zu bewerten, ob das Geschäftsmodell ein Erfolg werden kann.

All diese Aktivitäten sind nur möglich, wenn ein angemessenes Verständnis der Lösungs- und Systemebene der digitalen Lösung vorhanden ist. In Abschnitt 2.2.2 werden praktische Ansätze zur Erstellung von Designkonzepten für die Lösungs- und Systemebene als Ausgangspunkt für die Dokumentation und als Grundlage für die Verwaltung des Bauprozesses auf der Lösungs- und Systemebene erörtert.

1.3.4.2 Der Unterschied zwischen einem konzeptorientierten und einem realisierungsorientierten Bauprozess

Wie bei den Aktivitäten findet der Bauprozess auf der Systemebene und der Elementebene parallel statt (siehe Abbildung 5). Im vorigen Abschnitt haben wir gezeigt, dass der Prozess auf der Lösungs- und Systemebene beginnt. Wenn die Lösungs- und Systemebene richtig verstanden wurde, kann mit dem Design und der Konstruktion der einzelnen Elemente begonnen werden, allerdings in enger Zusammenarbeit mit dem Design und der Konstruktion der Lösungs- und Systemebene.

Wenn die Realisierung des ersten Elements einer digitalen Lösung beginnt, muss ein grundlegender Übergang im Management (des Bauprozesses) stattfinden. Mit dem Beginn der Realisierung eines Elements wandelt sich der Bauprozess von einem konzeptgesteuerten Prozess (d.h. reine Konzeptarbeit) in einen realisierungsgesteuerten Prozess (d.h. getrieben durch Umsetzungsarbeit). Ein Prozess auf Basis von Konzeptarbeit, der sich auf Design und

Konstruktion konzentriert, ist kostengünstig und schnell. Neue Erkenntnisse können leicht und mit geringem Aufwand in das Konzept eingearbeitet werden. Ein umsetzungsorientierter Prozess ist in der Regel teuer und langsam. Realisierungsteams (z. B. Softwareentwicklungsteams, vgl. [SeRP2017]) benötigen ständigen Input, um produktiv zu bleiben. Die Einarbeitung neuer Erkenntnisse und die Korrektur von schwerwiegenden Fehlern ist aufwendig. Bereits gebaute Teile müssen eventuell mit Mehraufwand geändert werden.

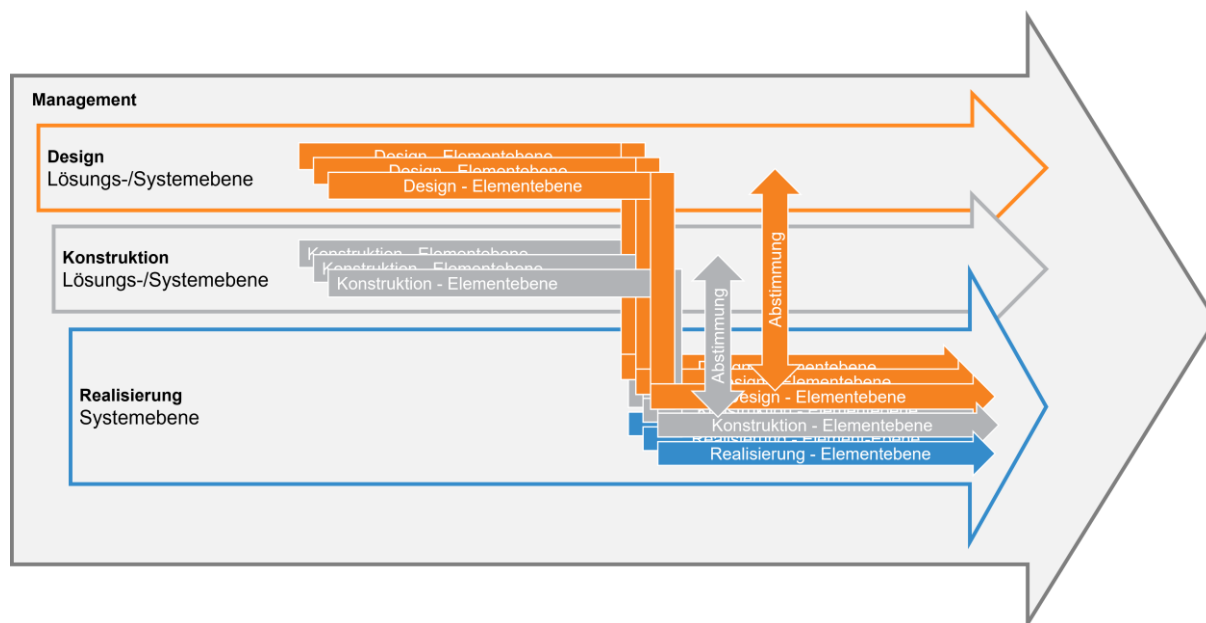


Abbildung 5 - Aktivitäten des Bauprozesses laufen parallel auf System- und Elementebene

Der Unterschied zwischen konzeptgetriebenen und realisierungsgetriebenen Prozessen ist kein Argument dafür, so lange wie möglich in der Konzeptarbeit zu verbleiben, um möglichst umfassende und evaluierte Konzepte zu erarbeiten. Sie ist als Warnung gedacht, dass der Beginn der Realisierung kostspielig ist und dass die wichtigen und kostspieligen Details geklärt werden müssen, bevor die Realisierung beginnt. Das Erkennen dieser wichtigen Details erfordert Erfahrung und Ausbildung.

1.4 Kompetenzprofil eines Digital Design Professionals

Ein Digital Design Professional (DDP) ist eine Person, die als fachkundig auf dem Gebiet des Digital Designs gilt. Dieses Handbuch beschreibt die Fähigkeiten, die für ein grundlegendes Verständnis des Digital Designs erforderlich sind. Ein Überblick über diese Fähigkeiten wird im folgenden Unterabschnitt gegeben. Darüber hinaus ist es wichtig zu erkennen, dass der DDP keine neue Rolle ist. Wir gehen in Abschnitt 1.4.2 näher darauf ein.

1.4.1 Der Digital Design Professional als Ausbildungsprogramm

Die Struktur dieses Handbuchs folgt der Idee des Pi-förmigen Kompetenzprofils von Digital Design [Bitk2017], das durch den griechischen Buchstaben π als Symbol inspiriert wurde, und wie in Abbildung 6 aufgebaut ist.

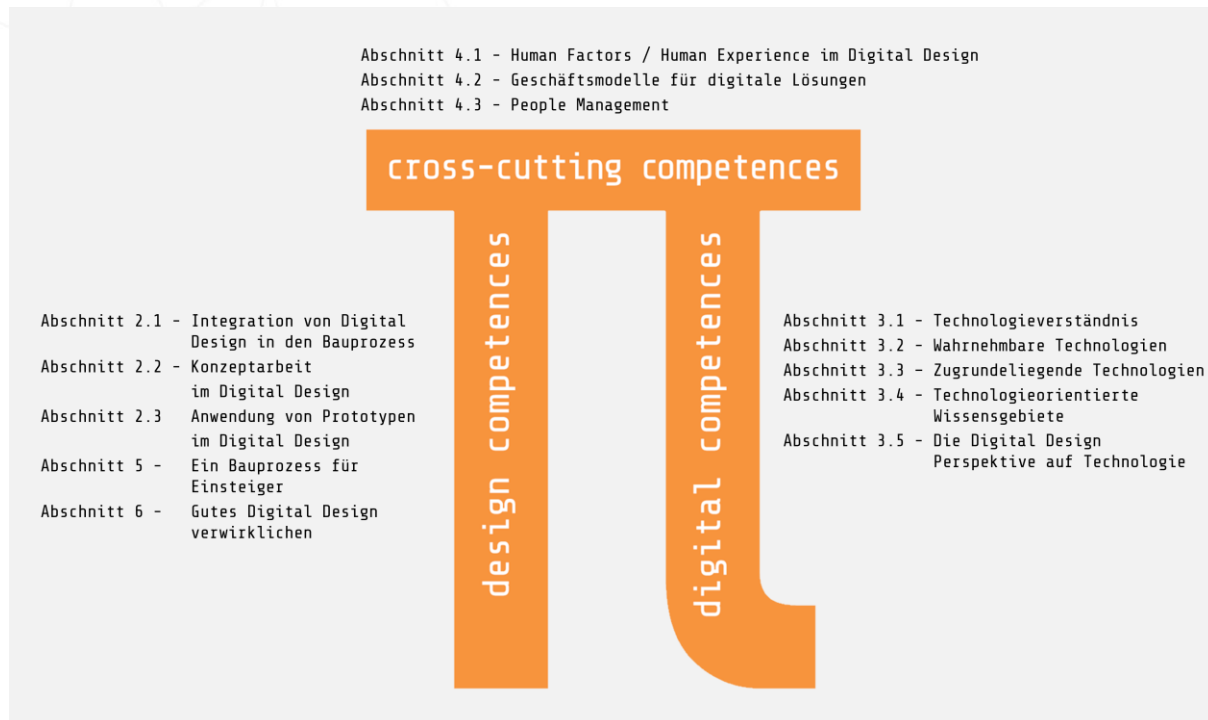


Abbildung 6 - Das π-förmige Kompetenzprofil eines DDP auf Foundation Level Niveau

Das linke Bein des Pi stellt die Design-Kompetenz dar. Im Foundation Level sind die folgenden Aspekte wichtig:

- Integration von Digital Design in den Bauprozess (2.1)
- Konzeptarbeit im Digital Design (2.2)
- Verwendung von Prototypen im Digital Design (2.3)

Das rechte Bein steht für das Verständnis von Digital als Material. Im Foundation Level sind die folgenden Aspekte wichtig:

- Technologieverständnis (3.1)
- Wahrnehmbare Technologien (3.2)
- Zugrundeliegende Technologien (3.3)
- Technologieorientierte Wissensgebiete (3.4)
- Die Digital Design Perspektive auf Technologie (3.5)

Der obere Teil des Pi stellt querschnittliche Aspekte des Digital Designs dar. Im Foundation Level sind die folgenden Aspekte wichtig:

- Human Factors / Human Experience im Digital Design (4.1)
- Geschäftsmodelle für digitale Lösungen (4.2)
- People Management (4.3)

Zur Anwendung dieses breiten Kompetenzprofils wird in Kapitel 5 ein Bauprozess vorgestellt, der für Einsteiger ins Digital Design gedacht ist. Dieser Prozess bietet eine Reihe von konkreten und pragmatischen Technologien, die für den Bau digitaler Lösungen mit mittlerer Komplexität geeignet sind.

Das Kompetenzprofil eines DDP wird schlussendlich durch die zehn Prinzipien für gutes Digital Design abgerundet. Im folgenden Unterabschnitt werden diese Prinzipien vorgestellt und in

Kapitel 6 wird erörtert, wie die Methoden und Technologien aus diesem Handbuch dazu beitragen, die zehn Prinzipien des guten Digital Designs zu erreichen.

1.4.2 Zehn Prinzipien für gutes Digital Design

Im Allgemeinen wird ein Berufsbild durch seine Methoden, Techniken und Werte definiert. Die im Digital Design Manifest [LBGH2018] vorgestellten zehn Prinzipien für gutes Digital Design definieren die Prinzipien, die die Werte des Digital Designs und die Grundhaltung der Menschen, die im Bereich des Digital Designs arbeiten, bestimmen.

In den folgenden Unterkapiteln werden die zehn Prinzipien gemäß dem Digital Design Manifest [LBGH2018] im Detail vorgestellt. Denken Sie daran, dass die zehn Prinzipien keine Checkliste sind; sie beschreiben vielmehr eine Haltung gegenüber dem digitalen Material und der Gestaltung guter digitaler Lösungen. Sie beziehen sich immer auf gutes Digital Design und meinen damit sowohl den Designprozess als auch das Ergebnis. Dies ist wichtig, weil wir der Meinung sind, dass der Prozess und das Ergebnis untrennbar miteinander verbunden sind.

P1 - Gutes Digital Design ist nützlich und gebrauchbar

Digitale Technologien gehören angeblich zu den leistungsfähigsten Technologien, die je von der Menschheit erfunden wurden. Gutes Digital Design nutzt diese Technologien, um Vorteile und Mehrwerte zu schaffen.

Gleichzeitig handelt es sich bei den digitalen Technologien jedoch um die vermutlich komplexesten Technologien, die jemals erfunden wurden. Diese Komplexität darf nicht zu einem Problem für den Benutzer werden. Deshalb ist gutes Digital Design so konzipiert, dass die daraus resultierenden digitalen Lösungen einfach, mit Freude und einer guten Erfahrung für den Benutzer verwendet werden können.

P2 - Gutes Digital Design ist elegant und ästhetisch

So wie gut gestaltete (analoge) Produkte oder Gebäude ihre eigene Eleganz und Ästhetik haben, so sind auch gute Digital Design-Lösungen elegant. Diese Eleganz bezieht sich auf die Erwartungen der Benutzer an die wahrnehmbare Form (z. B. Elemente, wie ästhetische User Interfaces oder elegant gestaltete Geräte für Benutzer).

Aber auch *unsichtbare Elemente* sind elegant. Zusammen mit der zugrundeliegenden Form, Funktion und Qualität einer digitalen Lösung können Algorithmen, Datenstrukturen und Softwarearchitekturen ihre eigene Eleganz entwickeln. Die unsichtbaren Elemente vermitteln dies z. B. durch ihre Einfachheit, effiziente Verarbeitung, Wiederverwendbarkeit, gute Wartbarkeit oder intelligente Nutzung der technischen Möglichkeiten der Digitalisierung. Sie fördern damit ihren eigenen Nutzen für die Entwicklung digitaler Produkte, Systeme oder Dienstleistungen.

P3 - Gutes Digital Design ist evolutionär

Keine digitale Lösung ist von Anfang an perfekt - alle digitalen Lösungen werden ständig weiterentwickelt. Darüber hinaus sind alle digitalen Lösungen ständig äußeren Einflüssen ausgesetzt, was zwangsläufig zur Notwendigkeit von Veränderungen führt. Gutes Digital Design ist so konzipiert, dass es eine lange Lebensdauer hat. Änderungen und Weiterentwicklungen werden so einfach wie möglich gemacht oder nicht unnötig behindert.

P4 - Gutes Digital Design ist explorativ

Zu glauben, dass man immer im Voraus weiß, welche Lösung funktionieren wird, ist naiv. Dies gilt insbesondere für den innovativen Bereich der Digitalisierung. Gutes Digital Design ist explorativ - es bietet Benutzern, Designern und Entwicklern verschiedene Möglichkeiten, ihre Ziele zu erreichen. Es zieht Schlüsse aus dem Verhalten der Benutzer, um den besten Weg zu erkennen und diesen Weg weiter zu entwickeln.

P5 - Gutes Digital Design nimmt den ganzen Menschen in den Fokus

Das benutzerzentrierte Design ist ein wichtiges Gestaltungsprinzip, greift aber zu kurz, da der Benutzer auch eine Person innerhalb der Gesellschaft als Ganzes ist. Die durch die Digitalisierung zu erwartenden Umwälzungen bedeuten daher, dass der Mensch in seiner Gesamtheit, samt seinem Umfeld und der Gesellschaft, in den Mittelpunkt gerückt werden muss.

So sind viele voll digitalisierte Arbeitsplätze heute oft an den Schreibtisch gebunden und führen bei vielen Büroangestellten zu Bewegungsmangel. Gutes Digital Design kann digitale Arbeitsplätze so gestalten, dass Bewegungselemente zu einem integralen Bestandteil der Arbeit werden und somit die Gesundheit der Mitarbeiter fördern.

P6 - Gutes Digital Design antizipiert die Auswirkungen seiner Ergebnisse

Die Digitalisierung findet nicht im luftleeren Raum statt, sondern hat Auswirkungen auf alle Menschen und die Gesellschaft insgesamt. Aktuelle Entwicklungen zeigen, dass nicht alle Effekte der Digitalisierung wirklich erwünscht sind und sogar zu unerwünschten Nebenwirkungen führen können. Einige dieser Auswirkungen werden erst im Laufe der Zeit sichtbar, wenn eine große Anzahl von Kunden eine Lösung angenommen hat. Manchmal stellt sich heraus, dass eine Lösung auf eine zunächst unbeabsichtigte Weise oder von Kunden genutzt wird, für die sie nicht konzipiert wurde. Dies ergibt sich aus den Interaktionen zwischen Kunden und anderen sozialen Effekten innerhalb der Gesellschaft.

Deshalb antizipiert gutes Digital Design die Auswirkungen seiner Ergebnisse und wägt die Vorteile einer Lösung gegen die entstehenden Nachteile ab.

P7 - Gutes Digital Design achtet den Datenschutz und die Datensicherheit

Welche Daten gespeichert werden und wie sie verarbeitet werden, ist eine Frage des Digital Designs. Deshalb beginnen Datenschutz und Datensicherheit bereits im Digital Design. Gutes Digital Design berücksichtigt von Anfang an die geltenden Datenschutzgesetze und geht sparsam mit Daten um, das heißt, es werden nur Daten verwendet, die für den beabsichtigten Zweck unerlässlich sind.

Bei gutem Digital Design werden sensible und kritische Daten entsprechend ihrer Bedeutung durch sorgfältiges Design und den Einsatz aktueller Technologien unter Berücksichtigung der Risiken des Datendiebstahls besonders geschützt.

P8 - Gutes Digital Design ist nachhaltig und schafft Nachhaltigkeit

Digitale Technologie verbraucht jeden Tag viel Energie. Die Herstellung von Endgeräten wie Smartphones, Tablets oder Smartwatches verbraucht ebenfalls viele Ressourcen. Durch intelligentes Design können Energie und wertvolle Ressourcen eingespart werden. Umgekehrt können intelligente digitale Lösungen Nachhaltigkeit schaffen. So können beispielsweise digitale Kommunikationsmittel die Notwendigkeit von Reisen verringern und damit Energie sparen.

Intelligente digitale Steuerungssysteme sparen schon heute in vielen Bereichen der Industrie und des privaten Lebens Energie.

Gutes Digital Design muss einen Beitrag zur Nachhaltigkeit leisten und bevorzugt daher Lösungen, die den Energie- und Ressourcenverbrauch im Verhältnis zu ihrem Nutzen minimieren. Nachhaltigkeit im Digital Design geht jedoch über die ökologische Dimension hinaus und muss als eine Reihe von Dimensionen verstanden werden, die auch individuelle, soziale, ökonomische und technische Aspekte umfasst (vgl. [BCDE2015]). Dazu gehört auch die Maximierung der Qualität der digitalen Lösung, um eine lange Lebensdauer zu gewährleisten. Bei der Digitalisierung gehört zur Nachhaltigkeit auch die Vorwegnahme der Entsorgung. Deshalb plant gutes Digital Design vor der Realisierung, was mit den Daten oder Endgeräten passiert, wenn die Lösungen nicht mehr genutzt werden oder die Benutzer verstorben sind.

P9 - Gutes Digital Design würdigt Analoges und Digitales in gleicher Weise

Analog und digital sind keine Gegensätze - sie beschreiben lediglich die Pole eines Spektrums. Nur weil etwas, das früher analog war, jetzt digital ist (z. B. Papierbücher im Vergleich zu E-Books), bedeutet dies nicht unbedingt, dass die digitale Lösung besser ist.

Gutes Digital Design muss sich nicht auf den digitalen Aspekt beschränken. Digitale Mittel sollten analoge Mittel nur dann ersetzen, wenn dies angemessen und konstruktiv ist. Wenn ein analoges Element gleichwertig oder sogar besser ist als das digitale Element, sollte das analoge Element verwendet werden. Das hohe Potenzial hybrider digitaler Lösungen kann nur dann für echte Innovationen genutzt werden, wenn analoge und digitale Mittel in gleicher Weise gewürdigt und berücksichtigt werden.

P10 - Gutes Digital Design setzt Digitales nur dort ein, wo es erforderlich ist

Zweifellos ist die Digitalisierung eine treibende Kraft für den Fortschritt. Aber gerade deshalb darf sie nicht zum Selbstzweck werden, denn dann würde sie ihre Glaubwürdigkeit und Innovationskraft verlieren, um das Leben aller Menschen in der Gesellschaft zu verbessern.

Gutes Digital Design setzt digitale Mittel bewusst und dort ein, wo es notwendig ist, und schafft Vorteile für das Leben der Menschen in unserer Gesellschaft.

1.4.3 Digital Design Professional ist keine neue Rolle

Der Unterschied zwischen einer Rolle und einem Berufsbild ist wichtig für das Verständnis von Digital Design:

- Eine Rolle ist eine Funktion, die eine Person in einer bestimmten Situation einnehmen kann. Eine Rolle wird durch ihre Aufgaben, Rechte, Pflichten und Verantwortlichkeiten definiert.
- Ein Berufsbild ist eine Tätigkeit, die eine spezielle Ausbildung erfordert.

Der DDP ist keine Rolle, es ist ein Trainingsprogramm für den Einstieg in das Berufsbild des Digital Design. Dieses Foundation Level Handbuch zielt darauf ab, einen breiten Überblick über Digital Design zu geben und gleichzeitig Methoden und Techniken für die praktische Arbeit zu vermitteln. Diese konkurrierenden Ziele erfordern eine Auswahl der dargestellten Inhalte und eine Vereinfachung der Inhalte. Um den Blick für das große Ganze auf Grundlagenniveau nicht zu verlieren, werden Hinweise auf weiterführende Literatur gegeben.

Während des Bauprozesses für eine digitale Lösung kann ein DDP in verschiedenen Rollen tätig werden, die mit dem Tätigkeitsbereich Gestaltung zusammenhängen (z. B. Product Owner, Business Analyst, Requirements Engineer, Usability Engineer). Aufgrund des breiten Spektrums von Digital Design erfordern bestimmte Rollen jedoch eine zusätzliche spezialisierte Ausbildung, um gutes Digital Design zu erreichen (siehe 6). Als Ausgangspunkt für diese Ausbildung werden in diesem Lehrplan Hinweise auf weiterführende Literatur gegeben.

Der Hauptvorteil der Zertifizierung als DDP besteht darin, dass sie Ihnen ein umfassendes Verständnis des Bauprozesses für digitale Lösungen vermittelt. Darüber hinaus verfügen Sie dann über eine breite Kompetenz in der Gestaltung digitaler Lösungen, einschließlich der notwendigen Material- und Querschnittskompetenzen. Dieses breite Wissen aus der DDP-Ausbildung unterstützt insbesondere Fachleute mit spezialisierter Ausbildung dabei, ihre persönlichen Stärken besser in den gesamten Bauprozess für eine digitale Lösung einzubringen.

2 Design-Kompetenz

In diesem Kapitel stellen wir die Welt der Design- (Gestaltungs-)Kompetenz für Digital Design vor.

Wir beginnen mit der Integration von Digital Design in den Bauprozess in Abschnitt 2.1. Das Design einer digitalen Lösung kann nicht verstanden werden, ohne den gesamten Bauprozess zu verstehen. Der Hauptgrund dafür ist, dass sich das Design einer digitalen Lösung über verschiedene Abstraktionsebenen (Lösung, System und Element) erstreckt und diese Abstraktionsebenen nicht mit einem Prozessmodell übereinstimmen. Ganz im Gegenteil: Aufgrund der Komplexität einer digitalen Lösung werden wichtige Designentscheidungen hinsichtlich der Gesamtlösung, die die digitale Lösung prägen, erst sehr spät im Bauprozess getroffen. Wir diskutieren dies im Detail, wenn wir die Grundlagen von Designprozessen zusammen mit einem pragmatischen Prozessmodell für den Bauprozess einer digitalen Lösung vorstellen.

Neben dem Prozesswissen besteht die Design-Kompetenz insbesondere in der Fähigkeit, mit den relevanten Stakeholdern über die geplante Lösung zu kommunizieren (vgl. [Cros2006]). Diese Kommunikation umfasst insbesondere die Ausarbeitung von Details und die Überprüfung der geplanten Lösung. Konzepte und Prototypen sind zentrale Instrumente für diese Kommunikation. In Abschnitt 2.2 führen wir daher in die Konzeptarbeit im Digital Design ein, während wir in Abschnitt 2.3 die Anwendung von Prototypen im Digital Design vorstellen.

2.1 Integration von Digital Design in den Bauprozess

Das Hauptmerkmal des Prozesses zum Design eines physischen Produkts ist die Trennung von Gestaltung und Herstellung: Der schöpferische Akt der Bestimmung und Definition von Form und Funktion eines Produkts findet vor dem physischen Akt der Herstellung (Realisierung) des Produkts statt, der lediglich aus einer wiederholten, oft automatisierten Replikation besteht (vgl. [Nobl1996]).

Jedes Prozessmodell, das am Ende des Gestaltungsprozesses mit impliziten Annahmen über Massenproduktionsprozesse arbeitet, ist für den Bau einer digitalen Lösung nur bedingt geeignet. Das Design einer digitalen Lösung ist ein fortlaufender Prozess während des gesamten Bauprozesses (siehe Abschnitt 1.3.3). Der Bau digitaler Lösungen erfordert Prozessmodelle, die Leitlinien für die Integration von Gestaltungsaktivitäten in den gesamten Bauprozess bieten.

Dennoch ist die Beschäftigung mit den Grundlagen von Designprozessen wichtig, um ein eigenes Verständnis für die Funktionsweise von Design zu entwickeln (vgl. [Dors2003]). In Abschnitt 2.1.1 geben wir daher einen Überblick über die Grundlagen von Designprozessen als Einstieg in die Welt des Designs und der Designprozesse.

In Abschnitt 2.1.2 stellen wir drei wesentliche Schritte des Bauprozesses für eine digitale Lösung vor und erörtern ihre Beziehung zum Digital Design. In Abschnitt 2.1.3 erörtern wir die Berücksichtigung der Qualität als Querschnittsaufgabe während des Bauprozesses. Unter 2.1.4 werden weitere wichtige Kompetenzfelder für den Bauprozess einer digitalen Lösung vorgestellt. Dieser Überblick dient zum einen als Ausgangspunkt für den weiteren Kompetenzerwerb und soll zum anderen aufzeigen, wie Digital Design mit anderen Bereichen zusammenhängt. Um die Integration von Digital Design in den Bauprozess abzuschließen, stellen wir in Abschnitt 2.1.5 ein idealisiertes Modell des Bauprozesses vor.

2.1.1 Grundlagen des Designprozesses

Das Erlernen von Design ist wie das Erlernen jeder anderen komplexen Fähigkeit: Es erfordert Bildung und Ausbildung [Cros2006]. Die Designausbildung stützt sich auf Modelle für Designprozesse und eine Beschreibung der Designarbeit, aber diese ersetzen nicht die Ausbildung und die reale Praxis. Kochen ist eine gute Analogie: Nehmen wir an, Sie wollen lernen, wie man gute italienische Pasta mit Tomatensauce kocht. Sie erhalten ein Rezept und befolgen die Anweisungen zum Kochen der Sauce. Das Ergebnis kann gut oder schlecht sein, weil das Rezept nur die abstrakten Schritte abdeckt und den konkreten Fall außer Acht lässt. Vielleicht waren die Tomaten beim ersten Kochversuch zu wässrig, und die Soße hätte länger als im Rezept beschrieben kochen müssen, um die Soße schmackhaft zu machen. Oder Sie hatten vielleicht die falschen Tomaten.

Sie können das Soßenrezept so lange studieren, wie Sie wollen; das Ergebnis wird beim nächsten Mal wahrscheinlich nicht viel besser sein. Die Pasta Sauce wird besser, wenn Sie mit dem Rezept experimentieren (z. B. mit Tomatenmark oder anderen Tomatensorten) und die Ergebnisse kritisch prüfen. Dies wird Ihr Verständnis für die innere Beziehung zwischen dem Rezept (Konzept), den Zutaten (Materialien) und der Pasta Sauce (Lösung) verbessern. Nur durch zahlreiche Experimente können Sie das Verfahren intuitiv verstehen und Ihre Fähigkeiten beim Kochen von Pasta Sauce verbessern. Im Laufe dieses Prozesses braucht man das Rezept immer weniger, und am Ende kann man die Sauce ohne Rezept kochen, weil man es verinnerlicht hat.

Die wichtige Lehre aus dieser Analogie ist, dass man Kochen nicht mit dem Studium des Rezepts verwechseln darf. Das Rezept ist der Ausgangspunkt und nicht das Ende. Designprozessmodelle sind, wie die Arbeitsanweisungen in einem Rezept, ein wesentlicher Bestandteil, aber sie sind ein Ausgangspunkt und es ist sinnvoll, sie zu studieren. Mit Hilfe eines Designmodells können Sie ein grundlegendes Verständnis für die Funktionsweise von Design erlangen, und solche Modelle sind ein guter Ausgangspunkt für einen DDP auf Foundation Level Niveau. Denken Sie jedoch daran, dass Sie Ihre Designfähigkeiten tatsächlich durch Üben und Experimentieren entwickeln.

Im Folgenden stellen wir ein Modell vor, das ein erstes Verständnis des Designprozesses vermittelt und es uns ermöglicht, wichtige Aspekte eines jeden Designprozesses zu diskutieren. Zum Abschluss der Grundlagen stellen wir ein Modell vor, das es uns ermöglicht, das Wesen der Designarbeit zu diskutieren.

2.1.1.1 Der Design-Squiggle als Designprozessmodell

Das erste Modell ist der Design-Squiggle von Damien Newman ([Newm2020], siehe Abbildung 7). Es zeigt, dass ein Designprozess typischerweise eine chaotische und iterative Aktivität ist, die irgendwie zu einem klaren Verständnis einer bestimmten Designlösung führt. Der Design-Squiggle zeigt drei verschiedene, streng voneinander getrennte Phasen:

- *Forschung & Synthese*: In dieser Phase geht es darum, den Problembereich zu verstehen und Einblicke in Benutzer, Kunden und die jeweilige Situation zu gewinnen. Sie ist typischerweise durch eine eher unstrukturierte Reise mit viel Rauschen und Unsicherheit gekennzeichnet. Irgendwann führt der Prozess jedoch zu einem Verständnis des Problems.

- *Konzept/Prototyp*: Sobald ein erstes Problemverständnis vorhanden ist, ist es möglich, erste Konzepte und Prototypen zu erstellen, um mögliche Lösungsideen zu untersuchen. Auch dieser Prozess ist nicht wirklich linear. Konzepte oder Prototypen können zu völlig neuen Erkenntnissen über das Problem führen. Es kann sogar bedeuten, dass das ursprüngliche Verständnis des Problems völlig verworfen werden muss und der Prozess von vorne beginnen muss.
- *Design*: Irgendwann wird eine Lösungsidee als endgültige Lösung präsentiert. Nun wird der Prozess linear, da die eine Lösung in allen Einzelheiten ausgearbeitet werden muss, bis sie zum endgültigen Design wird. Es ist wichtig zu erkennen, dass sich der Begriff Design hier auf das Endergebnis (Design als Substantiv) und nicht auf die Tätigkeit (Design als Verb) bezieht.

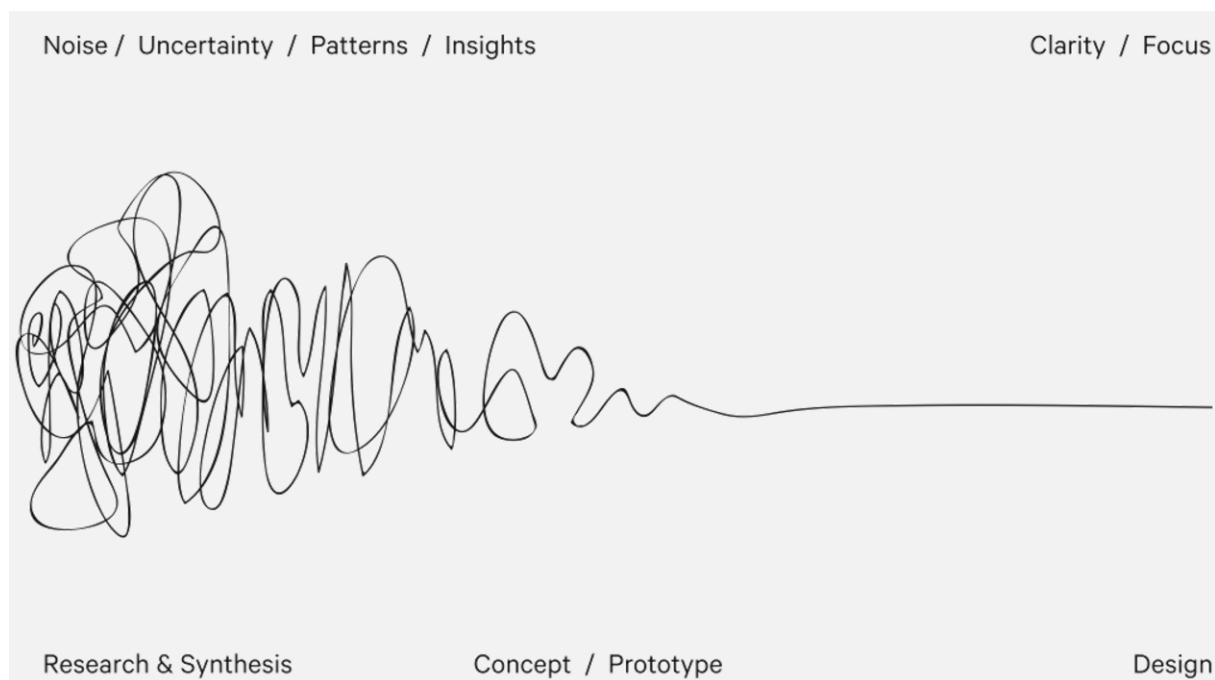


Abbildung 7 - Der Design-Squiggle [Newm2020]

Darüber hinaus stellt der Design-Squiggle drei wichtige Kernaspekte des Designprozesses vor, die Teil der Grundhaltung eines jeden DDP werden sollten.

Grundhaltung 1: Versuchen Sie immer, das Umfeld zu verstehen, bevor Sie an Lösungsideen arbeiten

Ein typischer Einsteigerfehler ist es, Design nur als die dritte Stufe des Design-Squiggle zu verstehen. Eine Voraussetzung für die Entwicklung guter Designlösungen ist das richtige Verständnis des gesamten Umfelds, für das die Lösung entwickelt werden soll. Besonderes Augenmerk sollte auf den Nutzungskontext gelegt werden, d. h. auf den Kunden der Lösung und die Umgebung, in der die Lösung eingesetzt wird.

Ein tiefgreifendes Verständnis der Umwelt ist wichtig, da viele detaillierte Designentscheidungen von den winzigen Details der Umwelt abhängen. Ein typischer Fallstrick für einen DDP ist der Glaube, dass er bereits ein gutes Verständnis der Umwelt hat und keine detaillierte Forschungsphase für eine bestimmte Lösung benötigt. Heutzutage kann sich das Umfeld sehr schnell ändern, so dass einmal erworbenes Wissen auch schnell wieder veraltet. Wir empfehlen

Ihnen daher dringend, die Phase der Forschung nicht zu überspringen und die Forschung klar von der Arbeit an der digitalen Lösung zu trennen.

In Abschnitt 2.1.2 finden Sie weitere Hinweise zum Verständnis der Umwelt.

Grundhaltung 2: Laufende Evaluierung von allem

Ein zweiter Einsteigerfehler besteht darin, zu schnell an das eigene Verständnis der Umwelt und die geschaffenen Lösungsideen zu glauben. Der Design-Squiggle zeigt deutlich, dass der Designprozess ein ziemlich chaotischer Prozess ist, der zwischen den verschiedenen Phasen hin und her pendelt. Ein Grund dafür ist, dass das anfängliche Verständnis der Umgebung selten richtig ist und dass die anfänglichen Lösungsideen selten die beste Lösung sind. Unerfahrene Designer lernen diese Lektion oft, wenn ihre Lösungsideen beim Kunden oder auf dem Markt scheitern.

Erfahrene Designer haben gelernt, die Ungewissheit zu tolerieren und haben die Evaluierung von allem (Verständnis der Umwelt und der Lösungsideen) zu einem Teil ihrer Grundhaltung gemacht, wobei sie immer nach Möglichkeiten suchen, ihr Verständnis und ihr Design zu evaluieren. Das bedeutet nicht, dass erfahrene Designer immer schwerfällige Evaluationsmethoden anwenden; es bedeutet nur, dass erfahrene Designer sich der Grenzen ihres eigenen Verständnisses bewusst sind, was in der Praxis auch die Diskussion von Erkenntnissen aus der Ich-Perspektive erfordert.

In Abschnitt 2.1.2 finden Sie weitere Hinweise dazu, wie Sie die Evaluierung im Rahmen des Bauprozesses angehen können.

Grundhaltung 3: Iteration als Arbeitsweise

Ein dritter Einsteigerfehler ist die Annahme eines linearen Designprozesses. Der Design-Squiggle macht deutlich, dass jeder Designprozess ein eher chaotischer und iterativer Prozess ist. Erst am Ende, wenn die Lösungsidee wirklich klar ist, wird der Prozess eher linear.

Erfahrene Designer haben gelernt, diese Arbeitsweise zu tolerieren, und haben die Iteration (d. h. das erneute Verstehen der Umwelt und die erneute Arbeit an Lösungsideen) zur normalen Arbeitsweise gemacht. Nur durch mehrfache Iterationen können mehrere Lösungsideen erstellt und evaluiert werden, um die wirklich vielversprechenden Lösungsideen zu identifizieren. Das bedeutet nicht, dass der gesamte Prozess und das gesamte Team eine Iteration durchführt - manchmal kann eine Iteration auch ein recht kurzes Ereignis sein, das nur im Kopf eines Designers stattfindet.

In Abschnitt 2.1.2 finden Sie weitere Hinweise zum iterativen Arbeiten im Bauprozess.

2.1.1.2 Das Dual-Mode-Modell des Designs

Ein guter Designer zu sein bedeutet, eine Designpersönlichkeit zu entwickeln. Für uns bedeutet eine Designer-Persönlichkeit vor allem, die Fähigkeit zu entwickeln, zwischen der Ich-Perspektive und der Dritte-Person-Perspektive zu wechseln. Die Dritte-Person-Perspektive bedeutet eine möglichst objektive Betrachtung der Designaktivitäten, die sich auf Modelle, Theorien, (Kunden-/Benutzer-)Forschung, Prinzipien usw. konzentriert. Die Ich-Perspektive auf Design bedeutet, Design aus eigener Erfahrung zu verstehen (aus der Arbeit mit Design) und im Austausch mit erfahrenen Designern zu lernen.

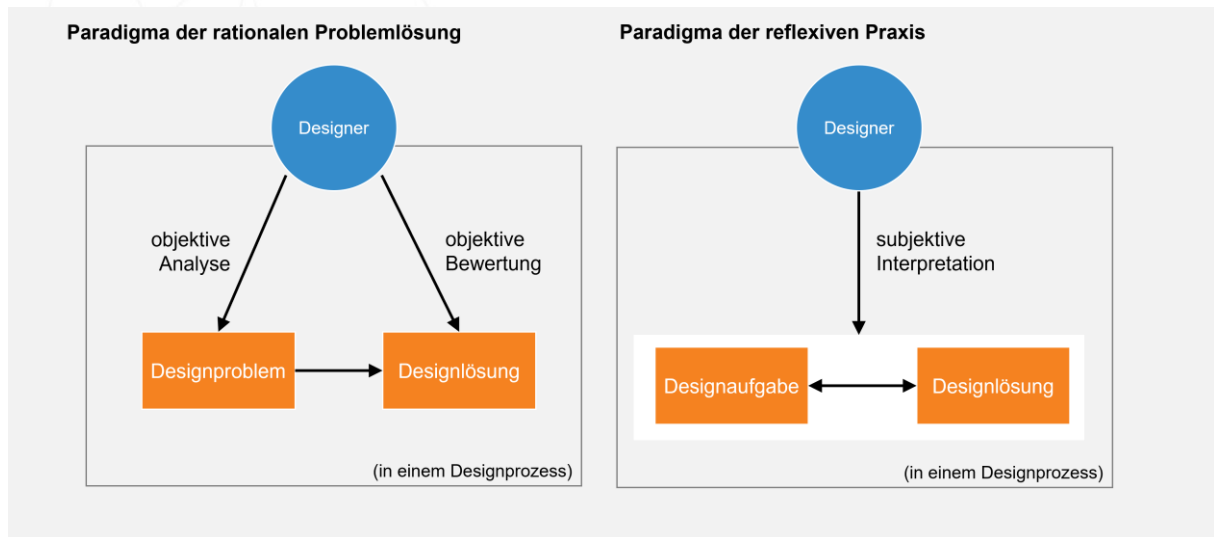


Abbildung 8 - Das Dual-Mode-Modell des Designs [Dors1997]

Das Dual-Mode-Modell des Designs [Dors1997] beschreibt diese beiden Perspektiven mit zwei Modi (siehe Abbildung 8):

- *Paradigma der rationalen Problemlösung / objektiver Modus:* In diesem Modus (linker Teil von Abbildung 8) arbeitet der Designer durch rationale Analyse und Beobachtung objektiv am Designproblem (Arbeit im Problemraum) und will es lösen. Das Ziel ist es, so objektiv und rational wie möglich zu arbeiten. Die Erkenntnisse aus Analyse und Beobachtung werden in eine Lösung umgesetzt (Arbeiten im Lösungsraum). Die Lösung selbst wird wiederum beobachtet, analysiert und bewertet.
- *Paradigma der reflektierenden Praxis / der subjektive Modus:* In diesem Modus (rechter Teil von Abbildung 8) arbeitet der Designer subjektiv an der gesamten Designsituation, will sie verstehen und zielt darauf ab, einen Weg zu finden, um mit der Designaufgabe fortzufahren. Der Designer bearbeitet die Designaufgabe (die gegebene Situation und den Zeitrahmen) in Bezug auf die gewünschte Designlösung. Der Designer ist sich seiner subjektiven Perspektive und seiner eigenen Hintergrundtheorie ebenso bewusst wie den Fähigkeiten seines impliziten Wissens. Außerdem können sie bewusst alternative Perspektiven auf die Designaufgabe einnehmen. Der subjektive Modus ist wichtig, wenn die Designaufgabe unklar oder schlecht definiert ist, oder wenn es ethische oder moralische Konflikte in Bezug auf die Designaufgabe gibt.

Für Design-Einsteiger bietet dieses Modell drei wichtige Lektionen:

- Design kann mit einer objektiven oder subjektiven Grundhaltung angegangen werden. Beide Modi sind wichtig, und ein erfahrener Designer muss beide Modi nutzen und bei Bedarf zwischen ihnen wechseln.
- Die Arbeit am Designprozess (z. B. die Planung der nächsten Schritte zur Analyse des Problems oder zur Erstellung eines Prototyps) ist Teil des Verfahrens der reflektierenden Praxis. Es gibt viele Möglichkeiten, an ein Designproblem heranzugehen, und die Botschaft der reflektierenden Praxis ist, dass die Art und Weise, wie man in einer bestimmten Situation vorgeht, von subjektiven Faktoren abhängt (z. B. von Erfahrung und Ausbildung).

- Wenn Sie mit Ihrem Designproblem oder mit der Lösung nicht weiterkommen, schalten Sie in den Modus der reflektierenden Praxis und versuchen Sie, das Gesamtbild zu verstehen. Hinterfragen Sie den Prozess, hinterfragen Sie das gegebene Problem und hinterfragen Sie Ihr bisheriges Verständnis.

Es gibt weitaus mehr Literatur zu den Grundlagen des Designs. In Kapitel 6 empfehlen wir mehrere Bücher als weiterführende Lektüre.

2.1.2 Die drei wesentlichen Schritte des Bauprozesses für eine digitale Lösung

Der Bauprozess für eine digitale Lösung kann in drei wesentliche Schritte unterteilt werden: den Scoping-Schritt (Auftragsklärung), den konzeptuellen Schritt und den Entwicklungs- und Betriebsschritt. Denken Sie daran, dass der Designprozess in jedem Schritt einem Prozess folgt, der dem Design-Squiggle ähnlich ist (siehe Abbildung 7). Darüber hinaus wird die Designarbeit während des gesamten Prozesses zwischen dem objektiven und dem subjektiven Modus hin und her wechseln (siehe Abschnitt 4.3). Der Hauptunterschied zwischen den beiden Schritten ist die Abstraktionsebene, auf der das Design stattfindet.

In den folgenden Unterabschnitten wird jeder Schritt in abstrakter Form beschrieben. Für jeden Schritt werden die allgemeinen Ergebnisse und Aktivitäten beschrieben. Abbildung 9 fasst die drei Schritte und die allgemeinen Arbeitsprodukte jedes Schritts (farbige Kästen) auf der jeweiligen Abstraktionsebene zusammen.

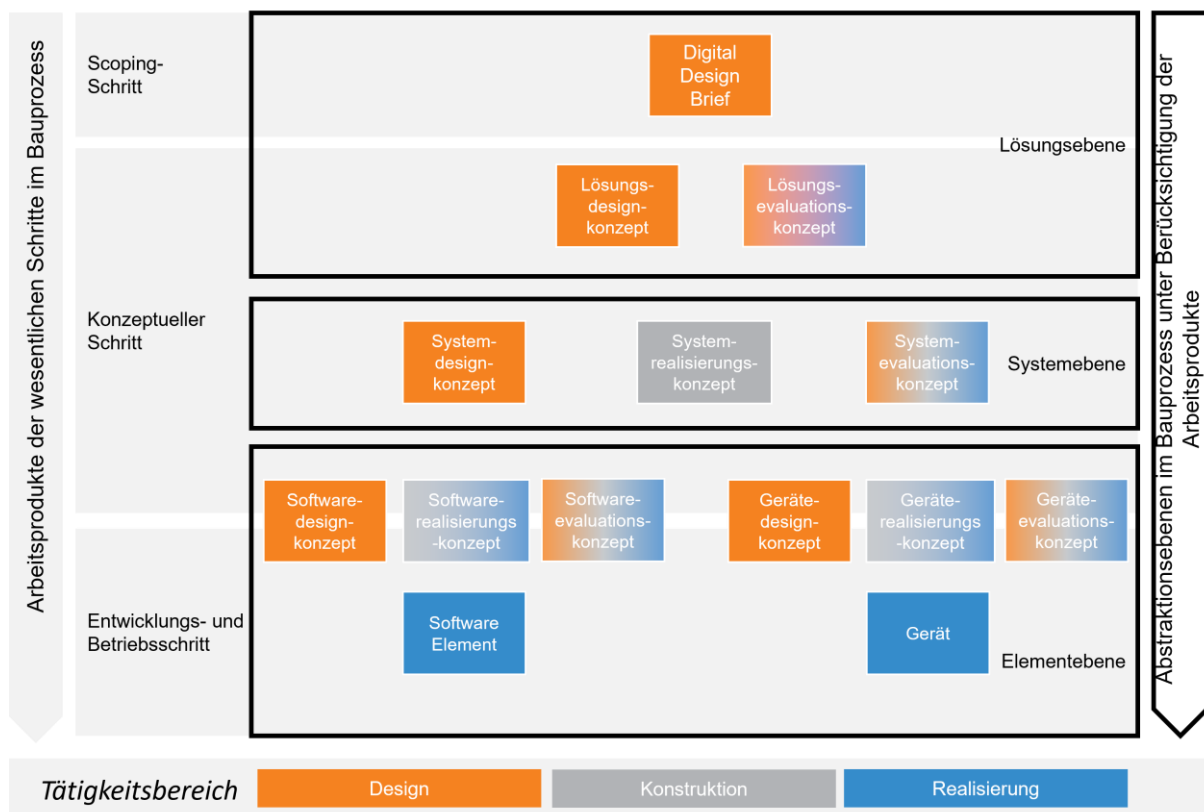


Abbildung 9 - Die drei wesentlichen Schritte und Arbeitsprodukte des Bauprozesses

Die Farbe jedes Kästchens gibt an, welcher Tätigkeitsbereich zuständig ist; Kästchen mit mehr als einer Farbe gehören zu mehreren Tätigkeitsbereichen. Beachten Sie, dass die verschiedenen Konzepttypen, die in Abbildung 9 gezeigt werden, nicht unbedingt als eigenständige Dokumente

oder in eigenständigen Werkzeugen gepflegt werden müssen. Die Verteilung von Konzepttypen auf Dokumente oder Werkzeuge ist eine Frage der Arbeitsorganisation.

Ziel der folgenden Beschreibung ist es, ein allgemeines Verständnis für die verschiedenen Schritte des Bauprozesses, einschließlich der allgemeinen Ergebnisse und der Beziehung zwischen den Schritten und den Ergebnissen, zu vermitteln. Es ist wichtig zu erkennen, dass die drei Schritte nicht als ein lineares Prozessmodell gedacht sind, das sofort angewendet werden kann. Iterationen sind wichtig, um eine Rückkopplungsschleife zwischen den drei Abstraktionsebenen zu schaffen (siehe Abschnitt 1.3.4.2). Stattdessen beschreiben die Schritte verschiedene Kategorien, die sich in ihrer Art unterscheiden, weshalb wir die drei Schritte als wesentlich bezeichnen. Anhand der drei Schritte lässt sich die Gesamtstruktur eines Bauprozesses für eine neue digitale Lösung verstehen. Sie können auch verwendet werden, um die weitere Entwicklung einer digitalen Lösung zu verstehen und zu strukturieren.

Ein anwendbarer Bauprozess für die Erstellung einer neuen digitalen Lösung wird in Kapitel 5 vorgestellt, einschließlich detaillierter Methoden, Technologien und Prozessrichtlinien.

2.1.2.1 Der Scoping-Schritt (Auftragsklärung)

Ziel des Scoping-Schrittes ist es, mit dem Auftraggeber und den relevanten Stakeholdern ein gemeinsames Verständnis des Handlungsbedarfs (warum beginnen wir einen neuen Bauprozess?) und der geplanten digitalen Lösung (was wollen wir erreichen?) zu erarbeiten. Aus der Perspektive des Digital Designs sollte beim Scoping einer digitalen Lösung zwischen vier Perspektiven des Kontexts unterschieden werden (siehe Abbildung 10): der ökologischen, der baulichen, der technischen und der sozialen Perspektive.

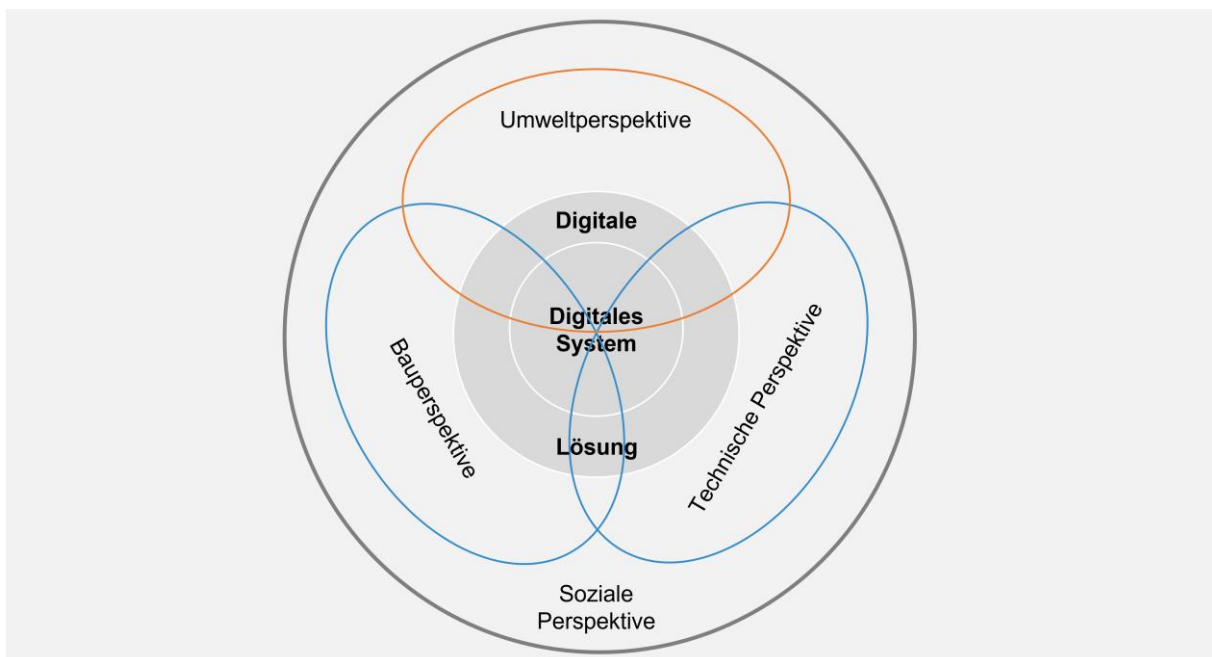


Abbildung 10 - Die Kontext-Scoping-Perspektiven des Digital Design

Diese vier Perspektiven müssen zusammen betrachtet werden, denn sie definieren das Spannungsfeld, in dem die digitale Lösung entstehen soll. Im Folgenden wird jede Perspektive erläutert, einschließlich ihrer Beziehung zu den anderen Perspektiven.

Die Umweltperspektive

Die Umweltperspektive konzentriert sich auf das Umfeld, in das die realisierte digitale Lösung eingebettet ist. Diese Perspektive wird oft als Kontext bezeichnet (vgl. [BeHo1998], [CPRE2020], [CPUX2018]) und kann in verschiedene Facetten unterteilt werden (vgl. [ISO2018]):

- *Direkter Nutzungskontext:* Umgebung (z. B. Ort, Zeit und Umgebungsbedingungen), in der die realisierte digitale Lösung betrieben und genutzt wird
- *Physische Umgebung:* Umgebung, in die der Nutzungskontext eingebettet wird (z. B. eine Abteilung, eine Familie)
- *Soziales Umfeld:* Umfeld, das für die Umgebung relevant ist und sich auf diese auswirkt (z. B. ein Unternehmen)

Das Verständnis des Nutzungskontextes und der direkten Umgebung einer digitalen Lösung ist besonders wichtig, da die Umgebung eine wichtige Informationsquelle für die Gestaltung der digitalen Lösung ist. Eine eingehende Analyse des Umfelds im Rahmen des Scoping-Schritts ist jedoch schwierig. Die Idee einer intensiven Erstanalyse des Umfelds stammt aus einer Zeit, als die Digitization das Hauptproblem war. Sie geht davon aus, dass der Kontext bereits existiert und für analytische Methoden zugänglich ist, um die Anforderungen zu verstehen, die zur Gestaltung der digitalen Lösung genutzt werden können. Für die Digitalisierung und die digitale Transformation bleibt dieses Verständnis wichtig, ist aber zu Beginn nicht möglich, da das durch eine neuartige digitale Lösung geschaffene Umfeld noch nicht existiert und daher nur teilweise für analytische Methoden zugänglich ist. Doch gerade hier ist der Designer gefordert, aus seiner subjektiven Perspektive im Sinne des Paradigmas der reflektierenden Praxis, ein gefühltes Verständnis für die Umwelt zu entwickeln.

Eine zweite Herausforderung für das Verständnis des Nutzungskontextes und des direkten Umfelds einer digitalen Lösung während des Scoping-Schrittes besteht darin, dass es oft keine klare Vorstellung der digitalen Lösung gibt. Daher wird empfohlen, die Umgebung einer digitalen Lösung während des Scoping-Schrittes generell zu charakterisieren, indem man sich auf die direkte und weitere Umgebung konzentriert. So ist es beispielsweise sinnvoll, sich auf das Problem oder den Handlungsbedarf zu konzentrieren, der den Bedarf an einer neuen digitalen Lösung begründet, anstatt sich in dieser frühen Phase auf den Kunden zu konzentrieren. In diesen frühen Gesprächen mit Personen in der Auftraggeber-Rolle werden Einflussfaktoren für potenzielle Veränderungen des Umfelds aufgedeckt und dadurch neue Ideen für das Wertversprechen hervorgehoben. Dies ermöglicht die Vorbereitung der Definition des Wertversprechens im konzeptuellen Schritt (siehe unten). Dies geschieht durch das Sammeln erster Ideen für erfolgreiche Geschäftsmodelle oder Customer Journeys, das Schaffen eines Bewusstseins für den Bedarf an neuen Geschäftsmodellen oder Wertversprechen (siehe Abschnitt 4.2), durch die Beschreibung der Motivation hinter der digitalen Lösung (the case for action) und durch die Etablierung einer gemeinsamen Sprache für die Beschreibung, Gestaltung und Analyse der Entwicklung der digitalen Lösung im Laufe des Bauprozesses (siehe Abschnitt 4.3).

Beispiel YPRC. In der YPRC-Fallstudie lag dieser frühe und breite Fokus auf der Idee, das Trainingserlebnis für Läufer mit digitaler Technologie zu verbessern. Es gab einige Ideen, wie diese Verbesserung aussehen könnte. Der Handlungsbedarf bestand darin, dass es einige digitale Technologien und Daten (Remote-Kommunikation, Trainingsdaten) zu geben scheint, die das Potenzial haben, diese Trainingserfahrung zu verbessern.

Die Bauperspektive

Die Bauperspektive konzentriert sich auf den Teil des Kontexts, in dem die digitale Lösung entsteht. Die Bauperspektive umfasst die folgenden Aspekte:

- *Das Umsetzungsteam:* Die Personen (als Stakeholder bezeichnete Mitglieder des Umsetzungsteams), die tatsächlich die Aktivitäten des Bauprozesses durchführen
- *Die Bauorganisation:* Die Organisation, die den Bauprozess vorantreibt und unterstützt
- *Die Kundenorganisation:* die Organisation, die für die Erstellung der digitalen Lösung verantwortlich ist, oder das Unternehmen, zu dem die Bauorganisation gehört (z. B. eine interne IT Abteilung).

Die Bauperspektive wird bei der Planung einer neuen digitalen Lösung oft vernachlässigt. Die Verfügbarkeit von Fachkräften und Ressourcen wird oft als selbstverständlich vorausgesetzt. Eine solche zweckorientierte Sichtweise auf die Bauperspektive stellt ein erhebliches Risiko für innovative Ideen dar, insbesondere wenn qualifiziertes Personal eine knappe Ressource ist.

Ein weiterer Faktor in dieser Perspektive ist die Organisation des Auftraggebers. Bestehende Lösungen oder Produkte in der Auftraggeber-Organisation können mit der neuen Lösung konkurrieren. Insbesondere wenn eine digitale Lösung darauf abzielt, bestehende Dienstleistungen zu ersetzen (z. B. der Online-Verkauf von Waren als Ersatz für Lagerhäuser), neigen Unternehmen, die solche Dienstleistungen anbieten, dazu, gegen neuartige Lösungen zu kämpfen, um ihr bestehendes Geschäft zu schützen. Die Berücksichtigung dieser Perspektive in einem frühen Stadium ist sehr wichtig, um die politischen Kräfte und Werte zu verstehen, welche die Organisation bestimmen. Es ist wichtig zu erkennen, dass die Bauperspektive nicht die Wettbewerber oder verwandte Lösungen einschließt, die von anderen Organisationen angeboten werden. Diese Aspekte sind Teil der sozialen Perspektive.

Die technische Perspektive

Die technische Perspektive konzentriert sich auf den technischen Kontext, der die digitale Technologie bereitstellt und entwickelt. Der technische Kontext umfasst die folgenden Facetten:

- *Verfügbare Technologien:* Die Technologie selbst, die für den Aufbau digitaler Lösungen verwendet wird.
- *Technologieanbieter:* Organisationen, die Hardware oder Software liefern, die zum Aufbau der digitalen Lösung verwendet werden kann. Anbieter werden besonders wichtig, wenn sie eine Kerntechnologie (z. B. einen Programmierrahmen) für eine digitale Lösung bereitstellen. In einer solchen Situation hängt die neue digitale Lösung (und die Organisation, die die digitale Lösung nutzt) vom Anbieter ab.
- *Menschen:* Technologie wird von Menschen bereitgestellt und entwickelt. Sie müssen eingestellt oder geschult werden, um mit der Technologie umgehen zu können.

Während des Scoping-Schrittes wird eine umfassende Betrachtung der technischen Perspektive empfohlen, um ein umfassendes Verständnis der verfügbaren Technologie zu erlangen. Neben den bestehenden Technologien sollten auch neue und potenzielle künftige Technologien (z. B. Blockchain) während des Scoping-Schrittes berücksichtigt werden. Selbst wenn sich eine Technologie in einem frühen Stadium befindet und noch nicht marktreif ist, kann sie im Laufe des Bauprozesses für die geplante digitale Lösung zur Marktreife gelangen. Hier ist es wichtig, daran zu denken, dass wir uns im Scoping-Schritt befinden und ein breites Verständnis der technischen Möglichkeiten anstreben.

Beispiel YPRC. Ein gutes Beispiel für diese weit gefasste Sichtweise auf Technologie aus der YPRC-Fallstudie ist der Einsatz von künstlicher Intelligenz (KI) auf dem Smartphone. In der Fallstudie war die aktuelle Smartphone-Hardware noch nicht leistungsfähig genug, um KI-Coaching-Tipps in Echtzeit zu geben. Dennoch wird z. B. die Kapazität der Smartphone-Hardware zunehmen, und schließlich wird die Hardware über genügend Leistung verfügen, um KI-Coaching-Tipps in Echtzeit zu geben. Dies bei der Konzeption der YPRC-Fallstudie zu berücksichtigen, ist von entscheidender Bedeutung, um diese Funktion einbeziehen zu können, wenn die Technologie verfügbar ist.

Die soziale Perspektive

Die soziale Perspektive stellt die gesamte soziale Welt dar, in die die digitale Lösung eingebettet ist. Die soziale Perspektive kann verschiedene Faktoren umfassen, die relevant werden können. Zu den wichtigen Aspekten der sozialen Perspektive gehören:

- Wettbewerber, die ähnliche digitale oder analoge Lösungen anbieten (z. B. Unternehmen, die Running Apps anbieten)
- Unternehmen mit verwandten Dienstleistungen (z. B. Unternehmen, die Laufdienstleistungen oder Laufausrüstung verkaufen)
- Organisationen im Sektor der digitalen Lösung (z. B. Lauftreffs)
- Politische Organisationen/nichtstaatliche Organisationen im Bereich der digitalen Lösung (z. B. politische Parteien)
- Wichtige politische oder gesellschaftliche Personen im Sektor der digitalen Lösung (z. B. der Sportminister eines Landes)

Um die Ergebnisse des Scoping-Schrittes festzuhalten, empfehlen wir die Erstellung eines Digital Design Briefs (Abkürzung: Design Brief):

Digital Design brief: The description of the context, vision, scope, and general terms for building a digital solution.

Eine detaillierte Vorlage für einen Digital Design Brief wird in Abschnitt 2.2.2.1 vorgestellt, einschließlich der Beziehungen zu den oben dargestellten Scoping-Perspektiven.

Ein Design Brief kann schnell erstellt werden, wenn sich die Auftraggeber-Rolle über den Umfang der geplanten Lösung im Klaren ist. Er kann auch zu einem eigenständigen Projekt werden, wenn die Auftraggeber-Rolle keine klare Vorstellung von der digitalen Lösung hat oder wenn die organisatorische Situation von Business und IT sehr komplex ist, was typischerweise in großen Unternehmen der Fall ist. In einer solchen Situation kann ein qualifizierter und erfahrener DDP die Rolle eines Beraters übernehmen, der die Auftraggeber-Rolle dabei unterstützt, eine überzeugende Vision und einen hochwertigen Design Brief zu entwickeln. Hier ist der Modus der reflektierenden Praxis wichtig (siehe Abbildung 8).

2.1.2.2 Der konzeptuelle Schritt

In der konzeptuellen Phase erarbeiten die Stakeholder und das Umsetzungsteam ein erstes gemeinsames Verständnis der digitalen Lösung (einschließlich des digitalen Systems), das ausreicht, um das Risiko des Beginns von Entwicklungs- und Betriebsphase zu akzeptieren. Der Schwerpunkt liegt hier ausdrücklich auf dem Risiko, da die Konzeptarbeit vor allem diesem Zweck dient. Umfang und Detaillierungsgrad der Konzeptarbeit sollten unter dem Gesichtspunkt des Risikos festgelegt werden. Ist beispielsweise der Umfang einer digitalen Lösung klein und gut

bekannt, kann eine sehr kurze Konzeptphase ausreichend sein. Wenn hingegen viele Aspekte einer Lösung noch unklar sind, empfiehlt sich eine wesentlich umfangreichere Konzeptarbeit.

Obwohl die digitale Lösung und das digitale System eng miteinander verknüpft sind (siehe Abschnitt 1.2), empfehlen wir, zwei dedizierte Designkonzepte zu erstellen: Eines für die digitale Lösung und eines für das digitale System. Die Hauptmotivation für die Erstellung von zwei getrennten Designkonzepten ist, dass beide Konzepte unterschiedliche Zielgruppen und unterschiedliche Zwecke haben (siehe Abschnitt 1.2 zum Unterschied zwischen der Lösungsebene und der Systemebene).

Die Lösungsperspektive im konzeptuellen Schritt

Das Lösungsdesignkonzept konzentriert sich auf die soziotechnische Perspektive und erfasst die Anforderungen der Auftraggeber- und der Kunden-Rolle an die digitale Lösung. Es ist wie folgt definiert:

Solution design concept: The description of the goals, the business model, and the overall idea of a digital solution.

Das Geschäftsmodell beschreibt, was die digitale Lösung aus Auftraggeber-Sicht leisten soll. Dies kann messbare Ziele, einen Kostentreiber, Einnahmequellen oder Annahmen zu Aspekten wie wichtigen Partnern oder Kundensegmenten umfassen. Die Gesamtidee ist eine Beschreibung der digitalen Lösung in der Geschäftssprache der Auftraggeber-Organisation und in der Sprache des Kunden im Sinne eines Wertversprechens. Um den Evaluationsansatz für die digitale Gesamtlösung zu erfassen, empfehlen wir, ein Konzept für diesen Teil der Qualitätssicherung zu erstellen:

Solution evaluation concept: The evaluation concept for a digital solution.

Auf der Lösungsebene gibt es mehrere Evaluierungsansätze, die angewandt werden können (z. B. Prototypen, Fokusgruppen, Expertenbewertungen oder Umfragen). Im Allgemeinen ist es wichtig zu erkennen, dass sich die Evaluierung eindeutig auf die Aspekte konzentriert, die durch die Lösung definiert sind, d.h. das Geschäftsmodell und das Wertversprechen. Das Lösungs evaluationskonzept wird zunächst im konzeptuellen Schritt erstellt.

Eine weitere Verfeinerung des Lösungs evaluationskonzepts findet während des Entwicklungs- und Betriebsschritts statt (siehe unten), da detaillierte Aspekte der digitalen Lösung während dieses Schritts definiert werden und daher während dieses Schritts bewertet werden müssen.

Beispiel YPRC. Der Preis für den Remote-Coaching-Service ist ein Beispiel für diese Verfeinerung. Bei der Realisierung der App könnte die Idee aufkommen, einzelne Remote-Coaching-Sitzungen zu einem bestimmten Preis zu verkaufen. Die Definition dieses Dienstes ist Teil der Lösungsebene und die Bewertung der Akzeptanz dieses Dienstes (z. B. durch eine Kundenbefragung) wird als Teil des Lösungs evaluationskonzepts definiert.

Die Systemperspektive im konzeptuellen Schritt

Das Systemdesignkonzept konzentriert sich auf den Systemteil der digitalen Lösung, erfasst die Systemebene und ist wie folgt definiert:

System design concept: The description of the system-relevant objectives and of the overall form, function, and quality of a digital system.

Das Systemdesignkonzept stellt die Systemperspektive dar - es beschreibt das digitale System innerhalb der Gesamtidee der digitalen Lösung in einem Detaillierungsgrad, der die Planung der technischen Realisierung ermöglicht. Als ergänzende Perspektive zum Lösungsdesignkonzept konzentriert sich das Systemdesignkonzept auf Elemente, Bausteine und deren Beziehungen und verwendet eine eher technische Sprache, die für nicht-technische Stakeholder normalerweise nicht direkt zugänglich ist. Dennoch sind die Lösung und das Systemdesignkonzept eng miteinander verbunden, und es liegt in der Verantwortung des Designs, beide Konzepte miteinander in Einklang zu bringen.

Beispiel YPRC. Wenn das Geschäftsmodell von YPRC (Lösungsdesignkonzept) vorsieht, dass Laufeinheiten auf einer Session-Basis gekauft werden, muss das Systemdesignkonzept in irgendeiner Weise eine Möglichkeit vorsehen, für den Service zu bezahlen. Wenn sich das Geschäftsmodell ändert - z. B. kann der Läufer Coaching-Einheiten als eine Art Belohnung erhalten - muss das Systemdesignkonzept die Möglichkeit definieren, diese Belohnungen zu erhalten.

Je nach Komplexität und technischer Neuartigkeit einer digitalen Lösung wird empfohlen, ein Realisierungskonzept zu erstellen, das sich auf die detaillierten Realisierungsaspekte des digitalen Systems innerhalb der digitalen Lösung konzentriert:

System realization concept: The description of the technically relevant system objectives and of the overall technical form, function, and quality of a digital system.

Das Konzept für die Systemrealisierung liegt in der Verantwortung von Konstruktionsexperten. Die konkrete Form und der Inhalt hängen von verschiedenen Faktoren ab und sind nicht Gegenstand dieses Handbuchs. Dennoch liegt es in der Verantwortung des Designs, die Notwendigkeit der Erstellung eines solchen Systemrealisierungskonzepts gemeinsam mit anderen Stakeholdern zu bewerten, um das Risiko zu minimieren, konzeptuelle Entscheidungen zu treffen, die zu teuren oder sogar unrealisierbaren Lösungen führen können. Darüber hinaus liegt es in der gemeinsamen Verantwortung von Design und Konstruktion, das Systemdesignkonzept und das Systemrealisierungskonzept aufeinander abzustimmen.

Beispiel YPRC. Ein Beispiel für ein Systemrealisierungskonzept in der YPRC-Fallstudie könnte die Sprachkommunikation für den Remote-Lauftrainer sein. Wenn das Projektteam unsicher ist, ob eine Fernsprachverbindung über eine mobile Internetverbindung technisch machbar ist, sollte es Voice-over-IP-Experten zu Rate ziehen und ein Systemrealisierungskonzept ausarbeiten, das mögliche Realisierungsalternativen für eine solche Funktionalität untersucht.

Um den Evaluierungsansatz des digitalen Systems zu erfassen, empfehlen wir, ein eigenes Konzept für diesen Teil der Qualitätssicherung zu erstellen:

System evaluation concept: The evaluation concept for a digital system.

Das Systemevaluationskonzept kombiniert die Aspekte Gestaltung, Konstruktion und Realisierung. Daher wird das Systemevaluationskonzept mit Fachleuten aus allen Tätigkeitsbereichen erarbeitet. Aus gestalterischer Sicht befasst sich das Konzept der Systemevaluation insbesondere mit den Qualitätsanforderungen, die das Gesamtsystem erfüllen muss.

Beispiel YPRC. Ein gutes Beispiel für ein Systemevaluationskonzept aus der YPRC-Fallstudie ist wiederum der Remote-Coaching-Service. Für diesen Service müssen mehrere Elemente des YPRC-Systems zusammenarbeiten: Die Smartwatch misst Gesundheitsdaten, die von der App an das Portal übertragen werden. Vom Portal werden die Daten dann auf den PC des Lauftrainers übertragen.

Das bedeutet, dass die Gesundheitsdaten durch drei Elemente übertragen werden, bevor der Coach sie sehen kann. Um eine echtzeitnahe Visualisierung der Gesundheitsdaten zu gewährleisten, sollten mehrere Systemtests in verschiedenen Umgebungen durchgeführt werden.

Solche Tests können sehr früh im Designprozess des Systems festgelegt werden. Es ist sogar möglich, einen eigenen Prototyp für diesen Zweck zu erstellen. Ein Beispiel für einen solchen Prototyp während der Konzeptionsphase wird in der YPRC-Fallstudie beschrieben.

Aus Sicht der Konstruktion und Realisierung befasst sich das Konzept der Systemevaluation mit Systemtests und Systemintegrationstests. Diese Tests befassen sich mit der technischen Integration des gesamten Systems oder mit einer Teilmenge von Elementen des Systems.

Ähnlich wie das Lösungs evaluationskonzept wird auch das Systemevaluationskonzept zunächst während des konzeptuellen Schritts erstellt und während der Realisierung der digitalen Lösungen weiter verfeinert.

Die Erstellung der Konzepte auf Lösungsebene und der Konzepte auf Systemebene ist ein wichtiger Schritt, da beide Ebenen die Grundlage für die Entscheidung über den nächsten Schritt des Bauprozesses bilden. Daher wird empfohlen, angemessene Anstrengungen zu unternehmen, um beide Ebenen mit Hilfe verschiedener Technologien und Prototypen zu erstellen und zu validieren.

Je nach Art der Lösung muss ein DDP weitere Fachleute in den Erstellungsprozess der beiden Konzepte einbeziehen. Die verschiedenen Zuständigkeitsbereiche werden in diesem Zusammenhang in Abschnitt 2.1.4 erläutert.

2.1.2.3 Der Entwicklungs- und Betriebsschritt

Die eigentliche Realisierung der digitalen Lösung erfolgt im Entwicklungs- und Betriebsschritt. Wir betrachten die Entwicklung und den Betrieb einer digitalen Lösung zusammen in einem Schritt, da sich die typische digitale Lösung kontinuierlich weiterentwickelt (z. B. werden neue Funktionen hinzugefügt, bestehende veraltete Funktionen werden entfernt). Das bedeutet, dass der Betrieb einer digitalen Lösung und die Weiterentwicklung im laufenden Betrieb bereits in einem frühen Stadium eines jeden Bauprozesses berücksichtigt werden sollte. Obwohl dieses Thema nicht in

den Zuständigkeitsbereich von Digital Design fällt, hat es Auswirkungen auf die Gestaltung einer digitalen Lösung. Wir werden auf dieses Thema später in diesem Abschnitt zurückkommen.

Der wichtige Unterschied zwischen diesem Schritt und dem vorangegangenen konzeptuellen Schritt besteht darin, dass die Konzepte bis zu einem Detailgrad ausgearbeitet werden müssen, der die Realisierung der digitalen Lösung ermöglicht.

Es ist Teil des Digital Designs, die Elemente der digitalen Lösung auszuarbeiten. Diese Details werden in der Regel in einem Designkonzept auf Elementebene festgehalten:

Element design concept: The description of the element-relevant objectives and of the form, function, and quality of an element of a digital solution.

Im Gegensatz zum Systemdesignkonzept konzentriert sich das Elementdesignkonzept auf ein bestimmtes Element der digitalen Lösung und muss einen Detaillierungsgrad erreichen, der für die Konstruktion und Realisierung ausreichend ist. Darüber hinaus wird das Elementdesignkonzept verwendet, um die Details eines bestimmten Elements mit den relevanten Stakeholdern zu diskutieren und zu bewerten. Es liegt in der Verantwortung von Digital Design, die richtigen Stakeholder zu bestimmen, die an einem bestimmten Detail eines Elements beteiligt sein müssen. Wird das zu prüfende Element beispielsweise vom Benutzer verwendet, sollte von diesem ein Feedback zu allen benutzerrelevanten Aspekten (z. B. Form des User Interface, Funktionalität) eingeholt werden.

Darüber hinaus liegt es in der Verantwortung des Digital Design, über Prototyping-Aktivitäten zu entscheiden, um wichtige Details eines Elements weiter zu evaluieren. Wenn z. B. die Eignung einer wichtigen Funktion für den Benutzer unklar ist oder die Realisierbarkeit einer Funktionalität fraglich ist, sollten Sie eine prototypische Realisierung dieser Funktion einplanen, um die Funktion zu evaluieren.

Es gibt wichtige Abhängigkeiten zwischen der Element- und der Systemebene, und es liegt in der Verantwortung des Digital Designs, die Konsistenz zwischen den Designkonzepten des Systems und der Elemente zu gewährleisten. Die Einzelheiten dieser Beziehungen werden in Abschnitt 2.2.6 erörtert.

Aus der Prozessperspektive kann der erforderliche Detaillierungsgrad nur mit einem iterativen Design-, Konstruktions- und Realisierungsprozess erreicht werden. Das bedeutet, dass ein DDP typischerweise mit weiteren Fachleuten aus den Bereichen Design (z. B. Industriedesigner, Interaktionsdesigner, Requirements Engineers, UX-Experten), Konstruktion (z. B. Softwarearchitekten, Elektroingenieure) und Realisierung (z. B. Softwareingenieure, Produktionsingenieure) zusammenarbeitet. Eine ausführliche Liste der damit verbundenen Kompetenzen ist in Abschnitt 2.1.4 zu finden.

Zusätzlich zum Elementdesignkonzept werden durch die Konstruktions- und Realisierungsaktivitäten typischerweise Realisierungskonzepte erstellt, die das Elementdesignkonzept ergänzen. Wir nennen solche Konzepte Elementrealisierungskonzepte.

Element realization concept: The description of the technically relevant element objectives and of the technical form, function, and quality of an element of a digital solution.

Um den Evaluierungsansatz für ein Element zu erfassen, wird das Elementevaluationskonzept wie folgt definiert:

Element evaluation concept: The evaluation concept for an element of a digital solution.

Ähnlich wie beim Systemevaluationskonzept handelt es sich beim Elementevaluationskonzept um ein gemeinsames Konzept der Gestaltungs-, Konstruktions- und Realisierungsaktivitäten. Der Umfang der Elementevaluation umfasst das jeweilige Element und die Interaktion mit den Elementen, die in direktem Zusammenhang mit dem betrachteten Element stehen. Aus gestalterischer Sicht umfasst die Elementevaluation die folgenden beiden Aspekte:

- *Evaluierung der Gestaltung eines bestimmten Elements*, zum Beispiel die Gestaltung der angebotenen Funktionen und die Gestaltung des User Interface. Diese Evaluierung kann anhand von Prototypen (z. B. User Interface Mock-ups) oder anhand eines bereits implementierten Teils des Elements (z. B. Usability-Test) erfolgen.
- *Evaluierung der ordnungsgemäßen Implementierung*, z. B. Prüfung, ob das Verhalten wie im Designkonzept definiert implementiert ist. Diese Evaluierung erfolgt in der Regel anhand von Testfällen, die auf den Designkonzepten basieren, und speziellen Testaktivitäten, die die Testfälle durchführen.

Aus der Sicht der Konstruktion und Realisierung umfasst die Elementevaluation verschiedene technische Aspekte. Ein wichtiger Aspekt ist die Prüfung der technischen Schnittstellen zu anderen Elementen des Systems. Diese Tests werden in der Regel in Form von Integrationstests durchgeführt. Eine weitere wichtige technische Perspektive ist die Evaluierung der detaillierten Implementierung (z. B. durch Unit-Tests) und die Bewertung technischer Qualitätskriterien, die sich z. B. auf die Wartbarkeit des Quellcodes beziehen (z. B. durch Code-Reviews oder statische Code-Analyse).

Die konkrete Vorgehensweise bei der Erstellung der Konzepte auf der Elementebene und bei der Realisierung der einzelnen Elemente hängt von dem gewählten Prozessmodell zum Bau der digitalen Lösung ab. Im Allgemeinen sollte vor Beginn der Umsetzung eine Vorbereitungsphase eingeplant werden. In dieser Phase werden die ersten Details der Elemente ausgearbeitet. Darüber hinaus werden vorbereitende Maßnahmen für die Realisierung durchgeführt, wie z. B. das Aufsetzen von Workshops, die Bereitstellung von Arbeitsbereichen sowie von Entwicklungswerkzeugen.

Die Erstellung einer ersten Version unterscheidet sich von der Entwicklung einer Lösung während des Betriebs

Sobald eine erste Version einer digitalen Lösung realisiert und in Betrieb ist, verlagert sich der Schwerpunkt des Bauprozesses für eine digitale Lösung erheblich.

Der erste Teil dieses Wandels besteht darin, dass die digitale Lösung im Betrieb Pflege und Wartung erfordert. Benutzer können Fehler melden, die behoben werden müssen, oder zusätzliche Funktionen anfordern. Dieser Wartungsaufwand wird oft unterschätzt und führt zu einem Konflikt zwischen der Wartung und der weiteren Entwicklung einer digitalen Lösung.

Der zweite Teil dieses Wandels besteht darin, dass bei jeder Entscheidung über die Weiterentwicklung einer digitalen Lösung berücksichtigt werden muss, dass es bereits eine bestehende Lösung gibt. Aus gestalterischer Sicht birgt dieser Wandel Chancen und Risiken. Zu den Änderungen gehört auch die Einholung von Feedback von echten Benutzern oder Kunden

in realen Umgebungen. Dieses Feedback kann genutzt werden, um eine digitale Lösung zu verstehen und weiter zu verbessern. Andererseits besteht das Risiko, dass sich die Benutzer oder Kunden an die Änderung der digitalen Lösung anpassen müssen. Darüber hinaus muss die Entwicklung einer digitalen Lösung die bestehenden technischen Strukturen und Randbedingungen berücksichtigen.

So erfordert beispielsweise die Änderung einer digitalen Lösung eine Aktualisierung der Softwareteile oder sogar den Austausch bestehender Geräte. Eine solche Aktualisierung oder Ersetzung kann je nach Art der digitalen Lösung erheblichen Arbeits- und Planungsaufwand erfordern. Eine weitere Herausforderung stellen Aktualisierungen und Änderungen bestehender Technologien dar, die zum Bau oder Betrieb einer digitalen Lösung verwendet werden. Wenn beispielsweise das Betriebssystem eines Smartphones aktualisiert wird, kann eine Änderung erforderlich sein, um die digitale Lösung betriebsbereit zu halten.

Lösungen im Betrieb bieten weitere Möglichkeiten für das Design

Schließlich ermöglicht eine bestehende digitale Lösung im Betrieb zusätzliche Methoden, die für die weitere Gestaltung einer digitalen Lösung relevant sind (vgl. z. B. [SaLe2016]):

- *Datengesteuerte Designentscheidungen:* Es ist möglich, explizit Funktionen zur Messung des Verhaltens von Benutzern oder Kunden einzubauen, um mehr über deren Bedürfnisse zu erfahren. So lässt sich beispielsweise die Zeit messen, die ein Benutzer in einer App verbringt, um die Attraktivität einer bestimmten App zu bewerten.
- *A/B-Tests:* A/B-Tests sind ein experimenteller Ansatz zur Bewertung von Designalternativen. Zwei alternative Funktionen werden realisiert und nach dem Zufallsprinzip verschiedenen Benutzer- oder Kundengruppen angeboten. Die digitale Lösung misst bestimmte Daten über das Benutzerverhalten, die zur Bewertung und zum Vergleich der Designalternativen herangezogen werden können.

Beide Methoden beruhen auf der Erfassung detaillierter Benutzer-/Kundendaten. Daher müssen diese Technologien mit Vorsicht angewendet werden und erfordern eine eingehende Prüfung der Rechtsvorschriften zum Datenschutz und zum Schutz der Privatsphäre, um nicht gegen bestehende Gesetze zu verstoßen.

2.1.2.4 Zur Gleichwertigkeit von Scoping/Konzeptions- und Entwicklungsarbeit

Der Begriff Upfront wird für Scoping oder Konzeptarbeit verwendet und hat in der Literatur oft einen negativen Beigeschmack (siehe z. B. [Meye2014]). Es ist zu erwarten, dass ein DDP sich mit der Diskussion um diesen Begriff befassen muss. Daher ist eine klare Position in dieser Frage erforderlich. Die Scoping- und Konzeptionsschritte im Digital Design sind ein Muss, denn nur so lassen sich relevante Informationen gewinnen, die für eine effektive und effiziente Entwicklung notwendig sind.

Ein Prozess auf Basis von Konzeptarbeit, der sich auf Design und Konstruktion konzentriert, ist kostengünstig und schnell. Neue Erkenntnisse können einfach und mit geringem Aufwand in Konzepte übernommen werden. Ein umsetzungsorientierter Prozess ist in der Regel teuer und langsam. Realisierungsteams (z. B. Softwareentwicklungsteams, vgl. [SeRP2017]) benötigen ständigen Input, um produktiv zu bleiben. Die Einarbeitung neuer Erkenntnisse und die Korrektur von schwerwiegenden Fehlern ist aufwendig. Bereits gebaute Teile müssen mit eventuellem Mehraufwand geändert werden.

Das bedeutet nicht, dass eine digitale Lösung bis ins kleinste Detail ausgearbeitet sein muss, bevor mit der eigentlichen Umsetzung begonnen werden kann. Andererseits ist es auch kein nachhaltiges Vorgehensmodell, alle Details einer digitalen Lösung erst kurz vor der Implementierung zu entwerfen.

Der Unterschied zwischen konzeptgetriebenen und realisierunggetriebenen Prozessen ist kein Argument dafür, so lange wie möglich in der Konzeptarbeit zu verbleiben, um möglichst umfassende und evaluierte Konzepte zu erarbeiten. Sie ist als Warnung gedacht, dass der Beginn der Realisierung kostspielig ist und dass die wichtigen und kostspieligen Details geklärt werden müssen, bevor die Realisierung beginnt. Das Erkennen dieser wichtigen Details erfordert Erfahrung und Ausbildung.

Es gibt viele Aspekte einer digitalen Lösung, die vor der Implementierung gestaltet und validiert werden können, insbesondere mit Hilfe von Prototypen (siehe 2.3). Es gibt ebenso viele Aspekte einer digitalen Lösung, die sinnvoll und effizient anhand der implementierten Lösung validiert werden können. Die eigentliche Herausforderung besteht darin, zu entscheiden, zu welcher Kategorie ein Aspekt gehört, und diese Entscheidung zu treffen, erfordert eine Menge Fachwissen (vgl. [Rein1997]).

2.1.3 Qualität als Querschnittsaspekt im Bauprozess

2.1.3.1 Ganzheitliche Betrachtung der Qualität im Bauprozess

Die Qualität einer digitalen Lösung wird durch verschiedene Aspekte bestimmt. Dazu gehören die Technologie, die für den Bau des digitalen Systems gewählt wurde, der Prozess, der für die Entwicklung der digitalen Lösung verwendet wurde und das Verständnis von Qualität in einem bestimmten Kontext (z. B. Projektumfeld, kulturelle Aspekte). Unseres Wissens gibt es keine ganzheitlichen Qualitätsmodelle, die all diese Aspekte berücksichtigen, allerdings machen Wissenschaftler auf den Bedarf an ganzheitlicheren Ansätzen aufmerksam [BrDP2005].

Ziel dieses Abschnitts ist es, ein Bewusstsein für die Bedeutung von Qualität und den umfassenden Einfluss verschiedener Aspekte auf die Qualität zu schaffen. Dieses Bewusstsein ist erforderlich, um die Qualität der digitalen Lösung während des gesamten Bauprozesses aktiv zu steuern und zu gestalten und eine hochwertige digitale Lösung aufzubauen.

Wichtige Aspekte, die für eine ganzheitliche Betrachtung der Qualität während des Bauprozesses zu berücksichtigen sind, beinhalten:

Qualität als Haltung

Alle nachstehend genannten Aspekte sind notwendig, um eine gute Qualität zu erzielen. Wir möchten jedoch betonen, dass gute Qualität vor allem von der Grundhaltung aller am Aufbau einer digitalen Lösung beteiligten Personen abhängt. Das liegt daran, dass die beteiligten Personen die zahlreichen kleinen, detaillierten Entscheidungen treffen, die letztlich die Gesamtqualität einer Lösung ausmachen. Wenn die beteiligten Personen eine positive Haltung zur Qualität haben, werden sie diese Entscheidungen im Sinne einer guten Qualität treffen. Wir haben die grundlegenden Elemente dieser Haltung in den zehn Prinzipien für gutes Digital Design formuliert (siehe Kapitel 6).

Das Bewusstsein, dass die Prozessqualität die Produktqualität beeinflusst

Obwohl das Ziel der meisten Unternehmen darin besteht, *hohe Qualität* zu liefern, müssen sie sich bewusst sein, dass die Qualität des Bauprozesses einen erheblichen Einfluss auf die Qualität

der digitalen Lösung und des digitalen Systems haben kann. Standardisierte und sich ständig verbessernde Prozesse [Demi2000] ermöglichen die Planung von Qualität. Das bedeutet auch, dass die Schaffung einer hochwertigen digitalen Lösung eine gemeinsame und ganzheitliche Aufgabe im Bauprozesses ist: Alle Tätigkeitsbereiche müssen zusammenarbeiten, um ein hochwertiges digitales System zu liefern, das die digitale Lösung realisiert. Qualitätskontrolle und -management sowie die kontinuierliche Evaluierung der erstellten Artefakte müssen zentrale Aspekte des zugrundeliegenden Prozesses sein, wie in Abschnitt 2.1 beschrieben.

Aktives Qualitätsmanagement und Verständnis verschiedener Qualitätsaspekte

Die Verbesserung der Qualität einer bereits realisierten digitalen Lösung kann erheblichen Aufwand erfordern. Daher muss die gewünschte Qualität von Beginn des Bauprozesses an definiert und verfolgt werden. Dazu gehört die Berücksichtigung verschiedener Perspektiven (d.h. technische, ökologische, bauprozessbezogene und soziale), wie in Abschnitt 2.1.2.1 erläutert, sowie die Unterscheidung zwischen Qualitätsmerkmalen, die für die Benutzer sichtbar und erfahrbar sind, und solchen, die den Benutzern verborgen bleiben (siehe Abschnitt 2.1.3.3). Darüber hinaus muss ein DDP die Qualitätsmerkmale sowohl für die digitale Lösung (siehe Abschnitt 2.1.3.5) als auch für das digitale System (siehe Abschnitt 2.1.3.4) kennen und verstehen, um entsprechend planen zu können.

Berücksichtigung von Risiko und Wert

Ziel eines DDP sollte es sein, den Kunden/Benutzern einer digitalen Lösung ein angemessenes und erwartetes Qualitätsniveau zu bieten, damit sie die digitale Lösung akzeptieren und annehmen. Qualität gibt es nicht umsonst, und oft bedeutet ein höheres Qualitätsniveau auch, dass die Entwicklung der digitalen Lösung teurer ist, was letztendlich zu höheren Kosten für die Kunden/Benutzer führt. Eine angemessene Berücksichtigung von Risiko und Wert ist daher wichtig. Dazu gehört auch das Verständnis dafür, auf welcher Detailebene ein Qualitätsaspekt beschrieben werden muss und welche Form der Dokumentation - einschließlich der Definition angemessener Testmetriken - erforderlich ist, um das Risiko zu verringern, dass eine digitale Lösung entwickelt wird, die die Bedürfnisse der Kunden/Benutzer nicht erfüllt [Glin2008].

Unterscheidung der Qualität der digitalen Lösung und der Qualität eines digitalen Systems

Für einen DDP ist es wichtig, den Unterschied zwischen der digitalen Lösung und dem digitalen System im Allgemeinen zu verstehen. Das gilt auch für die Qualität. Die Berücksichtigung dieses Unterschieds, aber auch das Bewusstsein, dass die Eigenschaften des digitalen Systems die digitale Lösung beeinflussen, ermöglichen es, die Qualität zu planen und z. B. geeignete Technologien zur Realisierung eines digitalen Systems auszuwählen (siehe Abschnitt 3.1). Dieser Aspekt wird im folgenden Abschnitt ausführlicher beschrieben.

2.1.3.2 Der Unterschied zwischen der Qualität einer digitalen Lösung und der Qualität eines digitalen Systems

Die Qualität einer digitalen Lösung hat einen wesentlichen Einfluss auf ihre Akzeptanz und ihren Erfolg. Dennoch sind die digitale Lösung und das digitale System nicht identisch. Das digitale System ist die Instanziierung der technischen Aspekte (d. h. der Hard- und Software) der digitalen Lösung und ist daher nur ein Teil der digitalen Lösung (siehe Abschnitt 1.2). Es gibt Qualitätsmerkmale einer digitalen Lösung, die unabhängig von den Qualitätsmerkmalen des digitalen Systems sind. Die Qualitätsmerkmale einer digitalen Lösung können sich jedoch auf die Qualitätsmerkmale des digitalen Systems auswirken und diese beeinflussen. Umgekehrt werden

die Qualitätsmerkmale eines digitalen Systems Teil der Qualitätsmerkmale einer digitalen Lösung und ergänzen diese. Dies gilt insbesondere für wahrnehmbare Qualitätsmerkmale, die sichtbar und erfahrbar sind (siehe Abschnitt 2.1.3.3).

Ein Online-Hotelbuchungsservice für Touristen ist zum Beispiel eine digitale Lösung. Diese digitale Lösung könnte zum Beispiel durch eine Smartphone-App realisiert werden, die die Funktionen für die Online-Buchung eines Hotels bietet. Diese App ist das entsprechende digitale System für den Hotelbuchungsservice. Der Online-Buchungsdienst selbst hat seine eigenen Qualitätsmerkmale, wie z. B. die Möglichkeit, Hotels in der ganzen Welt zu buchen und nach Hotels in verschiedenen Ländern zu suchen (siehe hedonische Qualitäten in Abschnitt 2.1.3.5). Diese Qualitätsmerkmale können unabhängig voneinander definiert werden, ohne ein bestimmtes digitales System mit seinen Qualitätsmerkmalen im Sinn zu haben. Dies könnte jedoch Auswirkungen auf die Anforderungen an ein digitales System haben. Das entsprechende digitale System kann seine eigenen Qualitätsmerkmale haben. Zum Beispiel sollte die Buchungs-App einfach zu bedienen sein, eine gute Leistung aufweisen und eine gute Ästhetik bieten. Diese Qualitätsmerkmale eines digitalen Systems tragen zur besseren Wahrnehmung und Qualität der digitalen Lösung bei.

2.1.3.3 Wahrnehmbare und zugrundeliegende Qualitätsattribute

Bei den Qualitätsmerkmalen ist es wichtig, zwischen Qualitätsmerkmalen zu unterscheiden, die von den Benutzern wahrgenommen werden können, und solchen, die den Benutzern verborgen sind. Die Qualität eines digitalen Systems umfasst sowohl wahrnehmbare Qualitäten (auch äußere Qualitäten genannt) als auch zugrundeliegende Qualitäten (auch innere Qualitäten genannt):

- Wahrnehmbare Qualitätsattribute sind für den Benutzer sichtbar oder können von ihm erlebt werden (z. B. Benutzbarkeit, Zuverlässigkeit). Sie schaffen Mehrwert für die Benutzer, helfen bei der Erfüllung von Anforderungen und fördern die Akzeptanz des Systems.
- Zu den zugrundeliegenden Qualitätsmerkmalen gehören Eigenschaften, die dem Benutzer verborgen bleiben (z. B. Wartbarkeit), die es den Entwicklern aber ermöglichen, das System mit geringen Kosten weiterzuentwickeln und zu warten.

Obwohl die Unterscheidung zwischen wahrnehmbaren und zugrundeliegenden Qualitätsmerkmalen eine bessere Planung und Prüfung der Qualität ermöglicht, ist es wichtig zu beachten, dass interne Qualitätsmerkmale die externen beeinflussen können [Mcco2004]. Wenn beispielsweise der Code eines digitalen Systems für die Entwickler nicht leicht zu verstehen ist, erschwert dies das Beheben von Fehlern und die Wartung des Systems (zugrundeliegende Qualität). Dies kann sich negativ auf die Zuverlässigkeit der digitalen Lösung auswirken, die von den Benutzern wahrgenommen wird.

Die Unterscheidung zwischen wahrnehmbarer und zugrundeliegender Qualität ermöglicht auch die Entwicklung einer angemessenen Bewertungsstrategie, die beide Perspektiven abdeckt [FrPr2009]. Um wahrnehmbare Qualitätsattribute zu bewerten, müssen die Benutzer des Systems einbezogen werden, während die zugrundeliegenden Qualitäten mit Hilfe von automatisierten Tests bewertet werden können, um sicherzustellen, dass z. B. ein Stück Code (eine Einheit/Unit) seinem Design entspricht und sich wie vorgesehen verhält. Diese Unit-Tests können von den Entwicklern ausgeführt werden und benötigen keine Beteiligung der Benutzer, um ausgeführt zu werden.

2.1.3.4 Ein Qualitätsmodell für digitale Systeme

Es gibt mehrere Modelle zur Definition der Qualität eines digitalen Systems (z. B. McCall [McRW1977], Boehm [BBKL1978], ISO/IEC 9126 [ISO2001], ISO/IEC 25010 [ISO2011]). Dies gilt auch für Ansätze und Modelle, die nicht-funktionale Anforderungen beschreiben, die häufig Qualitätsanforderungen als eine Hauptkategorie enthalten [Glin2007].

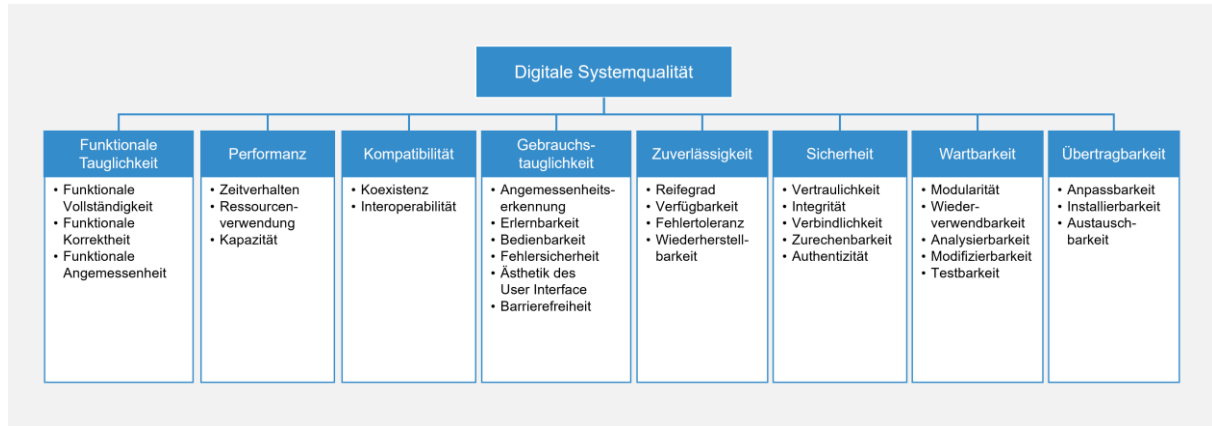


Abbildung 11 - Angepasste Qualitätsmerkmale nach ISO/IEC 25010 [ISO2011]

Definition der Systemqualitätsmerkmale nach [ISO2011]:

Funktionale Tauglichkeit: Dieses Merkmal gibt an, inwieweit ein Produkt oder System Funktionen bietet, die den angegebenen und implizierten Bedürfnissen entsprechen, wenn es unter bestimmten Bedingungen verwendet wird.

Performanz: Dieses Merkmal stellt die Leistung im Verhältnis zur Anzahl der verwendeten Ressourcen unter den angegebenen Bedingungen dar.

Kompatibilität: Grad, in dem ein Produkt, ein System oder eine Komponente Informationen mit anderen Produkten, Systemen oder Komponenten austauschen und/oder die erforderlichen Funktionen ausführen kann, während es dieselbe Hardware- oder Softwareumgebung nutzt.

Gebrauchstauglichkeit (Usability): Grad, in dem ein Produkt oder System von bestimmten Benutzern verwendet werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend in einem bestimmten Nutzungskontext zu erreichen.

Zuverlässigkeit: Grad, zu dem ein System, ein Produkt oder ein Bauteil bestimmte Funktionen unter bestimmten Bedingungen über einen bestimmten Zeitraum hinweg erfüllt.

Sicherheit: Grad, in dem ein Produkt oder System Informationen und Daten schützt, so dass Personen oder andere Produkte oder Systeme den ihrer Art und Berechtigung entsprechenden Grad des Datenzugriffs erhalten.

Wartbarkeit: Dieses Merkmal steht für den Grad an Effektivität und Effizienz, mit dem ein Produkt oder System verändert werden kann, um es zu verbessern, zu korrigieren oder an Veränderungen in der Umgebung und bei den Anforderungen anzupassen.

Übertragbarkeit: Grad der Effektivität und Effizienz, mit dem ein System, ein Produkt oder eine Komponente von einer Hardware-, Software- oder einer anderen Betriebs- oder Nutzungsumgebung auf eine andere übertragen werden kann.

Wenn es um Qualitätsmerkmale geht, ist die ISO/IEC 25010 ein guter Ausgangspunkt für das Verständnis der Qualität eines digitalen Systems. Sie definiert acht Hauptqualitätsmerkmale, die

auf digitale Systeme angewendet werden können. Jedes dieser Merkmale umfasst mehrere Untermerkmale (siehe Abbildung 11); insgesamt sind es 31 Untermerkmale [ISO2011]. ISO/IEC 25010 nennt die Benutzbarkeit als eine der wichtigsten Eigenschaften eines digitalen Systems, die sich vor allem darauf bezieht, inwieweit ein System einfach zu benutzen ist. Allerdings wird User Experience nicht ausdrücklich als Qualitätsmerkmal erwähnt, was in anderen Normen wie ISO/IEC 9241-210 [ISO2019] oder dem CUE-Modell [ThMa2007] berücksichtigt wird.

2.1.3.5 Beschreibung eines Qualitätsmodells für digitale Lösungen

Bei digitalen Lösungen plädieren wir für eine umfassendere und ganzheitliche Sichtweise der Qualität, bei der Aspekte der User Experience (UX) wie Zufriedenheit, Vergnügen und Freude berücksichtigt werden. Diese Sichtweise kann durch viele Faktoren beeinflusst werden, die nicht unbedingt Teil des digitalen Systems sind, wie z. B. der Ruf der Marke. Das Qualitätsmodell von Hassenzahl et al. [HaTr2006] ist ein guter Ausgangspunkt: Es unterscheidet zwischen pragmatischen und hedonischen Qualitäten. Pragmatische Qualitäten konzentrieren sich auf die effektive und effiziente Unterstützung von Aufgaben. Insgesamt beziehen sie sich auf die Benutzbarkeit und den Nutzen einer digitalen Lösung. Das bedeutet auch, dass das digitale System und seine wahrnehmbaren Qualitätsmerkmale die pragmatischen Qualitäten einer digitalen Lösung stark beeinflussen.

Obwohl sich beide Qualitäten auf den Benutzer konzentrieren, gehen die hedonischen Qualitäten über die Qualitäten digitaler Systeme (2.1.3.4) hinaus und bieten eine neue Perspektive auf die Qualitäten einer digitalen Lösung. Die wahrgenommene hedonische Qualität konzentriert sich auf die subjektive Wahrnehmung und die Empfindungen, die durch die Nutzung der digitalen Lösung hervorgerufen werden. Das Verständnis der positiven Emotionen des Benutzers ermöglicht weitere Untersuchungen. Dazu gehört z. B. die Frage, inwieweit die digitale Lösung den Benutzer motiviert oder begeistert. Wird sie sogar das Leben des Benutzers verändern? Solche Aspekte sind eher subjektiv und nur schwer in den Griff zu bekommen, aber sie können den Benutzer wirklich beeinflussen und ihm eine Bindung zur digitalen Lösung verschaffen. Studien zeigen, dass Benutzer klar zwischen pragmatischen und hedonischen Qualitäten unterscheiden können und dass Benutzer die Attraktivität einer digitalen Lösung auf der Grundlage beider Qualitäten insgesamt beurteilen. Das bedeutet auch, dass im Grunde beide Qualitätsaspekte gleich wichtig sind. Abhängig von der jeweiligen digitalen Lösung und den Bedürfnissen und Anforderungen der unterschiedlichen Stakeholder kann jedoch der eine oder andere Aspekt eine wichtigere Rolle spielen. Der Fragebogen AttrakDiff [HaTr2006] wurde entwickelt, um die verschiedenen Produkteigenschaften zu bewerten und ein Gesamturteil über die Attraktivität einer digitalen Lösung abzugeben (siehe Abschnitt 4.1).

Die Qualitätsaspekte, die für die digitale Lösung berücksichtigt werden, können sogar über die User Experience (UX) hinausgehen und auch die Customer Experience (CX) berücksichtigen, die alle Interaktionen eines Kunden mit einem Unternehmen umfasst. Dies kann zum Beispiel einen Anruf bei der Support-Hotline des Unternehmens beinhalten.

Abschließend können wir zusammenfassen, dass die Benutzbarkeit ein wichtiger Aspekt der User Experience ist, die wiederum ein wichtiger Aspekt der Customer Experience ist. Beachten Sie, dass Customer Experience und User Experience nicht identisch sind, insbesondere wenn der Kunde nicht gleichzeitig der Benutzer ist (siehe Abschnitt 1.2.3).

2.1.4 Zusätzliche Ressourcen für den Bauprozess

Die in diesem Handbuch vermittelten Kompetenzen definieren den Foundation Level, was bedeutet, dass die vermittelten Fähigkeiten zwar notwendige Grundlagen sind, aber nicht ausreichen, um das gesamte Spektrum der für die Gestaltung digitaler Lösungen erforderlichen Fähigkeiten abzudecken.

Ein DDP muss sich daher der Tatsache bewusst sein, dass zusätzliche Fähigkeiten erforderlich sind.

Beispiele hierfür sind:

- Requirements Engineering, zum Verstehen, Validieren und Verwalten von Anforderungen komplizierter Lösungen
- Business-Analyse, zum Verstehen und Bewerten von geschäftsorientierten Lösungen
- Industriedesign, zur Gestaltung von physischen Geräten als Teil einer digitalen Lösung
- Usability Engineering, zur Gestaltung und Bewertung des interaktiven Teils digitaler Lösungen
- Software-Testing, zur systematischen Qualitätssicherung der Softwareteile einer digitalen Lösung

Diese Liste ist natürlich nicht vollständig und umfassend. Sie bietet jedoch einen Überblick über mehrere verwandte Bereiche und verfolgt zwei Ziele:

1. Verweis auf weitere wichtige Bereiche, die ein DDP auf Foundation Level Niveau kennen sollte
2. Schaffen von Demut in Anbetracht der vielen verschiedenen Disziplinen und Verdeutlichung, dass ein DDP auf eine breite Palette von Fähigkeiten zurückgreifen kann, um eine digitale Lösung zu entwerfen

Eine Person kann durchaus zahlreiche Kompetenzen in diesen Bereichen haben. Es sollte jedoch klar sein, dass niemand alle Bereiche in der erforderlichen Tiefe beherrschen wird. Die unmittelbare Schlussfolgerung, die wir daraus ziehen können, ist, dass jeder Bauprozess Teamarbeit erfordert und dass die richtigen Kompetenzen zum richtigen Zeitpunkt eingebunden werden müssen (siehe Abschnitt 6.3). Im Hinblick auf die Gestaltung einer digitalen Lösung ist es die Aufgabe eines DDP, dafür zu sorgen, dass Personen mit den notwendigen Fähigkeiten zur Verfügung stehen, um eine erfolgreiche digitale Lösung zu schaffen. Diese Verantwortung beginnt im Foundation Level.

2.1.5 Zusammenfassung: Ein idealisiertes Modell des Bauprozesses

In diesem Abschnitt haben wir die drei Schritte des Bauprozesses zusammen mit einer Einführung in Designprozesse vorgestellt. Eine typische Einsteigerfrage zu diesem Zeitpunkt lautet wie folgt: Ich kenne jetzt die Schritte und etwas über einen Designprozess, aber wie sieht der Bauprozess in der Realität aus? Die Antwort auf diese Frage ist für Einsteiger in der Regel enttäuschend: Wir wissen es nicht im Detail. Die meisten Unternehmen passen dieses eher allgemeine Verfahren an ihre besonderen Bedürfnisse an. Die Definition und Anpassung von Bauprozessen ist eine Managementaufgabe und geht über den Rahmen von Digital Design hinaus.

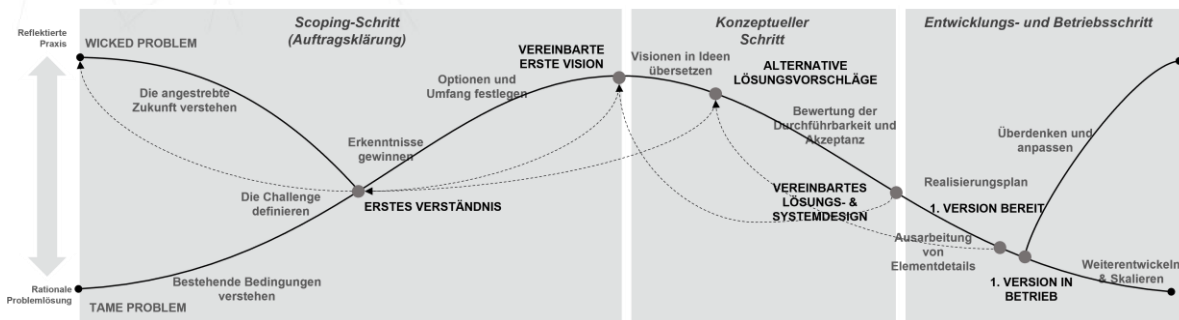


Abbildung 12 - Ein idealisiertes Modell des Bauprozesses

Wir sind uns jedoch bewusst, dass dies keine zufriedenstellende Antwort ist. Wir geben daher einen Überblick über einen idealisierten Bauprozess (siehe Abbildung 12 - Ein idealisiertes Modell des Bauprozesses), der alle in diesem Kapitel vorgestellten Inhalte zusammenfasst. Wir werden das Modell kurz beschreiben, beginnend auf der linken Seite mit dem Scoping-Schritt.

Arbeiten im Scoping-Schritt

Zu Beginn des Bauprozesses befinden wir uns in einer Situation, in der wir mit einem abstrakten und offenen Raum von Möglichkeiten konfrontiert sind. Um die Ausgangssituation zu vereinfachen, ist die Unterscheidung zwischen Tame Problems und Wicked Problems [RiWe1973] nützlich, um zwei verschiedene Ausgangssituationen einen Bauprozess zu definieren. Ein Wicked Problem kann wie folgt definiert werden:

Wicked problem: A problem that is difficult or impossible to solve because of incomplete, contradictory, and changing requirements.

Ein Tame Problem ist das Gegenteil eines Wicked Problems:

Tame problem: A problem that is well defined with clear and stable requirements.

Beispiel YPRC. Die YPRC-Fallstudie ist ein Beispiel für ein „wicked problem“: „Entwickeln Sie ein neues Coaching-Erlebnis für Langstreckenläufer.“ Der „Wicked“ Aspekt dieses beispielhaften Problems wird vor allem durch den Teil „neues Coaching-Erlebnis“ hervorgerufen, da niemand wirklich weiß, was neu bedeutet. Ein Beispiel für ein Tame Problem ist die „Messung der Gesundheitsdaten eines Läufers während einer Trainingseinheit“. Gesundheitsdaten können auf analytische Weise definiert werden. Darüber hinaus ist auch gängige Messtechnologie verfügbar.

Wenn man mit einem Wicked Problem konfrontiert ist, ist der Modus der reflektierenden Praxis (siehe Abschnitt 2.1.2) die richtige Arbeitsweise. Wir müssen die angestrebte Zukunft gemeinsam mit allen relevanten Stakeholdern verstehen, um ein erstes Verständnis darüber zu erlangen, worum es bei der digitalen Lösung geht. Werkzeuge für dieses Verständnis sind frühe Konzepte (z. B. Skizzen) und Prototypen (z. B. Papierprototypen oder Storyboards). Die in Abschnitt 2.1.2.1 vorgestellten Scoping-Dimensionen sind ein geeignetes Instrument zur Analyse der verschiedenen Aspekte des Wicked Problems. Bei einem Tame Problem ist der rationale Problemlösungsmodus die richtige Arbeitsweise. Wir müssen die gegebenen Bedingungen verstehen, unter denen das Problem existiert, und zwar aus zwei Gründen: Erstens, um zu verstehen, was erreicht werden muss, und zweitens, um sicherzustellen, dass es sich nicht um ein verstecktes Wicked Problem handelt.

Unabhängig von der Art des Problems, mit dem wir konfrontiert sind, präzisieren wir unser Verständnis des Gesamtproblems und versuchen, ein erstes gemeinsames Verständnis davon zu erlangen, was wir eigentlich erreichen wollen. Wenn es nicht möglich ist, dieses gemeinsame Verständnis zu erreichen, muss der Scoping-Schritt unter der Annahme, dass wir es mit einem Wicked Problem zu tun haben, von vorne beginnen.

Im gegenseitigen Einvernehmen arbeiten wir wieder im Modus der reflektierenden Praxis und beginnen, unser Verständnis des Geltungsbereichs weiter zu konkretisieren. Dazu wird das Problem aus verschiedenen Blickwinkeln betrachtet (z. B. durch die Betrachtung von Wettbewerbern oder die Analyse digitaler Technologien, die zur Lösung des Problems beitragen könnten). Am Ende dieses Prozesses wird eine konkrete und abgestimmte Vision für die digitale Lösung, in Form eines Design Briefs definiert. Wenn es nicht möglich ist, diese Einigung unter den relevanten Stakeholdern zu erzielen, muss eine weitere Iteration des Bauprozesses begonnen werden, um eine neue Vision zu definieren. In Abbildung 12 - Ein idealisiertes Modell des Bauprozesses sind die Iterationen durch gestrichelte Linien dargestellt. Die Visualisierung dieser Iterationen beschreibt den spätesten Zeitpunkt, an dem die Notwendigkeit einer Iteration erkannt werden kann.

Arbeiten im konzeptuellen Schritt

Mit einer abgestimmten ersten Vision (und den anderen Details aus dem Digital Design Brief) kann der konzeptuelle Schritt beginnen. Wenn wir mit dem konzeptuellen Schritt beginnen, befinden wir uns wieder in einer abstrakten und offenen Situation, da es mehrere alternative Möglichkeiten gibt, die Vision zu erreichen. Der Modus der reflektierenden Praxis ist in dieser Situation die richtige Arbeitsweise, da wir die Vision in verschiedene Lösungsideen umsetzen müssen (siehe Abschnitt 2.1.1.2). Dazu können wir erste Konzepte entwickeln und grundlegend unterschiedliche Prototypen erstellen, um die unterschiedlichen Lösungsrichtungen zu erkunden. Einzelheiten dazu finden Sie in den Abschnitten 0 und 2.3.

Das Verständnis für die unterschiedlichen Lösungsideen wird während dieser Arbeit immer konkreter. Ein wichtiges Ergebnis dieser Arbeit sind alternative Lösungsvorschläge. Die Entwicklung und Evaluierung von alternativen Lösungsvorschlägen ist ein wichtiger Ansatz für die Arbeit im Design. Durch alternative Ideen können wir den möglichen Lösungsraum systematisch erkunden und die Wahrscheinlichkeit erhöhen, eine gute Lösungsidee zu identifizieren. Wenn die entwickelten Lösungsideen nicht vielversprechend genug sind, sollte die ursprüngliche Vision in Frage gestellt werden, und der Bauprozess sollte zum Scoping-Schritt zurückkehren, um eine neue Vision für die digitale Lösung zu entwickeln.

Während der Konkretisierung wechselt die Arbeitsweise von der reflektierenden Praxis zum rationalen Problemlösungsmodus. Die verschiedenen Lösungsideen sollten in Systemideen umgesetzt werden. Die Machbarkeit und Akzeptanz dieser Ideen müssen dann mit den relevanten Stakeholdern evaluiert werden. Am Ende dieses Prozesses werden erste abgestimmte Lösungs- und Systemdesigns für die digitale Lösung erstellt, die ausreichend detailliert sind, um das Risiko des Entwicklungsbeginns zu akzeptieren. Wenn es nicht möglich ist, ein solches Design zu erreichen, muss die Entwicklung alternativer Ideen erneut beginnen.

Arbeiten im Entwicklungs- und Betriebsschritt

Obwohl die vereinbarten Lösungs- und Systemdesigns aus der Perspektive des konzeptuellen Schritts konkret sind, befinden wir uns erneut in einer abstrakten und offenen Situation, da nun die verschiedenen Realisierungsdetails des Systems zusammen mit einem ersten Plan für die

Realisierung der digitalen Lösung definiert werden müssen. In dieser Situation ist der rationale Problemlösungsmodus wieder die richtige Arbeitsweise, da viele detaillierte Designentscheidungen getroffen werden müssen, um die verschiedenen Details der digitalen Lösung auszuarbeiten und zu bewerten. Für digitale Lösungen ist dieser Teil des Prozesses eine echte Herausforderung und erfordert Fachleute aus verschiedenen Bereichen (siehe 2.1.4).

Am Ende dieser Arbeit steht, idealisiert betrachtet, eine erste betriebsbereite Version der digitalen Lösung. Jetzt ist eine kritische Entscheidung erforderlich: Ist diese Lösung einsatzbereit oder nicht? Wenn die Antwort nein lautet, geht der Prozess zurück zum konzeptuellen Schritt, da die Gesamtlösungsidee nicht angemessen ist, und neue Lösungsideen entwickelt werden müssen. Dieser Schritt zurück scheint ziemlich radikal zu sein, ist aber die einzig richtige Antwort, da die Änderung bestimmter Details der Lösungsidee natürlich ein Teil des Bauprozesses ist. Wenn es nicht möglich ist, die ausgewählte Lösungsidee in eine betriebsbereite Form zu bringen, ist die einzig richtige Antwort, neue Lösungsideen zu schaffen.

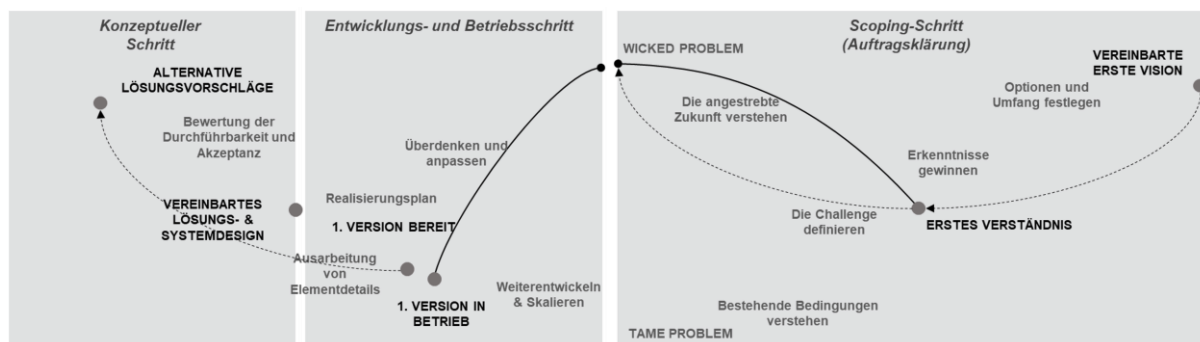


Abbildung 13 - Das Ende des Bauprozesses ist ein neuer Anfang

Sobald die Lösung in Betrieb ist, sind wir in der Lage, Feedback von echten Kunden im realen Leben zu erhalten. Unser Verständnis für das Erreichen der Lösung wird nun immer deutlicher. Im Allgemeinen gibt es nun zwei Möglichkeiten: Die eine besteht darin, dass die Lösung von den Kunden akzeptiert wird und den gewünschten Wert generiert, sodass die Lösung weiterentwickelt werden kann und hoffentlich skaliert, um erfolgreich zu werden. Die andere Möglichkeit ist, dass die Lösung nicht angenommen wird und die Unternehmensziele nicht erfüllt, obwohl alle vorangegangenen Evaluierungsmaßnahmen darauf hindeuteten, dass die Kundenakzeptanz sehr wahrscheinlich ist. Wir betrachten eine solche Situation als eine Art Wicked Problem, da wir unser gesamtes Verständnis der Lösung überdenken müssen. Unabhängig vom Ergebnis findet ein weiterer Scoping-Schritt statt und das idealisierte Modell des Bauprozesses kann erneut angewendet werden (siehe Abbildung 13).

2.2 Konzeptarbeit im Digital Design

Die Konzeptarbeit ist ein zentraler Aspekt der Arbeit des Digital Design. Um das breite Spektrum des Bauprozesses für eine digitale Lösung abzudecken, ist im Digital Design eine breite Palette verschiedener Konzepttypen erforderlich. Die verschiedenen Konzepttypen wurden in Abschnitt 2.1 als Teil des Bauprozesses vorgestellt.

Wenn man über Konzeptarbeit spricht, ist es wichtig, den Unterschied zwischen der inhaltlichen Perspektive und der Perspektive des Erstellungsprozesses zu erkennen:

- Die *inhaltliche Perspektive* befasst sich mit dem Inhalt von Konzepten und den Beziehungen zwischen den verschiedenen Teilen eines Konzepts und den Beziehungen zwischen dem Inhalt verschiedener Konzepte (in diesem Kapitel näher erläutert).
- Die *Perspektive des Erstellungsprozesses* befasst sich mit der Erstellung eines oder mehrerer Konzepte als Teil des Bauprozesses (wird in Kapitel 5 näher erläutert).

Die Vermischung beider Perspektiven führt allerdings zu einem falschen Verständnis von Konzepten, nämlich dem Irrtum, dass mehrere unterschiedliche Konzepte einen *Wasserfallansatz* implizieren, bei dem ein Konzept (z. B. das Lösungsdesignkonzept) abgeschlossen sein muss, bevor die Arbeit an einem weiteren Konzept (z. B. dem Systemdesignkonzept) beginnen kann. Dieser sequenzielle Ansatz wird in der Praxis fast nie angewandt.

In diesem Kapitel konzentrieren wir uns auf die inhaltliche Perspektive, stellen Vorlagen für jeden Konzepttyp vor und veranschaulichen ihre Beziehungen aus einer inhaltlichen Perspektive. Das Verständnis der inhaltlichen Perspektive ist eine Voraussetzung für effiziente und effektive Konzeptarbeit. Die Perspektive des Erstellungsprozesses ist ein zweiter Aspekt und stellt eine noch größere Herausforderung dar als die inhaltliche Perspektive. In Kapitel 5 veranschaulichen wir die Perspektive des Erstellungsprozesses anhand eines beispielhaften Bauprozesses für Einsteiger.

Die Welt der Konzepte ist für Einsteiger im Digital Design aufgrund der unterschiedlichen Konzepttypen, Perspektiven und Abstraktionsebenen (Lösung, System und Element) meist überwältigend.

Im Folgenden beginnen wir mit den Grundlagen der Konzeptarbeit (siehe Abschnitt 2.1.1) im Digital Design und arbeiten uns dann über die Abstraktionsebenen bis zu den Details der Elementebene vor. Wenn wir dort angekommen sind, arbeiten wir uns wieder nach oben und fügen die Details zu einem Gesamtbild zusammen, das im letzten Abschnitt dieses Kapitels vorgestellt wird.

Unter 2.2.2 werden Dokumentvorlagen für die verschiedenen Abstraktionsebenen vorgestellt und die Gesamtstruktur auf der Grundlage der Inhalte der einzelnen Kapitel präsentiert. Um die Lesbarkeit zu verbessern, haben wir die Beschreibung der Struktur der Dokumentvorlage von der Beschreibung der detaillierten Dokumentationstechniken, die zur Erstellung der Inhalte für ein bestimmtes Kapitel der Dokumentvorlage angewendet werden können, getrennt.

Die Dokumentationstechniken werden getrennt für die Lösungsebene in Abschnitt 2.2.3 (Digital Design Brief und Lösungsdesignkonzept) und für die System-/Elementebene in Abschnitt 2.2.5 (System-/Software-/Gerätedesignkonzept) vorgestellt.

Der Hauptgrund für diese Trennung liegt in den erheblichen Unterschieden in der Struktur der Designkonzepte auf der Lösungsebene und der System-/Elementebene. Auf diese Unterschiede

wird in Abschnitt 2.2.4 näher eingegangen. Abschließend fügt Abschnitt 2.2.6 alle Einzelteile und Details zu einem Gesamtbild zusammen.

2.2.1 Grundlagen der Konzeptarbeit

2.2.1.1 Konzepte sind Ideen im Denken oder in der Kommunikation

Im Allgemeinen sind Konzepte Ideen, die in Gedanken oder in der (schriftlichen oder mündlichen) Kommunikation verwendet und als Elemente der Gedanken betrachtet werden (vgl. [MaLa2015]). Konzeptarbeit im Digital Design bedeutet daher, gedanklich an der Erstellung einer digitalen Lösung zu arbeiten, d.h. die Ziele der digitalen Lösung und die entsprechende Form, Funktion und Qualität einer digitalen Lösung zu definieren. Konzepte können in einer eher linearen verbalen/schriftlichen Form oder in einer stark strukturierten technischen Form auftreten.

Beispiel YPRC. Die folgende kurze Beschreibung der YPRC-Fallstudie kann als verbale konzeptuelle Beschreibung betrachtet werden: YPRC bietet einen ganzheitlichen Trainingsservice für Laufinteressierte und besteht aus einer speziellen Smartwatch, einer Smartphone-App und einem Portal. Die Smartwatch misst die Gesundheitsdaten des Läufers, die von der App visualisiert werden. Die App kommuniziert im Hintergrund mit einem Portal. Über das Portal kann der Läufer Coaching durch künstliche Intelligenz (KI) und persönliches Remote-Coaching erwerben.

Diese kurze verbale Beschreibung ist ein effizientes Instrument, um den verschiedenen Stakeholdern die wichtigsten Ideen von YPRC zu vermitteln. Solche verbalen Beschreibungen sind jedoch nur bedingt geeignet, um alle Details zu definieren und zu kommunizieren, die für den tatsächlichen Bau von YPRC relevant sind - dies erfordert differenzierte Konzepte, die die unterschiedlichen Aspekte einer digitalen Lösung auf strukturierte Weise darstellen.

2.2.1.2 Nutzen und Grenzen von Konzepten

Ein Designkonzept wird aus mehreren Gründen erstellt:

- *Denkwerkzeug zur Beherrschung vermeintlich komplexer Lösungen/Systeme:* Ein Designkonzept strukturiert die Gedanken über die digitale Lösung. Im Folgenden gehen wir auf diesen Aspekt ein, indem wir dedizierte Dokumentationstechniken beschreiben, die einerseits als Dokumentvorlagen dienen und gleichzeitig die Strukturierung der Gedanken über eine digitale Lösung, das entsprechende System und seine Elemente unterstützen. Dieser strukturierte Ansatz ist wichtig, um das System und die Lösung als Ganzes zu verstehen und die Teile der Lösung/des Systems zu identifizieren, die wirklich komplex oder *nur* kompliziert sind (siehe Abschnitt 1.2.1). In Abschnitt 2.2.5 werden wir auf diesen Punkt zurückkommen und die einzelnen Bausteine beschreiben, aus denen die gesamte digitale Lösung besteht.
- *Grundlage für Konstruktion und Realisierung:* Ein Designkonzept dient als Input für die Tätigkeitsbereiche Konstruktion und Realisierung (siehe Abschnitt 1.3). Im Gegensatz zur anforderungsgetriebenen Arbeit (siehe Abschnitt 0, z. B. aus dem Requirements- oder Usability-Engineering) nimmt die Konzeptarbeit im Digital Design die lösungsgetriebene Perspektive des Designs ein (vgl. [Cros2006]) und fokussiert auf konkrete Lösungsideen anstelle einer intensiven Definition und Analyse von Anforderungen, die zu einer Lösung führen. Das bedeutet nicht, dass die Anforderungen im Digital Design vernachlässigt werden. Soweit erforderlich, werden Anforderungen dokumentiert und verwendet, insbesondere Qualitätsanforderungen, Randbedingungen und Ziele. Darüber hinaus sind

die Anforderungen der Stakeholder, insbesondere die der Auftraggeber-Rolle und der Kunden-Rolle, ein entscheidender Faktor bei der Erstellung von Designkonzepten.

- *Kommunikationsmittel*: Ein Designkonzept kommuniziert die digitale Lösung an unterschiedliche Stakeholder. Um diesen Kommunikationszweck zu unterstützen, müssen verschiedene Arten von Designkonzepten erstellt werden. In Abschnitt 2.2.2 wird diese Frage nochmals aufgegriffen und erklärt, an welche Zielgruppen sich die verschiedenen Konzepte richten.
- *Externes Gedächtnis für komplizierte Lösungen/Systeme*: Ein Designkonzept dient während des gesamten Lebenszyklus einer digitalen Lösung als externes Gedächtnis. Der Grund dafür ist, dass die Menge an Informationen über eine typische digitale Lösung, die während des Bauprozesses entsteht, die Kapazität des menschlichen Gedächtnisses bei weitem übersteigt. Um als externes Gedächtnis zu dienen, müssen alle Designkonzepte ständig überarbeitet und optimiert werden. Damit lassen sich die typischerweise komplizierten Strukturen von Lösungen/Systemen mit vertretbarem Aufwand bewältigen.
- *Evaluationsinstrument*: Ein Designkonzept kann zur Bewertung der beschriebenen digitalen Lösung herangezogen werden, beispielsweise im Hinblick auf Geschäftsvorfälle, Kunden- oder Benutzerakzeptanz oder rechtliche Fragen.
- *Bezugspunkt für die Evaluierung*: Ein Designkonzept dient als Grundlage, um zu bewerten, ob die realisierte digitale Lösung den Bedürfnissen der Stakeholder entspricht, und um sicherzustellen, dass die digitale Lösung gemäß dem festgelegten Konzept gebaut wird. Daher werden auf jeder Abstraktionsebene zusätzliche Evaluationskonzepte erstellt (siehe Abschnitt 2.1).

Die Erstellung und Arbeit mit Designkonzepten ist kostengünstig, hat aber auch gewisse Grenzen:

- *Konzepte sind nie vollständig*: Selbst das beste und detaillierteste Designkonzept ist unvollständig. Der Grund dafür ist einfach: Das Leben ist zu kompliziert, als dass man es vorhersehen und vollständig in einem Konzept erfassen könnte. Diese Tatsache ist jedoch keine Schwäche, denn Konzepte sind ein Kommunikationsmittel. Viele Schwachstellen und fehlende Details von Designkonzepten werden vor allem dann festgestellt, wenn die Entwicklung einer digitalen Lösung beginnt. Ein DDP auf Foundation Level Niveau sollte die Tatsache akzeptieren, dass Konzepte unvollständig sind und während des gesamten Bauprozesses erweitert und überarbeitet werden. Diese Arbeit ist keineswegs eine Zeitverschwendung. Die ständige Arbeit an Designkonzepten ist ein wichtiges Rückgrat, um den Bauprozess aus gestalterischer Sicht unter Kontrolle zu halten, da die Designkonzepte als externes Gedächtnis dienen (siehe oben).
- *Konzepte lassen immer Raum für Interpretationen*: So wie Konzepte nie vollständig sind, können sie auch nie frei von Interpretationen sein. Die Interpretation ist ein zentrales Merkmal der menschlichen Kommunikation, und da Konzepte Kommunikationsmittel sind, müssen sie interpretiert werden. Ein DDP auf Foundation Level sollte dies immer im Hinterkopf behalten und stets auf mögliche Missverständnisse und Fehlinterpretationen von Konzepten achten.

- *Konzepte sind nicht die digitale Lösung:* Die Arbeit mit Konzepten ist eine der Hauptaufgaben im Digital Design. Konzepte sind jedoch ein Mittel zum Zweck. Konzepte sind Werkzeuge, die dem Bauprozess dienen (siehe Abschnitt 2.1.2). Das Ziel des Bauprozesses ist es, eine digitale Lösung zum Leben zu erwecken. Ein DDP auf Foundation Level Niveau sollte sich dessen bewusst sein und die Konzeptarbeit ernst nehmen, aber immer im Hinterkopf behalten, dass eine gute digitale Lösung wichtiger ist als gute Konzepte. In Kapitel 6 behandeln wir dieses Thema erneut, wenn wir die zehn Prinzipien eines guten Digital Designs vorstellen.
- *Ausgefeilte Konzepte können falsches Vertrauen schaffen:* Gute Designkonzepte mit High-Fidelity-Prototypen der digitalen Lösungen können sehr beeindruckend sein, insbesondere für unerfahrene Personen in der Auftraggeber-Rolle. Die Kehrseite dieses Eindrucks kann sein, dass Auftraggeber und Stakeholder ein falsches Vertrauen in den Erfolg und den aktuellen Stand einer digitalen Lösung bekommen und zu optimistisch über den potenziellen Erfolg der digitalen Lösung werden. Dieser Punkt ist kein Argument für die Erstellung schlecht durchdachter Designkonzepte, sondern ein Ratschlag für einen sorgfältigen Umgang mit Konzepten.

Der Nutzen und die Grenzen von Konzepten bestimmen nicht die Genauigkeit und den Detaillierungsgrad, der für eine bestimmte Situation angemessen ist. Die Wahl der richtigen Genauigkeit und des richtigen Detaillierungsgrades bei Konzeptarbeit erfordert Erfahrung.

2.2.2 Pragmatische Dokumentvorlagen für die verschiedenen Abstraktionsebenen

In diesem Abschnitt stellen wir Vorlagen für die unterschiedlichen Designkonzepte vor, die in Abschnitt 2.1 vorgestellt wurden. Die Definition und Auswahl dieser Vorlagen basiert auf den praktischen Erfahrungen der Lehrplanautorinnen und -autoren.

In der Literatur gibt es zahlreiche alternative Ansätze für die konzeptuelle Arbeit, und erfahrene Fachleute kennen und können sicherlich andere Techniken anwenden und andere Vorlagenstrukturen definieren. Bei der Definition der hier beschriebenen Vorlagen wurde der Schwerpunkt auf leichte Erlernbarkeit und schnelle Anwendbarkeit gelegt. In diesem Sinne sind die folgenden Vorlagen als pragmatisch zu verstehen. Sie sind vor allem für Einsteiger in der Konzeptarbeit ein guter Ausgangspunkt.

2.2.2.1 Digital Design Brief

Tabelle 3 zeigt eine pragmatische Vorlage für einen Digital Design Brief. Der Digital Design Brief wird verwendet, um die Gesamtidee und den Umfang des Bauprozesses für die digitale Lösung im Scoping-Schritt mit der Auftraggeber-Rolle zu klären (siehe Abschnitt 2.1).

Das bedeutet, dass der Digital Design Brief nicht nur erste Informationen über die beabsichtigte digitale Lösung liefert, sondern auch Informationen über den Prozess, der die digitale Lösung bauen soll. Um beide Aspekte des Digital Design Briefs zu unterstützen, geht die Vorlage in den ersten beiden Kapiteln (Kontext und Vision) auf die beabsichtigte digitale Lösung ein. In Kapitel 3 wird der mögliche Umfang für den Bau der gesamten digitalen Lösung definiert und in Kapitel 4 werden die allgemeinen Begriffe für den Bauprozess festgelegt. In der Tabelle sind mögliche Technologien aufgeführt, die die strukturierte Dokumentation des Inhalts eines bestimmten Kapitels unterstützen.

In der Spalte *Dokumentationstechnik* der Tabelle sind einige Ratschläge für die Erstellung der einzelnen Abschnitte des Digital Design Briefs aufgeführt. Ein ausführliches Beispiel für

Dokumentationstechniken findet sich in der YPRC-Fallstudie. Einzelheiten zur Liste der Stakeholder und zur Future Press Release sind in Abschnitt 2.2.3 zu finden.

Tabelle 3 - Dokumentvorlage für einen Digital Design Brief

Kapitel	Inhalt	Dokumentations- technik
1 Kontext		
1.1 Grund für das Vorhaben	Was ist der Grund/die Begründung/die Motivation der Auftraggeber-Rolle für die Entwicklung einer neuen digitalen Lösung?	Textuelle Beschreibung
1.2 Potenzielle Kunden	Liste der Personen (einschließlich einer kurzen Beschreibung der einzelnen Personentypen), die potenzielle Kunden der digitalen Lösung werden könnten	Aufzählung von Kundentypen mit textueller Erläuterung
1.3 Potenzielle Benutzer	Liste der Personen (einschließlich einer kurzen Beschreibung der einzelnen Personentypen), die potenzielle Benutzer der digitalen Lösung sind	Aufzählung von Benutzertypen mit textueller Erläuterung
1.4 Potenzielle weitere Stakeholder	Eine Liste von weiteren Personen, die für den Bauprozess oder die digitale Lösung als relevant angesehen werden	Stakeholderliste
1.5 Verwandte Lösungen	Beschreibungen von Lösungen, die als vergleichbar mit der geplanten Lösung zu verstehen sind	Textuelle Beschreibung für jede Lösung
1.6 Potenzielle Wettbewerber	Beschreibungen möglicher bestehender Lösungen, die als Konkurrenten angesehen werden können	Textuelle Beschreibung für jeden Mitbewerber
2 Vision	Eine Vision der angestrebten Zukunft, die durch die digitale Lösung geschaffen werden soll. Bei Bedarf können auch alternative Ideen für die digitale Lösung beschrieben werden.	Pressemitteilung aus der Zukunft (Future Press Release)
3 Umfang		
3.1 Lösungsraum	Charakterisierung des Lösungsraums, der erforscht werden soll, um die Vision zu verwirklichen	Textuelle Beschreibung
3.2 Mögliche Technologien	Beschreibungen von Technologien, die für die Realisierung der Vision als geeignet angesehen werden	Textuelle Beschreibung für jede Technologie
3.3 Randbedingungen	Beschreibung der Randbedingungen, die die digitale Lösung erfüllen muss	Textuelle Beschreibung für jede Randbedingung
3.4 No-Gos	Beschreibung von Technologien, Merkmalen und anderen Aspekten, die als inakzeptabel gelten	Textuelle Beschreibung für jedes No-Go
4 Allgemeine Bedingungen		

Kapitel	Inhalt	Dokumentations- technik
4.1 Zeitplan	Ein grober Zeitplan für die Entwicklung der digitalen Lösung, unterteilt in Zeitpläne für Konzeptarbeiten, mögliche Prototypen und einen Zeitplan für die Entwicklung einer ersten Version der digitalen Lösung	Zeitplan mit Zeiträumen
4.2 Modus der Zusammenarbeit	Beschreibung der Art und Weise der Zusammenarbeit zwischen dem Umsetzungsteam und den betroffenen Stakeholdern. Dazu gehören z. B. regelmäßige Koordinierungssitzungen, aber auch Rechte und Pflichten des Umsetzungsteams und der relevanten Stakeholder.	Textuelle Beschreibung
4.3 Budget	Budget für den Bau der Lösung, aufgeteilt in die Schritte Konzeptarbeit, Entwicklung und Betrieb	Tabelle mit verschiedenen Budgetposten
4.4 Einnahmequellen	Kurze Beschreibung der potenziellen Einnahmequellen für die digitale Lösung, falls zutreffend	Textuelle Beschreibung
4.5 Ressourcen	Erforderliche oder verfügbare Ressourcen (Personal, Arbeitsplätze, Fabriken usw.) für die Konzeptions- und Entwicklungsschritte	Tabelle mit verschiedenen Ressourcen

2.2.2.2 Lösungsdesignkonzept

Das Lösungsdesignkonzept beschreibt die digitale Lösung aus der Sicht der Auftraggeber-Rolle (siehe Kapitel 1 für den Unterschied zwischen Lösungsebene und Systemebene). Tabelle 4 zeigt eine pragmatische Dokumentenvorlage für ein Lösungsdesignkonzept. In Kapitel 1 der Dokumentenvorlage wird die digitale Lösung vorgestellt, um den Handlungsbedarf und die Vision der digitalen Lösung zu erläutern. Kapitel 2 der Dokumentenvorlage konzentriert sich auf den Kontext der digitalen Lösung und beschreibt die beabsichtigten Kundensegmente, Benutzergruppen und weitere relevante Stakeholder der digitalen Lösung.

Im Gegensatz zum Digital Design Brief, der Kunden, Benutzer und Stakeholder in den Mittelpunkt stellt, konzentriert sich das Lösungsdesignkonzept ganz klar auf die Lösung. Im Design Brief werden potenzielle Kundentypen, Benutzertypen und Stakeholder beschrieben, um mögliche Lösungsalternativen zu identifizieren. Im Lösungsdesignkonzept wird eine bestimmte Lösung definiert, einschließlich der relevanten Kundensegmente, Benutzergruppen und Stakeholder.

Beispiel YPRC. In der YPRC-Fallstudie könnten Hausärzte als potenzielle Benutzer in Frage kommen. Sie könnten sich im Rahmen einer medizinischen Untersuchung Zugang zu den Gesundheitsdaten des Läufers verschaffen. Dieser Dienst führt zu weiteren potenziellen Stakeholdern (z. B. Personen aus dem Bereich der Rechtsvorschriften für Gesundheitsdaten). Wird diese Funktion in der Konzeptphase nicht berücksichtigt, wird diese Benutzergruppe, einschließlich der entsprechenden Stakeholder, nicht Teil des Lösungsdesignkonzepts.

Bei der Definition von Kundensegmenten und Benutzergruppen im Lösungsdesignkonzept ist es wichtig, den Unterschied zwischen Kunden und Benutzern in Bezug auf die Stakeholder-Rollen zu beachten (siehe Abschnitt 1.2.3). In den Kapiteln 3 und 4 der Dokumentenvorlage geht es um den konkreten Wert und das Kundenerlebnis, das eine digitale Lösung bieten soll, sowie um das Geschäftsmodell, das die digitale Lösung vorantreiben soll.

Tabelle 4 - Dokumentvorlage für ein Lösungsdesignkonzept

Kapitel	Inhalt	Dokumentations- technik
1 Motivation	Motivation für die digitale Lösung, einschließlich der Gründe für das Handeln, eine Einführung in die digitale Lösung und die Vision für die digitale Lösung	Textuelle Beschreibung und Pressemitteilung aus der Zukunft
2 Kontext		
2.1 Kundensegmente	Beschreibung der Personen, die Kunden der digitalen Lösung sind	Persona-Vorlagen
2.2 Benutzergruppen	Beschreibung der Personen, die Benutzer der digitalen Lösung sind	Persona-Vorlagen
2.3 Weitere Stakeholder	Liste der weiteren Stakeholder, die für die digitale Lösung relevant sind	Stakeholderliste
3 Wertversprechen (Value Proposition)	Eine Beschreibung des Wertversprechens der digitalen Lösung	Value Proposition Canvas
4 Kundenerlebnis (Customer Experience)	Eine Beschreibung des Kundenerlebnisses, das die digitale Lösung den Kunden bieten soll	Customer Journey Map
5 Geschäftsmodell	Das Geschäftsmodell, das die digitale Lösung tragen wird	Business Model Canvas
6 Randbedingungen	Beschreibung der Randbedingungen, die für das Design der Lösung relevant sind	Liste der Randbedingungen

Wir empfehlen die Verwendung von Canvas-Technologien wie Value Proposition Canvas, Customer Journey Maps und Business Model Canvas, um das Wertversprechen, das Kundenerlebnis und das Geschäftsmodell zu dokumentieren. Im Falle eines komplexen Geschäftsmodells für digitale Lösungen empfehlen wir, die Literatur aus dem Bereich der Unternehmensanalyse zu Rate zu ziehen (vgl. z. B. [PaYC2010]), um weitere Hinweise zur Dokumentation von Geschäftsmodellen zu erhalten.

Es gibt Arten von digitalen Lösungen, die kein eigenes Geschäftsmodell haben. Typische Beispiele sind unternehmensinterne Systeme wie Enterprise Resource Planning-Systeme oder Customer Relationship Management-Systeme. Diese digitalen Lösungen sind Teil einer größeren Wertschöpfungskette und daher nur ein Teil eines umfassenden Geschäftsmodells. In einer solchen Situation empfehlen wir, das Geschäftsmodell zu dokumentieren, das Teil der digitalen Lösung ist. Wir empfehlen außerdem, den Wert zu erfassen, den die jeweilige digitale Lösung für das gesamte Geschäftsmodell darstellt.

2.2.2.3 Systemdesignkonzept

Das Systemdesignkonzept beschreibt das digitale System, das die digitale Lösung realisiert (siehe Abschnitt 1.2.1 für weitere Einzelheiten über den Unterschied zwischen Lösung und System). Tabelle 5 zeigt eine pragmatische Dokumentvorlage für ein Systemdesignkonzept. Die Vorlage spiegelt die in Abschnitt 1.2.1 vorgestellte Struktur von Form, Funktion und Qualität wider.

Kapitel 1 der Dokumentvorlage bietet eine Einführung für Leser, die mit der digitalen Lösung noch nicht vertraut sind. In den Kapiteln 2 und 3 der Dokumentvorlage werden die Ziele und Randbedingungen beschrieben, die für das Design des Systems relevant sind. Die explizite

Dokumentation von Zielen und Randbedingungen ist wichtig, um Gründe für das Design des Gesamtsystems zu liefern.

Das Hauptaugenmerk des Systemdesignkonzepts liegt auf der Form des digitalen Systems (Kapitel 4 der Dokumentvorlage). Die Dokumentvorlage listet die Benutzertypen (als Teil des Systemkontextes), bestehende Objekte/Systeme und die zu realisierenden Elemente auf. Bei diesem Ansatz gibt das Systemdesignkonzept einen Überblick über die verschiedenen Elemente des digitalen Systems einschließlich ihrer Beziehungen.

Tabelle 5 - Dokumentvorlage für ein Systemdesignkonzept

Kapitel	Inhalt	Dokumentations-technik
1 Einleitung	Beschreibung der allgemeinen Systemidee als Ausgangspunkt für Leser, die mit der digitalen Lösung nicht vertraut sind	Textuelle Beschreibung
2 Ziele	Beschreibung der Ziele, die für das Design des Systems relevant sind	Zielvorlagen
3 Randbedingungen	Beschreibung der Randbedingungen, die für das Design des Systems relevant sind	Randbedingungs-vorlagen
4 Form	Überblick über das digitale System und seinen Kontext als Einführung	Übersichtsbild
4.1 Benutzertypen	Benutzertypen, die das digitale System als Teil des Kontexts nutzen werden	Benutzertyp-Vorlagen
4.2 Vorhandene Objekte	Objekte im Kontext, deren tatsächliche Präsenz angenommen wird	Objektvorlagen
4.3 Vorhandene Systeme	Systeme, die als Elemente Teil des Kontextes sind und tatsächliche Präsenz angenommen wird	Systemvorlagen
4.4 Zu realisierende Elemente	Elemente, die realisiert werden müssen	Softwareelement-/Geräte-Vorlage
5 Funktion	Exemplarische Beschreibung der Funktionen, die das System bietet	Szenario-Vorlagen
6 Qualitätsanforderungen	Beschreibung der Qualitätsmerkmale des Systems	Vorlagen für Qualitäts-anforderungen

Im Vergleich zum Kapitel über die Form wird das Kapitel über die Funktion des digitalen Systems eher kurz und exemplarisch beschrieben. Der Hauptgrund dafür ist, dass das Abstraktionsniveau des Systemdesignkonzepts hoch genug ist, um als Konzept zu dienen, das sich auf die Form des gesamten Systems konzentriert. Daher ist das Kapitel über die Form in die verschiedenen Arten von Elementen unterteilt.

Beispiel YPRC. Beispiele aus der YPRC-Fallstudie sind in Klammern angegeben:

- Benutzertypen (Läufer und Lauf-Coach)
- Vorhandene Objekte (das Smartphone des Läufers und der PC des Lauf-Coach)
- Bestehende Systeme (der Kartenmaterial- und der Zahlungsanbieter-System)
- Zu realisierende Elemente (die Smartwatch des Läufers, die App des Läufers und das Portal)

Die Hauptmotivation für die Wahl dieses Detaillierungsgrades für das Systemdesignkonzept ergibt sich aus der Idee der Systemebene als Managementinstrument für den gesamten Bauprozess. Nur wenn Sie die Form des digitalen Systems genau kennen, können Sie die

nachfolgenden Arbeiten entsprechend strukturieren (siehe Abschnitt 1.3.4). Die Einzelheiten der verschiedenen Funktionen des Systems werden auf Elementebene definiert, wo alle für eine angemessene Beschreibung erforderlichen Informationen verfügbar sind.

Oft unterschätzte Faktoren auf der Systemebene sind die Namen der verschiedenen Elemente. Wir empfehlen Ihnen, die Namen der verschiedenen Elemente mit Bedacht zu wählen, da diese Namen während des Bauprozesses für die Kommunikation über die verschiedenen Elemente verwendet werden. Die Namen sollten aussagekräftig sein und die Funktion des Elements widerspiegeln. Zweideutige Namen oder sehr ähnliche Namen können leicht zu Missverständnissen führen. In der YPRC-Fallstudie verwendet der Läufer beispielsweise *die Runner's App*. Das Element, das der Coach verwendet, heißt *Portal* und nicht *Coach's App*. Der Name „*Coach's App*“ liegt nahe an „*Runner's App*“ und man könnte nur über die *App* sprechen, was definitiv zu Missverständnissen führen würde.

2.2.2.4 Softwaredesignkonzept

Mit dem Softwaredesignkonzept gehen wir auf die Details eines bestimmten Softwareelements des digitalen Systems ein. Tabelle 6 zeigt eine pragmatische Dokumentvorlage für ein Softwaredesignkonzept. Der Name des auf der Systemebene beschriebenen Softwareelements muss als Titel des Konzepts verwendet werden. Das Softwaredesignkonzept in der YPRC-Fallstudie verwendet zum Beispiel *Runner's App* als Titel.

Tabelle 6 - Dokumentvorlage für ein Softwaredesignkonzept

Kapitel	Inhalt	Dokumentations-technik
1 Einleitung	Beschreibung der Idee für das Element als Ausgangspunkt für die Leser	Textuelle Beschreibung
2 Ziele	Beschreibung der Ziele, die für das Design des Softwareelements relevant sind	Zielvorlage
3 Randbedingungen	Beschreibung der Randbedingungen, die für das Design des Softwareelements relevant sind	Randbedingungs-vorlage
4 Form	Grafischer Überblick über den relevanten Kontext des Softwareelements	Übersichtsbild
4.1 Hardware-Schnittstellen	Wahrnehmbare und zugrundeliegende Hardware-Schnittstellen, die das Softwareelement im Kontext erwartet	Wahrnehmbare und zugrundeliegende Hardware-Schnittstellenvorlage
4.2 User Interface	Beschreibung der Schnittstellen des Softwareelements für den Benutzer	User Interface-Vorlage
4.3 Software-Schnittstelle	Beschreibung der Schnittstellen zu verwandten Softwaresystemen, d. h. API (Application Programming Interfaces)	Software-Schnittstellenvorlage
4.4 Entitäten	Beschreibung der Daten, die das Softwareelement speichert	Entitätsvorlage
5 Funktion		

Kapitel	Inhalt	Dokumentations- technik
5.1 Use Cases	Beschreibung der Use Cases, die das Softwareelement unterstützt	Use Case Vorlage
5.2 Funktionen	Beschreibung der zugrundeliegenden Funktionen, die das Softwareelement bereitstellt	Funktionsvorlage
6 Qualitätsanforderungen	Beschreibung der Qualitätsanforderungen, die das Softwareelement erfüllen muss	Qualitätsanforderung svorlage

Wie beim Systemdesignkonzept dient auch die Einführung (Kapitel 1 der Dokumentvorlage) des Softwaredesignkonzepts als Überblick über die digitale Lösung und soll ein bestimmtes Konzept ohne zusätzliche Informationen lesbar machen.

In den Kapiteln 2 und 3 der Dokumentvorlage werden die Ziele und Randbedingungen beschrieben, die für das Design des Softwareelements relevant sind. Wie beim Systemdesignkonzept ist auch die explizite Dokumentation von Zielen und Randbedingungen wichtig, um Gründe für das Design des Softwareelements zu liefern. Die auf der Systemebene und der Elementebene dokumentierten Ziele und Randbedingungen sind in der Regel miteinander verknüpft oder verfeinern sich gegenseitig.

Kapitel 4 der Dokumentvorlage befasst sich mit der Form des Softwareelements, einschließlich seines relevanten Kontexts. Es bietet einen Überblick über das gesamte digitale System aus der Perspektive des Softwareelements, einschließlich einer Beschreibung der relevanten Umgebung aus dem Systemdesignkonzept (Benutzer, vorhandene Objekte, vorhandene Systeme). Darüber hinaus beschreibt das Kapitel die wahrnehmbaren und zugrundeliegenden Hardware-Schnittstellen, die das Softwareelement im Kontext erwartet.

Die Beschreibung dieser Hardware-Schnittstellen ist wichtig, um die technische Umgebung zu erfassen, in der ein Softwareelement arbeiten wird. Typische Hardwareschnittstellen umfassen Displays, Audioeingang und -ausgang sowie Kommunikationshardware (z. B. eine Internetverbindung). Die Form des Elements selbst wird durch die User Interfaces für die Interaktion mit dem Benutzer, die Software-Schnittstellen für die Interaktion mit anderen Softwaresystemen und die Entitäten zur Beschreibung der von dem Softwareelement gespeicherten Informationen beschrieben.

Kapitel 5 der Dokumentvorlage befasst sich mit der Funktion des Softwareelements. Use Cases beschreiben die wahrnehmbare Funktion anhand von Interaktionsschritten zwischen dem Benutzer und dem Softwareelement. Die zugrundeliegende Funktion wird mit Hilfe spezieller Funktionsvorlagen beschrieben. Jede zugrundeliegende Funktion stellt eine relevante Transformation von Daten dar.

Im Gegensatz zu den Szenarien auf Systemebene müssen die Use Cases und Funktionen auf Elementebene eine vollständige Beschreibung der Funktion des Softwareelements auf einer bestimmten Abstraktionsebene liefern. Aus Konsistenzgründen dürfen die Use Cases und Funktionen auf der Elementebene nicht im Widerspruch zur Szenariobeschreibung auf der Systemebene stehen. Weitere Einzelheiten hierzu finden Sie in Abschnitt 2.2.5.

Kapitel 6 schließt das Softwaredesignkonzept mit den Qualitätsanforderungen ab, die das Softwareelement erfüllen muss.

2.2.2.5 Gerätedesignkonzept

Neben Softwareelementen kann ein digitales System auch aus Geräten bestehen, die speziell für das digitale System entwickelt und hergestellt werden. Ein Gerät besteht aus Hardware und Software. .

Tabelle 7 zeigt eine pragmatische Vorlage für ein Gerätedesignkonzept. Der Name des auf der Systemebene beschriebenen Geräts muss als Titel des Konzepts verwendet werden. Das Gerätedesignkonzept im YPRC-Fall trägt beispielsweise den Titel *Runner's Watch*. Die Struktur ähnelt der Dokumentvorlage des Softwaredesignkonzepts (siehe Tabelle 6). Wir diskutieren daher nur die Unterschiede zwischen den beiden Vorlagen.

Der Hauptunterschied zwischen einem Softwareelement und einem Gerät besteht darin, dass das Gerät Hardware (z. B. CPU, Arbeitsspeicher und Speicher) und Software kombiniert. Das bedeutet, dass das Gerät über die gesamte Hardware und Software verfügen muss, die zur Erfüllung seines Zwecks erforderlich ist.

Dieser Unterschied spiegelt sich in Kapitel 3 und Kapitel 4 der Dokumentvorlage wider. Kapitel 4 der Dokumentvorlage bietet einen Überblick über den Gerätekontext, eine Beschreibung seiner physischen Teile, seiner Schnittstellen und seiner gespeicherten Entitäten. Zusätzlich zur Hardware-Schnittstelle enthält Kapitel 4 einen Abschnitt über die physischen Teile des Geräts.

Tabelle 7 - Dokumentvorlage für ein Gerätedesignkonzept

Kapitel	Inhalt	Dokumentations- technik
1 Einleitung	Beschreibung der allgemeinen Geräteidee als Ausgangspunkt für Leser, die mit der digitalen Lösung nicht vertraut sind	Textuelle Beschreibung
2 Ziele	Beschreibung der Ziele, die für die Gestaltung des Geräts relevant sind	Zielvorlage
3 Randbedingungen	Beschreibung der Randbedingungen, die für das Design des Geräts relevant sind	Randbedingungs- vorlage
4 Form	Grafischer Überblick über den relevanten Kontext des Geräts	Übersichtsbild
4.1 Physische Bauteile	Beschreibung der physischen Teile, aus denen das Gerät besteht	Bauteilvorlage
4.2 Hardware-Schnittstellen	Beschreibung der wahrnehmbaren und zugrundeliegenden Hardware-Schnittstellen des Geräts	Vorlage für wahrnehmbare/zugrundeliegende für Hardware-Schnittstellen
4.3 Software-Schnittstelle	Beschreibung der (zugrundeliegenden) Schnittstellen zu anderen Softwaresystemen	Software-Schnittstellen- vorlage
4.4 User Interface	Beschreibung der (wahrnehmbaren) Schnittstellen zu den Benutzern des Softwareelements	User Interface- Vorlage
4.5 Entitäten	Beschreibung der Daten, die das Gerät speichert	Entitätsvorlage

Kapitel	Inhalt	Dokumentations- technik
5 Funktion		
5.1 Use Cases	Beschreibung der Use Cases, die das Gerät unterstützt	Use Case Vorlage
5.2 Funktionen	Beschreibung der zugrundeliegenden Funktionen, die das Gerät bietet	Funktionsvorlage
6 Qualitätsanforderungen	Beschreibung der Qualitätsanforderungen, die das Produkt erfüllen muss	Qualitätsanforderungsvorlage

Es ist wichtig zu wissen, dass die Gerätedesignvorlage allein nicht ausreicht, um ein vollwertiges Gerät zu bauen. Es muss durch spezielle Konzepte ergänzt werden, die sich mit dem Design, der Konstruktion und der Realisierung der Hardwareteile des Geräts befassen.

Für das Design und den Bau eines Geräts sind zusätzliche Fachleute (z. B. aus dem Industriedesign, der Produktentwicklung usw.) und Konzepte (z. B. technische Zeichnungen, Schaltpläne, Produktpläne) erforderlich. Das vorgestellte Gerätedesignkonzept ist für die Perspektive des Digital Designs und für die Gestaltung des digitalen Teils des Geräts erforderlich.

2.2.3 Dokumentationstechniken für die Lösungsebene

Im Folgenden stellen wir eine Reihe von Techniken vor, die für die Arbeit mit Designkonzepten auf der Lösungsebene nützlich sind (siehe Abschnitt 1.3.4). Table 8 gibt einen Überblick über die Techniken und ihre Anwendbarkeit in den verschiedenen Designkonzepten.

Table 8 – Überblick über Dokumentationstechniken und Arbeitsprodukte für die Lösungsebene

<i>Dokumentationstechnik</i>	<i>Digital Design Brief (Bezugnahme auf Tabelle 3)</i>	<i>Lösungsdesignkonzept (Bezugnahme auf Tabelle 4)</i>
<i>Pressemitteilung aus der Zukunft (Future Press Release)</i>	Dokumentation der Vision (Kapitel 2)	Dokumentation der Vision (Kapitel 1)
<i>Persona</i>	-	Dokumentation von Kundentypen (Kapitel 2.1) und Benutzergruppen (Kapitel 2.2)
<i>Stakeholderliste</i>	Dokumentation potenzieller Stakeholder (Kapitel 1.3)	Dokumentation der relevanten Stakeholder (Kapitel 2.3)
<i>Value Proposition Canvas</i>	-	Dokumentation des Wertversprechens (Kapitel 3)
<i>Customer Journey Map</i>	-	Dokumentation des Kundenerlebnisses (Kapitel 4)
<i>Business Model Canvas</i>	-	Dokumentation des Geschäftsmodells (Kapitel 5)

2.2.3.1 Pressemitteilung aus der Zukunft (Future Press Release)

Die „Pressemitteilung der Zukunft“ [Ross2019] ist eine Technik zur Beschreibung der Vision für eine digitale Lösung. Sie nimmt den Leser mit in die Zukunft und beschreibt den zentralen Erfolg, den die geplante digitale Lösung erreicht haben wird. Um eine Pressemitteilung aus der Zukunft zu erstellen, definiert [Ross2019] die folgenden Regeln:

Regel 1: Die Pressemitteilung aus der Zukunft muss zu einem zukünftigen Zeitpunkt erfolgen, zu dem der Erfolg erreicht und realisiert wurde. Sie beschreibt damit den zentralen Zweck der digitalen Lösung.

Regel 2: Beginnen Sie mit dem Kunden. Aus der Pressemitteilung muss hervorgehen, warum die digitale Lösung für die Kunden wichtig ist und wie ihre Erfahrungen verbessert wurden. Diese Regel konzentriert sich auf den Kontext der digitalen Lösung. Eine kurze Geschichte eines Kunden, einschließlich eines Zitats, ist ein gutes Instrument für eine kurze Beschreibung dieses Aspekts.

Regel 3: Setzen Sie sich ein ehrgeiziges und klares Ziel. Die in der Pressemitteilung beschriebenen erreichten Ziele sollten messbar sein und können auch finanzielle Ergebnisse oder Marktanteile beinhalten. Ein Zitat eines Managers ist ein gutes Mittel, um diesen Aspekt kurz zu beschreiben.

Regel 4: Skizzieren Sie die Grundsätze, die zum Erfolg geführt haben. Diese Regel konzentriert sich auf die wesentlichen Details der Lösung und kann auch wichtige Randbedingungen für die Lösung enthalten. Eine kurze Merkmalsliste ist ein gutes Hilfsmittel für eine kurze Beschreibung dieses Aspekts.

Alles in allem sollte sich die Pressemitteilung aus der Zukunft auf die künftigen funktionalen Nutzenbringer oder Problemlöser konzentrieren, aber auch auf das künftige emotionale Erlebnis aus der Sicht der Auftraggeber und Benutzer. Die Pressemitteilung aus der Zukunft sollte auf eine einzige Seite passen. Ein Foto, das wichtige Aspekte der Lösung zeigt, sollte ebenfalls beigefügt werden. Abbildung 14 zeigt einen Screenshot einer Pressemitteilung aus der Zukunft für die YPRC-Fallstudie. Der vollständige Text dieser Pressemitteilung aus der Zukunft ist in der YPRC-Fallstudie zu finden.

2.2.3.2 Persona zur Charakterisierung von Kunden- und Benutzergruppen

Personas sind eine gängige, auf der Benutzer-/Kundenforschung basierende Methode zur Beschreibung von Kunden- und Benutzergruppen [Coop2004]. Sie werden häufig verwendet, um ethnografische Forschung zu synthetisieren und Empathie aufzubauen:

Persona: A fictitious character representing a group of people with similar needs, values, and habits who are expected to use a system or benefit from it in a similar way.

Personas sind ein psychologisches Designhilfsmittel, das es Designern ermöglicht, die Gewohnheiten und Werte der potenziellen Benutzer zu verstehen und Empathie für sie zu



Abbildung 14 - Eine Pressemitteilung aus der Zukunft

entwickeln. Sie sind ein wichtiges Werkzeug im Designprozess, um die Ziele des Kunden oder Benutzers zu entwickeln und ein entscheidendes Instrument zur Ideenfindung und Validierung von Designkonzepten. Gute Personas helfen dabei, die Erfahrungen, das Verhalten und die Motivation des Kunden oder Benutzers zu erkennen. Insbesondere auf der Lösungsebene helfen Personas dabei, herauszufinden, welche Kunden oder Benutzer im Designprozess am wichtigsten sind.

In einem ersten Schritt können Personas auf der Grundlage von Hypothesen erstellt werden. Gute Personas sollten diese Hypothesen jedoch verifizieren und sich stets auf valide empirische Forschung stützen.

Personas können im Designprozess hinsichtlich der folgenden Punkte helfen [CRCN2014]:

1. Aufbau eines gemeinsamen Verständnisses für einen Kunden oder Benutzer
2. Bestimmung der Funktionen, die eine erfolgreiche digitale Lösung erfüllen muss
3. Ankerpunkt für Designentscheidungen und gemeinsame Diskussionen mit Stakeholdern
4. Unterstützung bei der Evaluierung von Designkonzepten

Personas sind daher auch ein wichtiges Instrument zur Vermeidung von selbstreferenziellem Design, bei dem Designer ihre eigenen Bedürfnisse, Ziele, Fähigkeiten und mentalen Modelle in die Lösung projizieren.

Im Gegensatz zu Benutzerprofilen, die oft stereotyp sind, sollten Personas auf ethnografischen Daten aus erster Hand beruhen. Im Gegensatz zu Marktsegmenten, die sich stark auf soziodemografische Aspekte konzentrieren, zielen Personas auf Nutzungsverhalten und Ziele ab.

Wesentliche Kriterien für gute Personas sind:

- Wirkt die Persona wie eine echte Person und nicht wie eine Karikatur oder ein Stereotyp?
- Ist die narrative Kraft der Persona überzeugend?
- Hebt die Persona die Kerneigenschaften und übergeordneten Ziele des Benutzers hervor?
- Ist die Persona genug fokussiert, um Konzeptentscheidungen zu ermöglichen?
- Ist die Persona brauchbar?
- Ist die Persona in ihrer formalen Qualität überzeugend?

Damit Personas effektiv sein können, müssen einige Stolpersteine vermieden werden [Fla2018]. Ein solcher Stolperstein ist, dass alle Stakeholder verstehen sollten, was Personas sind und wofür sie verwendet werden. Personas sollten auch nicht in getrennten Bereichen erstellt und einem Projekt einfach aufgezwungen werden. Wichtig ist auch, dass die Führungsebene die Personas inhaltlich vollständig mitträgt. Um sicherzustellen, dass die Personas schließlich auch genutzt werden, ist es wichtig, sie an prominenter Stelle zu platzieren (z. B. durch Aushang im Büro).

Eine Persona sollte aus einem Bild, grundlegenden Hintergrundinformationen und Informationen über Ziele und Verhalten bestehen. Darüber hinaus können auch Schwierigkeiten bei der aktuellen Situation oder Lösung angegeben werden.

Abbildung 15 zeigt eine Persona-Vorlage mit den folgenden Elementen:

- *Demografische Daten*: Name, Foto und weitere Angaben zur Persona
- *Verhalten*: Informationen über das Verhalten der Persona in Bezug auf Technologie, Gemeinschaften, Hobbys und die Lösung
- *Ziele*: Informationen über die Ziele der Persona in Bezug auf Leben, Arbeit und die Lösung

<p>Demografische Daten</p> <div style="border: 1px solid black; width: 100px; height: 80px; margin: 0 auto; text-align: center; line-height: 80px;">Foto</div> <p>Name:</p> <p>Alter:</p> <p>Angaben zur Familie:</p> <p>Berufliche Tätigkeit:</p>	<p>Verhalten</p> <p>Technologie:</p> <p>Communities:</p> <p>Hobbys:</p> <p>Beziehung zur Lösung:</p> <hr/> <p>Ziele</p> <p>Lebensziele:</p> <p>Arbeitsziele:</p> <p>Warum sie/er die Lösung verwendet (oder nicht verwendet):</p> <p>Was sie/er mit der Lösung erreichen will:</p>
---	--

Abbildung 15 - Eine Persona-Vorlage

2.2.3.3 Stakeholderliste

Eine Stakeholderliste ist eine strukturierte Tabelle zur Dokumentation von Stakeholdern. Sie besteht aus den folgenden Spalten [CPRE2020]:

- Name des Stakeholders
- Funktion (Rolle) des Stakeholders
- Persönliche Daten und Kontaktdaten, einschließlich der Verfügbarkeit
- Verantwortungsbereich und Umfang der Fachkenntnisse

Darüber hinaus kann es nützlich sein, nicht-öffentliche Informationen über die Stakeholder zu dokumentieren: Temperament, Verständnis und Engagement (siehe Abschnitt 4.3) sowie Ziele und Interessen in Bezug auf die digitale Lösung.

2.2.3.4 Value Proposition Canvas

Das Value Proposition Canvas ist eine Vorlage zur Beschreibung von Kundenprofilen in Bezug auf die angebotenen Werte [OPBS2014]. Die Persona-Vorlagen (siehe 2.2.3.2) werden verwendet, um jeden Kunden-/Benutzertyp im Allgemeinen zu beschreiben, während das Value Proposition Canvas sich auf den besonderen Wert der digitalen Lösung für Kunden und Benutzer konzentriert. Sie ermöglicht eine genauere Beschreibung der Wertversprechen und der Zielkundensegmente. Sie ermöglicht auch eine Bewertung der Übereinstimmung zwischen dem beabsichtigten Wert und den Erwartungen, die die Kunden haben sollten.

Ein Value Proposition Canvas (siehe Abbildung 16) besteht aus einem Kunden- /Benutzer-Profil und einer Value Map. Das Kunden-/Benutzer-Profil beschreibt:

- *Kundenaufgaben*: Was die Kunden bei der Arbeit oder in ihrem Leben zu erledigen versuchen
- *Nutzen*: Die Ergebnisse und/oder Vorteile, die die Kunden in Bezug auf ihre Arbeit erreichen wollen
- *Probleme*: Schlechte Ergebnisse, Risiken und Hindernisse im Zusammenhang mit den Kundenaufgaben

Bei der Verwendung von Persona-Vorlagen zur Beschreibung von Kundengruppen (siehe Abschnitt 2.2.3.2) sollte dieser Teil des Value Proposition Canvas mit den Persona-Beschreibungen übereinstimmen.

Die Value Map beschreibt:

- Produkte und Dienstleistungen, die den Kunden durch die digitale Lösung angeboten werden
- Nutzenbringer, die beschreiben, wie Produkte und Dienstleistungen Gewinne schaffen
- Problemlöser, die beschreiben, wie Produkte und Dienstleistungen Probleme abbauen

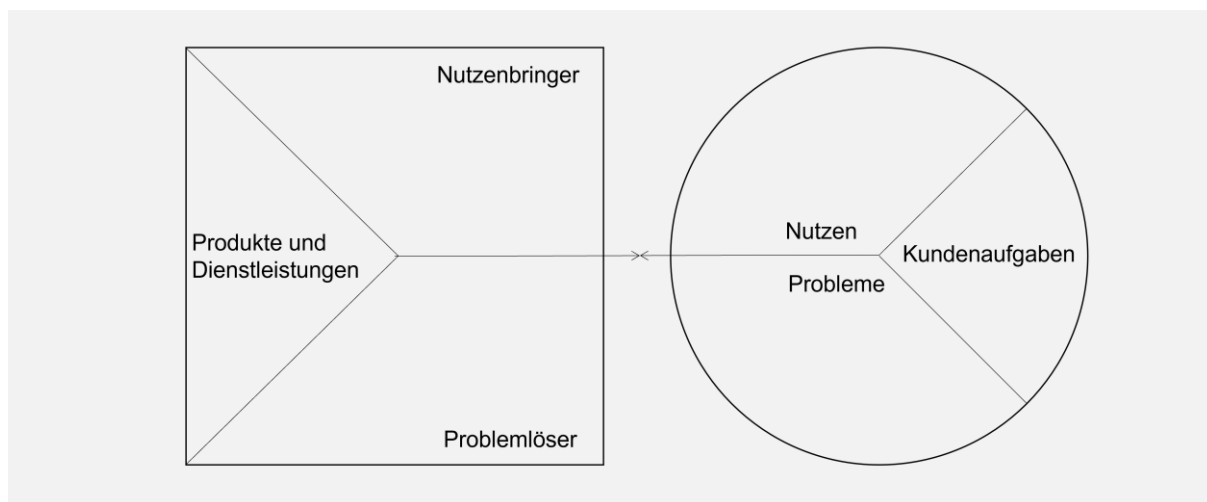


Abbildung 16 - Eine Vorlage für ein Wertversprechen (Value Proposition Canvas)

Die oben gezeigte Vorlage wird als physisches Canvas verwendet, d. h. das Canvas wird auf ein großes Poster gedruckt (oder auf ein Whiteboard gezeichnet) und mit Haftnotizen gefüllt, um den Inhalt in den verschiedenen Teilen des Canvas zu dokumentieren. Zu Dokumentationszwecken kann das Canvas mit einem Zeichenwerkzeug neu gezeichnet, über ein Foto dokumentiert oder mit einer Aufzählung für jeden Teil des Canvas dokumentiert werden.

In einem Lösungsdesignkonzept wird für jeden Kundentyp (wenn die digitale Lösung mehr als einen Kundentyp hat) ein Wertversprechen erstellt. Ein Beispiel und eine Anleitung für die Arbeit mit dem Value Proposition Canvas finden Sie in Abschnitt 5.2.

Das Value Proposition Canvas kann vor, während und nach der Entwicklung eines tiefgreifenden Verständnisses der Kunden eingesetzt werden. Wenn es vorher verwendet wird, zeigt es auf, was ein Umsetzungsteam über die Kunden lernen muss und was im Hinblick auf die Wertversprechen zu bewerten ist. Wenn das Team es im Nachhinein verwendet, hilft es dem Team, die Übereinstimmung zwischen dem beabsichtigten Wert und den Kundenerwartungen zu analysieren und zu bewerten.

Das Value Proposition Canvas kann auf neue und bestehende Wertversprechen und Kundensegmente gleichermaßen angewendet werden. In beiden Fällen hilft es, das Verständnis und das Denken über die digitale Lösung zu strukturieren und die Ideen greifbarer zu machen.

2.2.3.5 Customer Journey Map

Eine Customer Journey Map ist ein Werkzeug aus dem Service Design [PoLR2013] welches genutzt werden kann, um an Form, Funktion und Qualität digitaler Lösungen und damit am Kundenerlebnis zu arbeiten.

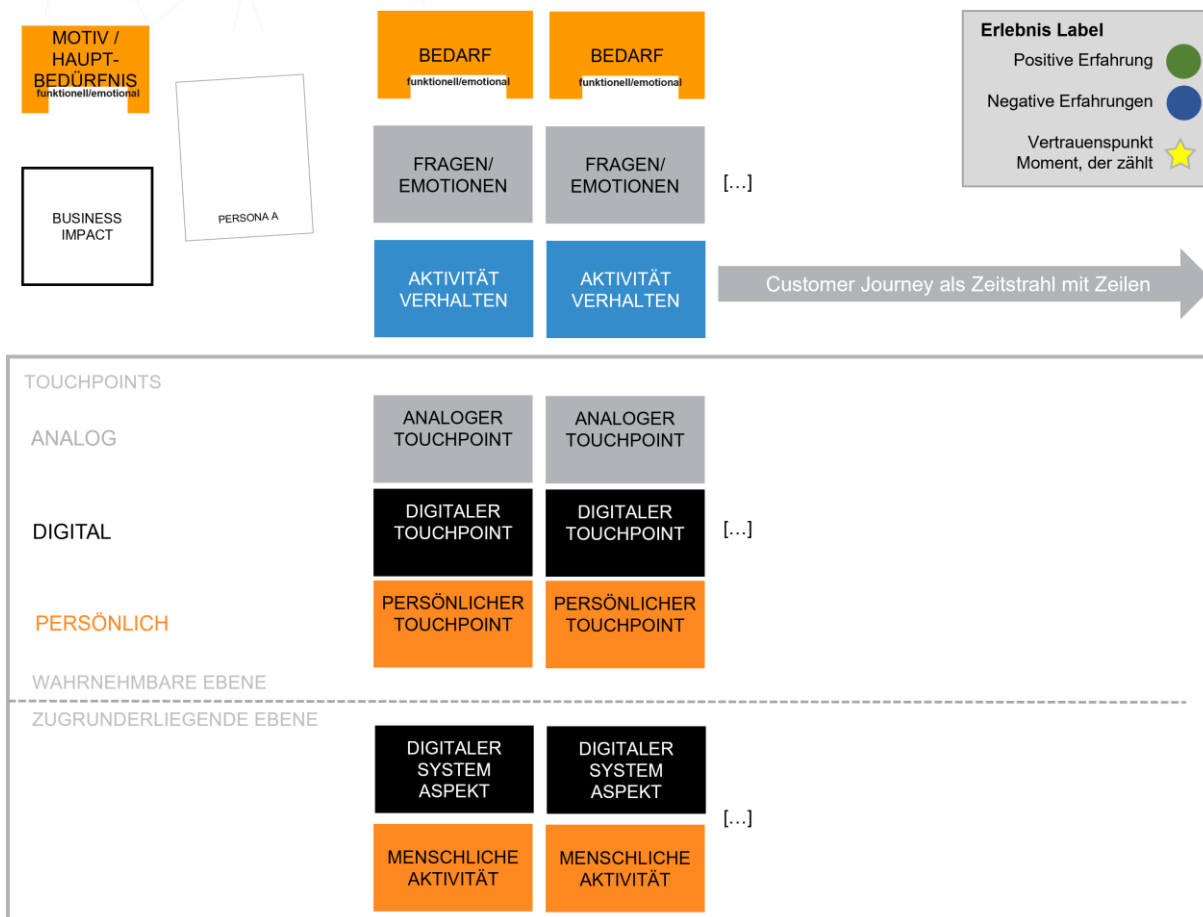


Abbildung 17 - Vorlage für eine Customer Journey Map Canvas

Abbildung 17 zeigt eine Vorlage für ein Customer Journey Map Canvas mit den folgenden Elementen:

- Das Motiv des Kunden als zentrales Kundenbedürfnis. Wir empfehlen, zwischen funktionalen Bedürfnissen (z. B. *Verbesserung der persönlichen Laufleistung*) und emotionalen Bedürfnissen (z. B. *Vermeidung von Gruppendruck*) zu unterscheiden.
- Ein Bild und der Name als Hinweis auf die Kunden-Persona, deren Reise dargestellt wird. Wir empfehlen die Erstellung von Customer Journey Maps für alle im Lösungsdesignkonzept definierten Kunden-Personas.
- Eine kurze Erklärung zu den positiven Auswirkungen der vorgestellten Reise auf das Unternehmen (z. B. *Umsatzsteigerung*).
- Eine Tabelle für den Zeitplan der Reise. Jede Spalte stellt einen Touchpoint mit einer detaillierten visuellen und/oder textuellen Beschreibung als eine Zeile dar. Eine Zeile besteht aus den folgenden Elementen:
 - Die Aktivität oder das Verhalten des Kunden, mit optionalen Bedürfnissen und Fragen/Emotionen des Kunden
 - Die wahrnehmbare Ebene der digitalen Lösung in Form von analogen, digitalen oder persönlichen Touchpoints
 - Die der digitalen Lösung zugrundeliegende Ebene in Bezug auf einen digitalen Systemaspekt oder einen menschlichen Aspekt

Neben der Zeitleiste sind Erfahrungskennzeichnungen nützlich, um die Qualität der Erfahrung für den Kunden in Form von positiven oder negativen Erfahrungen und besonderen

Vertrauenspunkten oder Momenten, die für den Kunden von besonderer Bedeutung sind, hervorzuheben.

2.2.3.6 Business Model Canvas

Das Verständnis der Werte und der Art der Beziehungen, die diese Werte schaffen, wird durch das Business Model Canvas unterstützt. Um tatsächlich eine nachhaltige digitale Lösung zu realisieren, ist die Geschäftsmodellperspektive als zusätzlicher Blickwinkel notwendig. Ein Business Model Canvas [OsPi2010] ist eine Vorlage, die ein Geschäftsmodell in kompakter Form beschreibt.

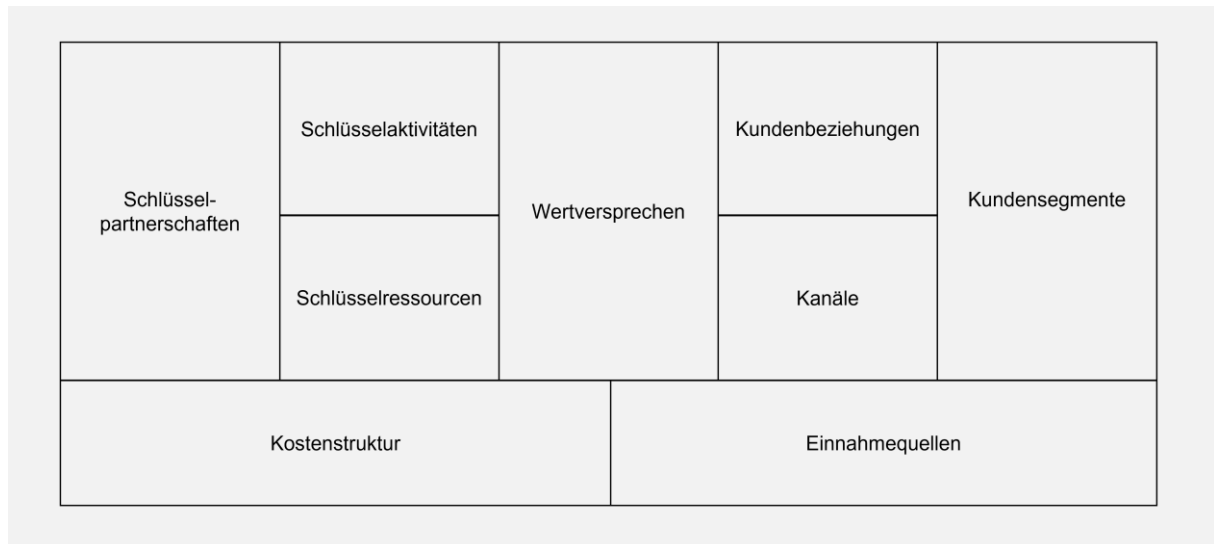


Abbildung 18 - Eine Vorlage für ein Business Model Canvas

Es besteht aus den folgenden Elementen (siehe Abbildung 18):

- **Schlüsselaktivitäten:** wichtige Aktivitäten zur Umsetzung des Wertversprechens eines Unternehmens
- **Schlüsselressourcen:** Ressourcen, die notwendig sind, um einen Mehrwert für den Kunden zu schaffen
- **Schlüsselpartnerschaften:** Beschreibung der Partner für den Aufbau des Geschäftsmodells
- **Wertversprechen:** Die Sammlung von Produkten und Dienstleistungen, die ein Unternehmen anbietet
- **Kundensegmente:** Beschreibung der Kundensegmente
- **Kanäle:** Vertriebs- und Lieferkanäle
- **Kundenbeziehungen:** Beabsichtigte Beziehung zu Kundensegmenten
- **Kostenstruktur:** Kosten, die für den Betrieb des Unternehmens notwendig sind
- **Einnahmequellen:** Einkünfte aus dem Geschäft

Das Business Model Canvas kombiniert zwei zentrale Perspektiven: die Außenperspektive (Kunde) und die Innenperspektive (Bauendes Unternehmen oder Auftraggeber-Organisation). Bei der Arbeit am Business Model Canvas wechselt ein Umsetzungsteam ganz natürlich zwischen beiden Perspektiven und kann so verschiedene Stakeholder-Perspektiven einbeziehen.

2.2.4 Allgemeine Überlegungen zu Designkonzepten auf System- und Elementebene

Bislang haben wir in Abschnitt 2.2.2 Dokumentvorlagen für die unterschiedlichen Abstraktionsebenen und in Abschnitt 2.2.3 Dokumentationstechniken für die Lösungsebene vorgestellt. Jetzt betreten wir die Welt der Systemebene und der Elementebene. Für Einsteiger ist dies ein wichtiger Schritt in Bezug auf den Schreibstil und die Arbeitsweise. Um diesen Schritt zu erleichtern, beginnen wir mit einer allgemeinen Diskussion über den Unterschied zwischen der Lösungsebene und der System-/Elementebene in Bezug auf den Schreibstil und die Arbeitsweise.

Die Konzepte auf der Lösungsebene (Digital Design Brief und das Lösungsdesignkonzept) setzen auf eine lineare Fließtextdokumentation und einen Canvas-orientierten Dokumentationsansatz. Für die Erstellung von Designkonzepten auf der System- und Elementebene ist ein anderer, mehr strukturierter Schreibstil aus folgenden Gründen besser geeignet:

- 1) Er muss den Detailgrad widerspiegeln, der für den Aufbau eines digitalen Systems und seiner Elemente erforderlich ist, und
- 2) Er muss die Integration in die anderen Aktivitäten des Bauprozesses unterstützen.

Der Schreibstil bei der Erstellung von System- oder Elementdesignkonzepten (kurz: SoE-Konzepte) unterscheidet sich grundlegend vom Schreibstil, den wir im Digital Design Brief, im Lösungsdesignkonzept, im Alltag (z. B. in Briefen) oder in der Schule bzw. an der Universität (z. B. bei Klausuren oder Diplomarbeiten) verwenden. Diese *Alltagssprache* unterliegt in der Regel der Prämisse, dass Kommunikation in linearer Form erfolgt, d. h. ein Brief oder eine Seminararbeit wird von Anfang bis zum Ende gelesen, um den Sinn des Textes oder der Mitteilung zu erfassen. Das bedeutet übrigens nicht, dass die Texte auch linear (d.h. von vorne nach hinten) geschrieben sind.

Ein SoE-Konzept unterliegt der gleichen linearen Struktur - das liegt in der Natur der Sprache - und wird auf zwei Ebenen definiert:

- Die *Überschriftenebene* definiert die Dokumentstruktur eines Konzepts und bietet eine klare Orientierung und einen einfachen Zugang zum Inhalt der einzelnen Abschnitte des Konzepts.
- Die *Bausteinebene* (siehe Abschnitt 2.2.5) wird verwendet, um einen bestimmten Aspekt des Systems (oder des Elements) zu beschreiben, der realisiert werden muss⁶.

Ein SoE-Konzept wird jedoch nicht mit dem Ziel erstellt, linear gelesen und verstanden zu werden; es muss vielmehr eine klare Struktur aufweisen. Ein gutes Vorbild für klare Strukturen sind hierarchisch strukturierte Dokumente - Lexika, Kataloge oder Fachbücher werden beispielsweise nach Baumstrukturen erstellt. Darüber hinaus sind Software-Werkzeuge für die Verwaltung und Pflege derartiger Konzepte unerlässlich.

Wenn der linearen Verständlichkeit zu viel Bedeutung beigemessen wird, führt dies oft zu SoE-Konzepten, die an vielen Stellen sich wiederholende Erklärungen enthalten, die viel besser an einer zentralen Stelle im Dokument platziert wären. Dies führt unweigerlich zu Redundanzen im Konzept, zu mehr Text, der gelesen werden muss, und gleichzeitig zu einem erhöhten Risiko von Unstimmigkeiten und Lücken.

Ähnliche Ansätze zur Beschreibung von Konzepten finden sich z. B. in der Architektur (vgl. [AII1977]).

Kurz gesagt, ein SoE-Konzept ist weder ein Shakespeare-Stück noch ein Sachbuch für Kinder. Eine brauchbare Analogie für ein Konzept sind Baupläne. Es geht nicht in erster Linie darum, eine schöne Geschichte zu erzählen oder einen Sachverhalt anschaulich zu beschreiben, sondern darum, eine digitale Lösung in all ihren Facetten und in möglichst kompakter Form zu beschreiben, um sie beispielsweise umsetzen zu können oder den Aufwand für die Umsetzung abzuschätzen.

Um es ganz klar zu sagen: Ein Konzept, das leicht zu lesen und zu verdauen ist, kann durchaus als kleines Kunstwerk betrachtet werden. Die Begriffe „leicht verständlich“ und „schön zu lesen“ bedeuten im Zusammenhang mit Designkonzepten jedoch etwas ganz anderes als im Zusammenhang mit Romanen oder Sachbüchern.

Im Folgenden wird der Schreibstil für SoE-Konzepte näher erläutert.

2.2.4.1 *Der Wert der Nicht-Redundanz und Instrumente zur Vermeidung von Redundanz*

In erklärenden Texten wird die Wiederholung von Fakten oft als didaktisches Mittel verstanden, um das Lernen bestimmter Informationen zu fördern oder das Nachschlagen wichtiger Informationen an anderen Stellen eines Textes zu vermeiden.

Redundanzen erhöhen den Aufwand für das Lesen von Konzepten. Sie stellen ein Risiko für die Qualität der SoE-Konzepte und den Prozess ihrer Erstellung dar, da redundante Informationen eine Quelle für Missverständnisse und Inkonsistenzen sind und sie zusätzlich die Fähigkeit zur Änderung und Erweiterung der SoE-Konzepte beeinträchtigen. Außerdem erhöht Redundanz den Aufwand für die Erstellung und Pflege, da die Informationen redundant beschrieben and auch verändert werden müssen. Dieser Mehraufwand macht sich auch bei Copy & Paste bemerkbar, da Textfragmente in der Regel nicht eins zu eins von einem Abschnitt des Dokuments in einen anderen übertragen werden können. Wir empfehlen, die YPRC-Fallstudienkonzepte durchzusehen, um sich ein Bild von dieser Arbeitsweise zu machen.

Querverweise sind das stilistische Mittel der Wahl, um Redundanzen in SoE-Konzepten zu vermeiden. Das Prinzip für die Definition von Querverweisen ist einfach. Wenn eine Information (z. B. ein Regelwerk, ein Datensatz oder eine Funktion) mehr als einmal im Konzept benötigt wird, dann sollte diese Information isoliert beschrieben und mit einem Querverweis an den entsprechenden Stellen referenziert werden. Betrachten wir ein Beispiel mit einem hohen Maß an Redundanz (jeder Aufzählungspunkt steht für einen Baustein).

- A1: Der Benutzerdatensatz besteht aus einem Benutzernamen (min. 6 Zeichen, max. 15 Zeichen, keine Sonderzeichen) und einem Passwort (min. 8 Zeichen, max. 15 Zeichen, kein Dollarzeichen (\$)).
- A2: Das System muss den Zugang zu den Benutzerdaten durch Eingabe eines Benutzernamens (min. 6 Zeichen, max. 15 Zeichen, keine Sonderzeichen) und eines Passworts (min. 8 Zeichen, max. 15 Zeichen, kein Dollarzeichen (\$)) sicherstellen.
- A3: Wenn sich ein neuer Benutzer registriert, muss das System sicherstellen, dass:
 - Der Benutzername folgende Kriterien erfüllt: min. 6 Zeichen, max. 15 Zeichen, keine Sonderzeichen
 - Das Passwort folgende Kriterien erfüllt: min. 8 Zeichen, max. 15 Zeichen, kein Dollarzeichen (\$)
- A4: Wenn der Benutzer sein Passwort ändert, muss das System sicherstellen, dass das Passwort folgende Kriterien erfüllt: min. 8, max. 15 Zeichen, kein Dollarzeichen (\$).

Natürlich ist das Beispiel übertrieben, aber es soll deutlich machen, dass dieselben Informationen über die Struktur des Benutzernamens und des Passworts an verschiedenen Stellen benötigt werden. In diesem Beispiel ist die Redundanz sehr offensichtlich. In der täglichen Arbeit können die dargestellten Informationen jedoch in verschiedenen Teilen eines Dokuments enthalten sein und möglicherweise sogar von verschiedenen Personen formuliert werden.

Eine weniger redundante Form der Beschreibung kann durch die Verwendung von Querverweisen erreicht werden.

- A1: Der Benutzerdatensatz besteht aus einem Benutzernamen (Kriterien siehe A5) und einem Passwort (Kriterien siehe A6).
- A2: Das System muss den Zugang zu den Benutzerdaten durch Eingabe eines Benutzernamens (siehe A5) und eines Passworts (siehe A6) sicherstellen.
- A3: Bei der Registrierung eines neuen Benutzers muss das System sicherstellen, dass die Kriterien für den Benutzernamen (siehe A5) und das Passwort (siehe A6) erfüllt sind.
- A4: Wenn der Benutzer sein Passwort ändert, muss das System sicherstellen, dass die Kriterien für das Passwort (siehe A6) erfüllt sind.
- A5: Der Benutzername muss die folgenden Kriterien erfüllen: Min. 6 Zeichen, max. 15 Zeichen, keine Sonderzeichen.
- A6: Das Passwort muss die folgenden Kriterien erfüllen: Min. 8 Zeichen, max. 15 Zeichen, kein Dollarzeichen (\$).

Die zweite Form der Dokumentation verlagert die Kriterien für Benutzernamen und Kennwort in einen eigenen Baustein und verweist an allen geeigneten Stellen auf den entsprechenden Baustein. Dadurch werden redundante Beschreibungen der Kriterien vermieden und die Lesbarkeit der einzelnen Aussagen verbessert. Gleichzeitig wird die Änderbarkeit und Erweiterbarkeit verbessert, da die Kriterien für Benutzernamen oder Passwort nur an einer Stelle geändert oder erweitert werden müssen.

Der Nachteil dieser Redundanzreduzierung ist natürlich die fehlende unmittelbare Verfügbarkeit von Informationen über die Kriterien sowie die schlechtere lineare Lesbarkeit. Um auf diese Informationen zuzugreifen, müssen Sie beim Lesen des Dokuments an die entsprechende Stelle springen. Sie können diesen Aufwand durch den Einsatz geeigneter Werkzeuge minimieren. In der Praxis können viele Querverweise ein Hindernis darstellen, insbesondere bei der späteren Implementierung des Systems, da die Entwickler die Informationen für ihre Implementierungsaufgabe möglicherweise über ein ganzes Dokument verstreut finden.

Der Einsatz von integrierenden Strukturen (z. B. User Storys, siehe Kapitel 5), die die relevanten Informationen für eine Implementierungsaufgabe in geeigneter Form zusammenfassen, kann hier helfen.

2.2.4.2 Modularität der Information

Die Satzstruktur der natürlichen Sprache macht es möglich, jeden beliebigen Sachverhalt in einem einzigen Satz auszudrücken. Wir können Sätze mit dem einfachen Wort *und* verbinden und etwas mit Kommas in einen Satz einfügen, einfach so, ohne dass der Satz endet. Solche Konstruktionen finden sich häufig in Alltagstexten und dienen zum Beispiel dazu, relevante Informationen in einem Satz zusammenzufassen. Bei SoE-Konzepten ist die Kombination verschiedener Fakten in einem Satz wiederum eine Quelle der Redundanz und verringert die Änderbarkeit oder Erweiterbarkeit des Konzepts.

Hier ist ein Beispiel:

- A1: Das System muss den Zugang zu den Benutzerdaten mit Benutzernamen und Passwort sichern und den Zugang sperren, wenn das Passwort dreimal falsch eingegeben wird.

In diesem Beispiel wird die Art des Zugriffsschutzes mit einem Zugriffsschutzverhalten kombiniert (drei falsche Eingaben führen zur Sperrung). Auch wenn diese Kombination aus sprachlicher Sicht unproblematisch ist, ist die Kombination beider Aspekte im Hinblick auf gute Konzepte nicht wünschenswert. Eine Trennung in zwei Sätze schadet an dieser Stelle nicht und führt zu den beiden folgenden Aussagen:

- A1: Das System muss den Zugang zu den Benutzerdaten mit Benutzernamen und Passwort sichern.
- A2: Wenn das Passwort dreimal falsch eingegeben wird, muss das System den Zugang sperren.

Bei der Erstellung von Konzepten wird empfohlen, nur einen Sachverhalt (z. B. ein Problem, eine Funktion oder eine Anforderung) pro Baustein zu formulieren.

2.2.4.3 Beziehungen zwischen den Bausteinen

Neben dem Verständnis der einzelnen Bausteine eines Konzepts ist es auch wichtig, die Beziehungen zwischen diesen Bausteinen zu verstehen. Das Verständnis der Zusammenhänge ist wichtig für die Erstellung eines vollständigen und konsistenten Konzepts:

- *Vollständigkeit* bedeutet, dass das Konzept in sich vollständig ist, d. h. jede Information, auf die in einem Konzept verwiesen wird, ist tatsächlich vorhanden und im referenzierten Teil des Konzepts beschrieben. Beschreibt ein Konzept beispielsweise eine Funktion *zur Validierung des Geburtsdatums*, so bezieht sich diese auf das Attribut *Geburtsdatum* der Entität *Kunde*. Wenn das Attribut *Geburtsdatum* nicht in einem bestimmten Datenelement (*Entität*, siehe unten) des *Kunden* beschrieben wird, dann ist dieses Konzept unvollständig.
- *Konsistenz* bedeutet, dass eine bestimmte Information in einer Vorlage auf die gleiche Weise wie in allen anderen Vorlagen des Konzepts verwendet wird und dass die Informationen an verschiedenen Stellen des Konzepts nicht miteinander in Konflikt stehen. Wenn zum Beispiel eine Funktion *Geburtsdatum validieren* sich auf eine Entität *Kunde* als *Kunde* bezieht und eine andere Funktion *Adresse validieren* sich auf die Entität *Kunde* als *Benutzer* bezieht, dann ist dieses Konzept inkonsistent.

Für einen DDP auf Foundation Level Niveau ist es sinnvoll, zwischen der instruktiven und der konstruktiven Perspektive zu unterscheiden, um die Beziehungen zwischen den Bausteinen der SoE-Konzepte zu verstehen.

Die instruktive Perspektive liefert erklärende Informationen, die Form, Funktion und Qualität einer digitalen Lösung begründen. Wichtige instruktive Beziehungen sind:

- *Quellenbeziehung*: Beschreibt die Quelle eines bestimmten Elements; z. B. verlangt ein Strategiepapier eines Unternehmens die Entwicklung einer speziellen Funktionalität. In diesem Fall handelt es sich bei dem Strategiepapier um ein Dokument, das die Quelle für eine bestimmte Funktion ist.

- *Constraint-Beziehung*: Beschreibt eine Randbedingung (Constraint), die zu einem bestimmten Element geführt hat; z. B. eine Standardanforderung, dass jede geschäftliche Interaktion in einer speziellen Protokolldatei gespeichert wird.
- *Zielbeziehung*: Beschreibt das Ziel in Bezug auf die Motivation für ein bestimmtes Element. Zum Beispiel wird das Ziel *Support-Anleitung für neue Benutzer* formuliert. Um dieses Ziel zu erreichen, bietet die digitale Lösung einen Lernmodus, der den Benutzer in die wichtigsten Funktionen der Lösung einführt. Jedes Element dieses Lernmodus (z. B. ein spezielles User Interface) kann dann auf das jeweilige Ziel bezogen werden.

Die konstruktive Perspektive beschreibt die Beziehungen zwischen den Bausteinen, die die Form, Funktion und Qualität beschreiben, aus denen die digitale Lösung besteht. Wir werden auf die konstruktive Beziehung in Abschnitt 2.2.6 zurückkommen.

2.2.4.4 Lesbarkeit versus strukturierte Dokumentation

Wir möchten diesen Abschnitt mit einigen Bemerkungen zur Lesbarkeit abschließen. Der in diesem Kapitel vorgestellte Ansatz für die Erstellung von System- und Elementdesignkonzepten wird zu hoch strukturierten Konzepten führen. Unsere Erfahrung zeigt, dass diese Konzepte für Stakeholder, die nicht in der Arbeit mit solchen Strukturen geschult sind, schwer zu verstehen sind. Wir haben die Erfahrung gemacht, dass Stakeholder, die regelmäßig mit solchen Konzepten zu tun haben, sich an die Struktur gewöhnen und stark strukturierte Konzepte sogar schätzen.

Dies bedeutet jedoch nicht, dass diese Konzepte allen Stakeholdern ohne weitere Anpassungen vorgelegt werden sollten. Denken Sie daran: Digital Design bedeutet, dass Sie für die richtige Kommunikation mit allen Stakeholdern verantwortlich sind. Diese Verantwortung kann bedeuten, dass Designkonzepte für bestimmte Stakeholder (z. B. die Unternehmensleitung) gesondert erstellt werden müssen, damit diese den Inhalt der Konzepte mit angemessenem Aufwand verstehen und Rückmeldungen geben können.

2.2.5 Dokumentationstechniken für die System- und Elementebene

Im Folgenden stellen wir eine Reihe von Dokumentationstechniken und Vorlagen vor, die für die Arbeit mit Designkonzepten auf System- und Elementebene nützlich sind.

2.2.5.1 Eine Anmerkung zur Idee der perfekten Technologie

Die Annahme einer perfekten Technologie im Vergleich zur realen Technologie [WaMe1986] unterstützt das Design digitaler Systeme und ihrer Elemente. Es bedeutet insbesondere fehlerfreie Technologie sowie unendliche Rechenkapazität, Speicherkapazität und unendliche Kommunikationskapazität. Diese Annahme vereinfacht die Arbeit an Designkonzepten, da die Grenzen der Technologie im Designkonzept nicht berücksichtigt werden müssen.

Die Annahme einer perfekten Technologie erstreckt sich jedoch nicht auf die Benutzer oder auf bestehende Systeme. Unerwartetes Verhalten von Benutzern muss berücksichtigt werden (z. B. bei der Definition von Use Cases, siehe Abschnitt 2.2.5.6). Auch die vorübergehende Nichtverfügbarkeit bestehender Systeme (z. B. eines Zahlungsanbieters) muss bei der Erstellung von Konzepten berücksichtigt werden.

Die Annahme einer perfekten Technologie ist nicht frei von Überlegungen zu den Grenzen der Technologie

Die Annahme einer perfekten Technologie ist lediglich eine Vereinfachung bei der Erstellung von Designkonzepten. Im weiteren Verlauf des Bauprozesses müssen die Annahmen der perfekten Technologie Schritt für Schritt abgebaut werden. Dies geschieht durch intensive Zusammenarbeit mit dem Tätigkeitsbereich Konstruktion (siehe Abschnitt 1.3).

2.2.5.2 Eine allgemeine Bausteinvorlage

Die Verwendung von Vorlagen ist eine bewährte Technologie zur Bereitstellung einer Referenzstruktur, um Elemente von Konzepten oder Spezifikationen (vgl. [CPRE2020]) auf modulare Weise darzustellen, wie in Abschnitt 2.2.4 vorgestellt. Eine allgemeine Bausteinvorlage besteht aus den folgenden Abschnitten⁷:

- Identifikationsnummer mit Titel
- Beziehungen zu anderen Elementen
- Beschreibung des jeweiligen Elements

Praxistipp

Ein pragmatischer Ansatz für die Definition der Identifikationsnummer ist der Bezug auf die Art des Elements mit einer Abkürzung und einer Nummer (z. B. F-3 für Funktion Nr. 3 oder UC-12 für Use Case Nr. 12). Auf diese Weise kann der Leser eines Designkonzepts anhand der Identifikationsnummer erkennen, auf welchen Typ Bezug genommen wird. Eine wichtige Regel ist, dass Identifikationsnummer nicht wiederverwendet werden dürfen und dass eine alphanumerische Sortierung der Identifikationsnummern nicht notwendig ist. Am Anfang braucht man etwas Zeit, um sich daran zu gewöhnen, da man dies von den Kapitelnummern gewohnt ist. In der Praxis stellt dies jedoch kein Problem dar.

Optionale Bestandteile einer Basisvorlage sind:

- Quelle (z. B. nützlich für die Referenzierung von Zusatzinformationen)
- Status (z. B. um zu beschreiben, ob ein Element beispielsweise vereinbart, in Umsetzung oder erledigt ist)
- Änderungsprotokoll (z. B. zur Dokumentation der Entwicklung eines Elements)

Die Verwendung der optionalen Abschnitte hängt von dem jeweiligen Bauprozesses ab.

2.2.5.3 Zielvorlage

Die Ziele werden in der Regel aus dem Lösungsdesignkonzept abgeleitet (siehe Tabelle 4). Gute Quellen für Ziele sind das Vision Statement, das Wertversprechen und das Geschäftsmodell (siehe Abbildung 18). Dennoch ist es nicht sinnvoll, einfach alle Ziele im Systemdesignkonzept zu wiederholen. Die Beschreibung der Ziele sollte sich auf Ziele beschränken, die einen klaren Bezug zu den Elementen des digitalen Systems haben.

Eine Zielvorlage kann verwendet werden, um jedes einzelne Ziel zu erfassen. Abbildung 19 zeigt eine beispielhafte Zielvorlage mit Inhalten aus der YPRC-Fallstudie. Die Überschrift enthält die ID (G für goal), die Nummer des Elements (hier 1) und den Titel. Der Titel eines Ziels sollte eine

⁷ Für Fachleute der Informationsmodellierung kann diese Grundvorlage als Superklasse (oder Superentität) betrachtet werden, von der andere Klassen abgeleitet werden.

knappe Zusammenfassung des Ziels sein. Weitere Einzelheiten zu diesem Ziel werden in der Beschreibung folgen.

Hinweise zur Erstellung der Beschreibung

Die Beschreibung besteht aus einer genaueren Erläuterung des Ziels, einschließlich einer Begründung, warum das Ziel für das digitale System wichtig ist. In dem genannten Beispiel bezieht sich das Ziel auf den Läufer; daher muss das Ziel die Bedeutung des Ziels für den Läufer beschreiben.

Es ist schwierig, eine gute Zielbeschreibung zu formulieren, insbesondere im Hinblick auf die Funktionen eines digitalen Systems. Eine gute Faustregel für die Zielformulierung ist es, die einzelnen Elemente des digitalen Systems in der Zielformulierung nicht zu erwähnen, da sie die Mittel zum Erreichen eines Ziels sind und daher nicht Teil des Ziels selbst sein sollten. Des Weiteren ist es sinnvoll, zwischen den folgenden zwei Arten von Zielen zu unterscheiden:

- *Harte Ziele* ermöglichen eine objektive Messung.
- *Weiche Ziele* ermöglichen eine subjektive Messung.

Wenn möglich, sollten weiche Ziele durch detaillierte harte Ziele oder andere Kriterien, die eine Bewertung der Zielerfüllung ermöglichen, konkretisiert werden.

Die Konkretisierung eines Ziels kann in die Beschreibung aufgenommen werden, um zu erläutern, wie ein Ziel durch das digitale System erreicht werden soll. Mit diesen zusätzlichen Angaben wird eine klare Grenze zwischen dem Ziel selbst (erster Absatz) und der Zielerreichung (Absätze zwei und drei) gezogen.

G-1 Provide information on current performance to a runner during a training session



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- [DDev-1 Runner's Watch](#)
- [DSys-1 Runner's App](#)
- [U-1 Runner](#)

Description:

Information about the current training performance is important to allow runners (U-1) to achieve the best possible results. Therefore, runners should be continuously informed about their training performance during their training session.

To achieve this goal, the watch (DDev-1) shall:

1. Inform the runner about their current heart rate
2. Inform the runner visually and non-visually about their performance

Furthermore, the app (DSys-1) shall:

1. Inform the runner about their current heart rate and performance
2. Summarize the training performance of the runner at the end of the training session

Abkürzungen (nicht Teil der Vorlage)

G-x	Ziel (Goal)
DDev-x	Digitales Gerät
DSys-x	Software-System
U-x	Benutzer (User)

Abbildung 19 - Eine Zielvorlage/Beispiel aus der YPRC-Fallstudie

Darüber hinaus ermöglicht die Beschreibung der Zielerreichung eine explizite Definition der wichtigen Elemente des digitalen Systems, die zur Erreichung des Ziels erforderlich sind. Diese Beziehungsbeschreibungen machen dem Leser wichtige Designentscheidungen für das digitale System deutlich. Es wurde in diesem Beispiel entschieden, dass die Smartwatch und die App dafür verantwortlich sind, den Läufer über seine Trainingsleistung zu informieren. Die detaillierten Aussagen können dann als Ziele für die einzelnen Elemente betrachtet und ausdrücklich im Ziele-Abschnitt des jeweiligen Elements dokumentiert werden.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element aus dem Systemdesignkonzept, das im Text erwähnt wird und einen Bezug zu dem beschriebenen Ziel hat, wird erwähnt.

Praxistipp

Die Ziele ändern sich häufig während der Erstellung eines Systemdesignkonzepts. Sie bieten jedoch einen guten Ausgangspunkt für das Verständnis und die Strukturierung eines digitalen Systems. Wir empfehlen daher, die Ziele eines digitalen Systems zu Beginn detailliert zu definieren. Dabei wird in der Regel eine detaillierte Liste von Zielen erstellt, die im weiteren Verlauf des Prozesses überarbeitet werden muss. Diese Überarbeitung der Ziele ist sehr wichtig, um ein klares Systemdesignkonzept zu erstellen. Die Überarbeitung umfasst insbesondere das Streichen von Zielen, die für das digitale System nicht mehr als relevant angesehen werden.

Schließlich kann die Liste der Ziele für ein digitales System kurz sein. Drei bis fünf Ziele auf der richtigen Detailebene sind oft ausreichend, um die Kernziele eines digitalen Systems zu erfassen. Längere Listen könnten Ziele auf einer Detailebene beschreiben, die typischerweise auf die Elementebene gehört. Das Ziel in Abbildung 19 zum Beispiel besagt nur, dass der Läufer über seine Trainingsleistung informiert werden soll. In dem Ziel wird nicht näher ausgeführt, wie diese Informationen erfolgen oder wie sie gesammelt werden.

2.2.5.4 Randbedingungs-vorlage

Wie bei der Zieldokumentation lassen sich einige Randbedingungen aus dem Lösungsdesignkonzept ableiten. Zusätzliche Randbedingungen können jedoch auch aus verschiedenen anderen Quellen resultieren. Oftmals werden Randbedingungen auftreten, wenn weitere Details eines digitalen Systems ausgearbeitet werden. Gute Beispiele sind technische oder rechtliche Randbedingungen.

Für die Erfassung jeder einzelnen Randbedingung eines digitalen Systems kann jeweils eine eigene Vorlage verwendet werden. Abbildung 20 - Eine Randbedingungs-vorlage zeigt eine beispielhafte Randbedingungs-vorlage mit Inhalten aus der YPRC-Fallstudie. Die Überschrift enthält die ID (Con für Randbedingung (engl. Constraint)), die Nummer des Elements (hier 2) und den Titel. Der Titel einer Randbedingung sollte eine kurze Zusammenfassung der Randbedingung sein. Weitere Details können in der Beschreibung folgen.

Hinweise zur Erstellung der Beschreibung

Die Beschreibung sollte eine detaillierte Erläuterung der Randbedingung enthalten, einschließlich einer kurzen Begründung, warum die Randbedingung für das digitale System wichtig ist. Darüber hinaus sollten die Elemente, die der Randbedingung entsprechen müssen, ausdrücklich genannt werden.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element aus dem Systemdesignkonzept, das im Text erwähnt wird und einen Bezug zu der beschriebenen Randbedingung hat, wird erwähnt.

Con-2 Encrypt training data transfer



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- [DDev-1 Runner's Watch](#)
- [DSys-1 Runner's App](#)
- [DSys-2 Coaching Portal](#)
- [ESys-3 Web browser installed on the coach's PC](#)

Description:

The training data and the voice data are personal data and legislation requires that the transfer of this data is encrypted to avoid data theft. This constraint applies to the following communication channels:

- Communication between watch (DDev-1) and app (DSys-1)
- Communication between app (DSys-1) and portal (DSys-2)
- Communication between portal (DSys-2) and browser of the coach (ESys-3)

Abkürzungen (nicht Teil der Vorlage)

DDev-x	Digitales Gerät
DSys-x	Software-System
ESys-x	Vorhandenes System

Abbildung 20 - Eine Randbedingungs-vorlage

Praxistipp

Die Festlegung von Randbedingungen für ein digitales System ist ein fortlaufender Prozess während des gesamten Bauprozesses. Es ist sinnvoll, zu Beginn der Erstellung des Systemdesignkonzepts einige Zeit mit dem Sammeln von Randbedingungen zu verbringen. Zu den möglichen Quellen für Randbedingungen gehören die Gesetzgebung, die rechtlichen Anforderungen eines Bereichs oder die IT-Governance einer Organisation. Dennoch sollten Sie sich darüber im Klaren sein, dass der Randbedingungen-Teil des Systemdesignkonzepts ständig aktualisiert werden muss. Vor allem in späteren Phasen des Bauprozesses wird der Abschnitt über die Randbedingungen oft vernachlässigt, da diese für die am Bauprozess Beteiligten offensichtlich sind. Ein gut ausgebildeter DDP ist sich dessen bewusst und nimmt die Dokumentation von Randbedingungen ernst.

2.2.5.5 Form und Funktion auf der Systemebene

Form bedeutet auf der Systemebene die Beschreibung aller Elemente, aus denen das digitale System besteht, das die digitale Lösung realisiert. Dazu gehören die Benutzertypen, die vorhandenen Elemente/Objekte und die zu realisierenden Elemente. Funktion bedeutet auf der Systemebene die Beschreibung der von den Elementen des Systems als Ganzes bereitgestellten Funktionalität.

2.2.5.5.1 Form: Benutzertyp-Vorlage

Abbildung 21 - Eine Benutzertyp-Vorlage zeigt eine beispielhafte Benutzertyp-Vorlage für den Läufer aus der YPRC-Fallstudie. Die Überschrift enthält die ID (U für User), die Nummer des Elements (hier 1) und den Titel.

Hinweise zur Erstellung der Beschreibung

Die Beschreibung ist ein kurzer Satz, der den Benutzertyp charakterisiert.

Hinweise für die Dokumentation von Beziehungen

Neben der Beschreibung sollte die Benutzertyp-Vorlage Verweise auf alle anderen Bausteine enthalten, die für den Benutzertyp wichtig sind. Dies sind:

- Ziele (Instruktive Beziehung), die für den beschriebenen Benutzertyp relevant sind
- Szenarien (konstruktive Beziehung), an denen der Benutzer beteiligt ist
- Elemente (konstruktive Beziehung), mit denen der Benutzer interagiert

Praxistipp

Die Informationen in dem Element sind an sich sehr kurz gehalten. Die Beschreibung des Benutzers ist jedoch von großer Bedeutung für die Nachvollziehbarkeit über alle Designkonzepte hinweg. Eine gesonderte Beschreibung ist z. B. insbesondere bei Systemen mit unterschiedlichen Benutzertypen wichtig, um einen eindeutigen Bezug zu dem in Use Cases oder Szenarien betrachteten Benutzertyp herzustellen.

U-1 Runner



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- [Scen-1 "AI shows the runner that they took on too much in a training session"](#)
- [G-1 Provide information on current performance to a runner during a training session](#)
- [G-2 Offer AI coaching tips to the runner based on previous and current training data](#)
- [G-3 Offer affordable remote personal coaching during training](#)

Description:

A person who wants to train in long-distance running.

Abkürzungen (nicht Teil der Vorlage)

G-x	Ziel (Goal)
Scen-x	Szenario

Abbildung 21 - Eine Benutzertyp-Vorlage

2.2.5.5.2 Form: Vorlage für ein vorhandenes Objekt

Abbildung 22 - Vorlage für ein bestehendes Objekt zeigt eine beispielhafte Objektvorlage mit Inhalten aus der YPRC-Fallstudie. Die Überschrift enthält die ID (Obj für Objekt), die Nummer des Elements (hier: 1) und den Titel.

Hinweise zur Erstellung der Beschreibung

Die Beschreibung sollte aus einer kurzen verbalen Erläuterung des vorhandenen Objekts bestehen, einschließlich eines Überblicks über die Funktionen, die das Objekt innerhalb des Systems bietet. Es sollten Verweise auf andere Elemente des Systems angegeben werden, um die Schnittstellen des beschriebenen Objekts zu anderen Elementen des Systems zu verdeutlichen.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element aus dem Systemdesignkonzept, das im Text erwähnt wird und eine Beziehung zu dem beschriebenen Objekt hat, wird erwähnt.

Praxistipp

Ähnlich wie beim Benutzertyp ist die Beschreibung der vorhandenen Objekte an sich nicht sehr aussagekräftig. Allerdings ist hier eine explizite Dokumentation für die Traceability über alle Konzepte hinweg wichtig. Darüber hinaus dokumentiert die Beschreibung der vorhandenen Objekte wesentliche Annahmen über die digitale Lösung, nämlich die Objekte, die vorausgesetzt werden, um das digitale System nutzen zu können.

Obj-1 Runner's Smartphone



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- [DDev-1 Runner's Watch](#)
- [DSys-1 Runner's App](#)
- [DSys-2 Coaching Portal](#)
- [ESys-1 Map Server](#)
- [ESys-2 Payment Provider](#)
- [U-1 Runner](#)

Description:

A smartphone that the runner (U-1) already owns and on which the runner's app (DSys-1) is installed. The smartphone provides the following functions to the app:

- Interaction with the runner via the smartphone user interface
- Internet connection to the portal (DSys-2)
- Internet connection to the map server (ESys-1)
- Internet connection to the payment provider (ESys-2)
- GPS for position and speed determination
- Date and time via smartphone clock
- Connection between app (DSys-1) and watch (DDev-1) via Bluetooth
- Connection between app (DSys-1) and headset (Obj-2) via smartphone audio

Abkürzungen (nicht Teil der Vorlage)

DDev-x	Digitales Gerät
DSys-x	Software-System
ESys-x	Vorhandenes System
U-x	Benutzer (User)

Abbildung 22 - Vorlage für ein bestehendes Objekt

2.2.5.5.3 Form: Vorlage für ein bestehendes System

Abbildung 23 - Eine Vorlage für ein bestehendes System zeigt beispielhaft eine Vorlage für ein bestehendes System mit Inhalten aus der YPRC-Fallstudie. Die Überschrift enthält die ID (ESys für das bestehende System), die Nummer des Elements (hier 1) und den Titel. Die Dokumentation der bestehenden Systeme ist wichtig für die Beschreibung des gesamten digitalen Systems und für das Verständnis ihres Beitrags zur digitalen Gesamtlösung.

Hinweise zur Erstellung der Beschreibung

Die Beschreibung in der Vorlage sollte einen kurzen Überblick über den Beitrag des Elements geben, z. B. Daten oder Dienste, die anderen Elementen zur Verfügung gestellt werden.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element aus dem Systemdesignkonzept, das im Text erwähnt wird und einen Bezug zu dem beschriebenen System hat, wird erwähnt.

Praxistipp

Die Dokumentation bestehender Systeme, analog zu bestehenden Objekten, ist wichtig für die Nachvollziehbarkeit und für die Dokumentation von Annahmen über bestehende Systeme. Das YPRC ist beispielsweise zwingend auf einen Landkartenserver angewiesen. Ohne ein solches bestehendes System wäre YPRC nicht funktionsfähig.

ESys-1 Map Server



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- DSys-1 Runner's App
- DSys-2 Coaching Portal

Description:

The map server provides current map data for display in the app (DSys-1) or portal (DSys-2). The map server is an external system and is purchased as a web service.

Abkürzungen (nicht Teil der Vorlage)
DSys-x Software-System

Abbildung 23 - Eine Vorlage für ein bestehendes System

2.2.5.5.4 Form: Digitale Geräte und Software Elementvorlage

Selbst einfache digitale Systeme bestehen aus zwei oder mehr Elementen, die realisiert werden sollen. Die recht einfache YPRC-Fallstudie besteht aus drei Elementen (Smartwatch, App und Portal). Die wichtige Triebkraft für die Elemente eines digitalen Systems ist oft die zugrundeliegende Form, mit Elementen, die Daten und Funktionen für ein wahrnehmbares Element bereitstellen. Die Erfassung dieser Strukturen der zugrundeliegenden Form ist eine wichtige Aufgabe der Systemebene.

Für jedes Element wird eine eigene Vorlage erstellt. Um zwischen den verschiedenen Arten von Elementen (Geräte- und Softwareelemente) zu unterscheiden, führen wir zwei verschiedene

Vorlagen ein: Eine für Geräte und eine für Softwareelemente. Der Hauptunterschied zwischen den beiden Vorlagen ist die ID, die den Typ des Elements angibt.

Abbildung 24 - Eine Gerätevorlage zeigt eine beispielhafte Gerätevorlage und Abbildung 25 - Eine Software-Vorlage zeigt eine beispielhafte Softwareelementvorlage mit dem Inhalt der YPRC-Fallstudie. Die Überschrift enthält die ID (DDev für Gerät und DSys für Softwareelement), die Nummer des Elements (hier 1 für beide Beispiele) und den Titel. Der Titel sollte ein kurzer und eindeutiger Name für das Element sein, der einprägsam und verständlich ist.

Hinweise zur Erstellung der Beschreibung

Die Beschreibung sollte einen kurzen Überblick über das Element geben, ohne zu viele Details über die Funktionalität zu liefern. Sie sollte sich auf die Daten konzentrieren, die das Element speichern wird (denken Sie daran, dass Daten auch Teil der Form sind) und auf die Beziehung zu anderen Elementen des digitalen Systems. Einzelheiten, die zum Verständnis der Form des Elements erforderlich sind, können ebenfalls genannt werden. Die Smartwatch des Läufers wird zum Beispiel eine Taste haben, die eine Interaktion mit dem Läufer ermöglicht. Im Falle eines digitalen Systems sollte sich die Beschreibung auf das vorhandene Gerät beziehen, von dem angenommen wird, dass es den Softwareteil des digitalen Systems betreibt. Auf den ersten Blick scheint die Beschreibung eines Elements auf der Systemebene recht kurz und nicht wirklich nützlich zu sein. Aber wenn man sich kurzfasst, kann man ein klares Verständnis für diese Elemente erreichen.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element aus dem Systemdesignkonzept, das im Text erwähnt wird und einen Bezug zu dem beschriebenen Element hat, wird erwähnt.

DDev-1 Runner's Watch



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- [DSys-1 Runner's App](#)
- [Obj-1 Runner's Smartphone](#)
- [U-1 Runner](#)

Description:

This device is used for measuring heart rate of the user (U-1) and for displaying important training data. A vibration alarm informs the runner about important events. It has a Bluetooth connection to the runner's smartphone (Obj-1) to connect with the runner's app (DSys-1). It further has a button to control the functions of watch.

Abkürzungen (nicht Teil der Vorlage)

DSys-x	Software-System
Obj-x	Vorhandenes Objekt
U-x	Benutzer (User)

Abbildung 24 - Eine Gerätevorlage

DSys-1 Runner's App



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- [DDev-1 Runner's Watch](#)
- [DSys-1 Runner's App](#)
- [DSys-2 Coaching Portal](#)
- [ESys-1 Map Server](#)
- [ESys-2 Payment Provider](#)
- [Obj-1 Runner's Smartphone](#)
- [Obj-2 Runner's Headset](#)
- [U-1 Runner](#)

Description:

The app is the central element of the digital solution and is installed on the runner's smartphone (Obj-1). It stores the training data from the watch (DDev-1) via Bluetooth, establishes the connection to the coaching portal (DSys-2), and displays relevant information about the route to the runner (U-1). For example, the runner can view their running distance on a map (ESys-1) or their current speed. The runner can also view data from previous training sessions.

The runner can further book and pay for AI or personal coaching services with the app. Therefore, the app is connected to a payment provider (ESys-2) and can provide a voice connection to a coach for the runner via a headset (Obj-2).

Abkürzungen (nicht Teil der Vorlage)

DSys-x	Software-System
ESys-x	Vorhandenes System
Obj-x	Vorhandenes Objekt
U-x	Benutzer (User)

Abbildung 25 - Eine Software-Vorlage

Praxistipp

Denken Sie daran, dass das Systemdesignkonzept die dokumentierte Darstellung unseres Verständnisses des digitalen Systems ist. Wenn Sie nicht in der Lage sind, jedes Element eines digitalen Systems kompakt und klar zu beschreiben, ist es sehr wahrscheinlich, dass Sie die einzelnen Elemente des digitalen Systems nicht richtig verstehen. Daher ist diese kurze Beschreibung ein guter Maßstab für Ihr eigenes Verständnis der einzelnen Elemente des digitalen Systems, und die Mühe, diese kurze Beschreibung zu schreiben, ist gut investiert, um sich zu vergewissern, dass Sie das System, das Sie gerade entwerfen, verstanden haben.

Außerdem werden viele Stakeholder auf dieser Verständnisebene arbeiten, und die kurze Beschreibung der einzelnen Elemente wird in späteren Phasen des Bauprozesses sehr nützlich sein, um über das digitale System zu kommunizieren. Das bedeutet, dass diese Beschreibungen während des gesamten Bauprozesses beibehalten werden müssen.

2.2.5.5.5 Funktion: Szenario-Vorlage

Auf der Systemebene wird die Funktion durch die jeweiligen Elemente erzeugt und kann sehr kompliziert sein. Wenn Sie Einsteiger im Digital Design sind, empfehlen wir, um den richtigen Detaillierungsgrad beizubehalten, die Funktion eines digitalen Systems anhand von Szenarien zu beschreiben.

Als Faustregel gilt, dass für jedes Ziel, das ein digitales System erreichen soll (siehe Kapitel 3 des obigen Systemdesignkonzepts in Abschnitt 2.2.2.3), ein Szenario beschrieben werden sollte. Für jedes Szenario wird eine Szenario-Vorlage erstellt.

Hinweise zur Erstellung der Beschreibung

Abbildung 26 zeigt eine beispielhafte Szenario-Vorlage mit Inhalten aus der YPRC-Fallstudie. Die Überschrift enthält die ID (Scen für Szenario), die Nummer des Elements (hier: 1) und den Titel. Der Titel eines Szenarios sollte eine Zusammenfassung des Szenarios in einem Satz sein. Weitere Details können in der Beschreibung folgen.

Wer mit dem Schreiben von Szenarien beginnt, sollte sich an fünf einfache Schreibregeln halten:

- 1) Ein Ziel pro Szenario: Ein Szenario sollte die Erreichung eines bestimmten Ziels auf genau eine Weise erklären. Beschreiben Sie keine Alternativen; vermeiden Sie if-then-else-Strukturen.
- 2) Stellen Sie das Szenario zu Beginn in einem Satz vor.
- 3) Schreiben Sie jeden Satz in der Struktur *Subjekt - Prädikat - Objekt*; Subjekte sind Elemente des digitalen Systems.
- 4) Maximal eine Interaktion pro Satz: Verwenden Sie einfache Sätze und beschreiben Sie eine Interaktion zwischen zwei Elementen des digitalen Systems in einem Satz.
- 5) Nehmen Sie den Verweis auf die Elementvorlagen auf, um die Interaktion deutlicher zu machen.

Diese Regeln führen zu sehr vereinfachten, aber klaren Szenariobeschreibungen, wie folgendes Beispiel zeigt. Einige zusätzliche Details (z. B. der Hinweis auf den sonnigen Morgen im Beispiel) sind natürlich erlaubt, um die Szenarien lesbarer zu machen und eine verständliche Geschichte zu schaffen. Zusammen mit Personas sind derartige Szenarien ein zentrales Gestaltungsmittel, um die digitale Lösung zu reflektieren und Empathie mit den Benutzern zu erzeugen.

Scen-1 "AI shows the runner that they took on too much in a training session"



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- [DDev-1 Runner's Watch](#)
- [DSys-1 Runner's App](#)
- [Obj-1 Runner's Smartphone](#)
- [U-1 Runner](#)

Description:

The runner (U-1) Marcus starts a training session on a sunny morning. He starts the app (DSys-1) on his smartphone (Obj-1) and watches his last training session. He chooses a challenging route with steep inclines to test his fitness. At the push of a button on the watch (DDev-1), he starts his workout and starts running. On a steep incline, the watch (DDev-1) warns him that his pulse is very high. He then slows down. At the end of the training he is exhausted and has to recover on a park bench. In the evening, Marcus checks his app (DSys-1). In the meantime, the portal (DSys-2) has analyzed his training data and the AI coach recommends that Marcus slow down in his next session because he took on too much during the steep incline. Marcus plans to run slower next time.

Abkürzungen (nicht Teil der Vorlage)

DSys-x	Software-System
ESys-x	Vorhandenes System
Obj-x	Vorhandenes Objekt
U-x	Benutzer (User)

Abbildung 26 - Eine Szenario-Vorlage

Der Verweis auf die Kennungen der Elementvorlagen (Regel 4), wie im Beispiel gezeigt, erschwert das einfache Lesen des Szenarios. Dennoch sind diese Hinweise wichtig, um deutlich

zu machen, welches Element des digitalen Systems in welchem Schritt des Szenarios wie beteiligt ist. Ohne den Verweis U-1 ist zum Beispiel nicht klar, dass der Läufer Marcus ein Benutzer von YPRC ist.

Das Beispiel zeigt ein Szenario mit einem beteiligten Benutzer. Beachten Sie, dass ein Szenario auch eine Funktion beschreiben kann, die aus einer reinen zugrundeliegenden Form besteht. In einem solchen Szenario gibt es keinen Benutzer, sondern nur technische Interaktionen zwischen den Elementen. Solche Szenarien finden sich vor allem bei digitalen Lösungen, die eine Automatisierung zum Ziel haben - zum Beispiel die Steuerung einer Heizungsanlage in einem intelligenten Gebäude in Abhängigkeit von der verfügbaren Sonnenenergie und anderen Energiequellen.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element aus dem Systemdesignkonzept, das in der Beschreibung erwähnt wird und einen Bezug zu dem beschriebenen Szenario hat, wird aufgelistet. Darüber hinaus sollte das Ziel, das in diesem Szenario erreicht wird, in die Liste aufgenommen werden.

Praxistipp

Textuelle Szenarien nach den fünf Regeln sind natürlich eine sehr eingeschränkte Technik, um die Funktion eines digitalen Systems zu beschreiben. Für Einsteiger im Digital Design sind diese eingeschränkten Szenarien jedoch ein guter Ausgangspunkt, um die Beschreibung der Funktionalität auf Systemebene zu üben, und sie sind besonders nützlich, um den richtigen Detaillierungsgrad für das Systemdesignkonzept zu treffen.

2.2.5.6 Form und Funktion auf der Elementebene

Auf der Elementebene bedeutet „Form“ die Beschreibung der Struktur eines bestimmten Elements in Beziehung zu den anderen Elementen. Bei der Beschreibung der Form unterscheiden wir zwischen folgenden Bausteinen:

- Zugrundeliegende und wahrnehmbare Hardware-Schnittstellenvorlagen
- Vorlagen für das User Interface, um die Interaktion mit den Benutzern zu detaillieren
- Software-Schnittstellenvorlagen, um die Interaktion mit anderen Softwareelementen zu beschreiben
- Entitätsvorlagen zur Dokumentation der Daten
- Teilevorlagen zur Beschreibung von Teilen eines Geräts

Auf der Elementebene bedeutet Funktion die Beschreibung der Funktionalität, die ein bestimmtes Element den Benutzern oder anderen Elementen des digitalen Systems bietet. Bei der Beschreibung der Funktion unterscheiden wir zwischen folgenden Bausteinen:

- Funktionsvorlage zur Dokumentation der zugrundeliegenden Funktionen
- Use Case Vorlage zur Dokumentation der wahrnehmbaren Funktionen

2.2.5.6.1 Form: Hardware-Schnittstellenvorlage

Schnittstellen sind ein wichtiger Bestandteil der Elemente eines digitalen Systems. Sie ermöglichen die Interaktion zwischen verschiedenen Elementen und sind wie folgt definiert:

Interface: A shared boundary across which information is passed.

In Bezug auf die Hardware definieren wir eine Hardware-Schnittstelle wie folgt:

Hardware interface: An interface between an element of a system and a device.

Diese Definition ist eher abstrakt, um ein breites Spektrum an möglichen Ebenen von Schnittstellen abzudecken.

Wir unterscheiden zwischen *wahrnehmbaren Hardware-Schnittstellen*, die es den Benutzern ermöglichen, mit einem Gerät zu interagieren, und *zugrundeliegenden Hardware-Schnittstellen*, bei denen ein Systemelement mit einem Gerät auf eine Weise interagiert, die für die Benutzer des Systems nicht wahrnehmbar ist. In digitalen Systemen gehören zu den Hardware-Schnittstellen beispielsweise Anzeigen, Audioein- und -ausgang sowie Kommunikationshardware.

Hinweise zur Erstellung der Beschreibung

Auf Foundation Level Niveau genügt es, die wahrnehmbaren und zugrundeliegenden Hardware-Schnittstellen in Textform zu beschreiben. Die Beschreibung sollte eine kurze Erläuterung der Schnittstelle enthalten, einschließlich des anderen Elements des Systems, mit dem die Schnittstelle verbunden ist. Die Beschreibung der Schnittstelle sollte auch die Funktion, den Use Case oder die Entität nennen, die für die Schnittstelle relevant ist.

UHI-1 Smartphone GPS Interface



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- Obj-1 Runner's Smartphone
- F-2 Record training data and average values during solo training

Description:

The runner's smartphone (Obj-1) is expected to have a GPS module. This GPS module can be accessed through this interface to obtain the following information for the app:

- Current position as GPS coordinates
- Current speed in km/h
- Current altitude in meters above sea level

Abkürzungen (nicht Teil der Vorlage)

F-x	Funktion
Obj-x	Vorhandenes Objekt

Abbildung 27 - Eine Hardware-Schnittstellenvorlage

Abbildung 27 zeigt ein Beispiel für eine zugrundeliegende Hardware-Schnittstellenvorlage aus der YPRC-Fallstudie. Beachten Sie den Verweis auf das Smartphone des Läufers als Verweis auf das Objekt auf Systemebene, das diese Schnittstelle bereitstellt. Beachten Sie auch den Verweis auf die Funktion F-2, die Daten von dieser Schnittstelle verwendet wird.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Denken Sie daran, dass eine Schnittstelle Beziehungen zu anderen Elementen auf der Systemebene hat; daher sollten Sie das jeweilige Element aus dem Systemdesignkonzept erwähnen. Es ist auch möglich, die entsprechende Schnittstelle aus dem Elementdesignkonzept des referenzierten Elements zu nennen.

Praxistipp

Für Einsteiger ist es sinnvoll, zwischen den zugrundeliegenden Hardware-Schnittstellen, die Verbindungsmöglichkeiten bieten, und den Hardware-Schnittstellen zu unterscheiden, die den Zugang zu weiteren Fähigkeiten des Geräts ermöglichen, mit dem das Softwareelement betrieben wird (z. B. Zugang zu Ortungssensoren).

Typische zugrundeliegende Hardware-Schnittstellen, die Verbindungsmöglichkeiten bieten, sind:

- Bluetooth oder andere drahtlose Netzwerkverbindungen
- Ortungssensoren für den Zugriff auf Daten aus dem Positionsnetz (z. B. GPS oder Galileo)
- Internetverbindung (z. B. LTE oder Wireless LAN) zur Verbindung mit bestehenden Systemen

Vor allem die Internetverbindung wird oft als selbstverständlich angesehen. Dennoch ist es sehr wichtig, diese zugrundeliegende Hardware-Schnittstelle zu dokumentieren, da die Internetverbindung in der Regel einen erheblichen Einfluss auf die Fähigkeiten und Qualitäten eines Elements und damit auf die gesamte digitale Lösung hat (siehe Abschnitt 2.1.3).

Geräte bieten auch zusätzliche Fähigkeiten/Informationen für ein Softwareelement. Typische Beispiele für solche zugrundeliegenden Hardware-Schnittstellen sind:

- Smartwatch, um Informationen über Zeit und Datum zu erhalten
- Informationen über den Akkustand bei einem mobilen Gerät
- Orientierungsinformationen eines mobilen Geräts

Die Dokumentation und das Wissen über die Hardware-Schnittstellen eines Elements sind eine wichtige Grundlage für die Zusammenarbeit mit Hardwareexperten.

2.2.5.6.2 Form: User Interface-Vorlage

Ein User Interface (siehe Abschnitt 3.2) gehört zur wahrnehmbaren Form und ist typischerweise eine Visualisierung von Daten auf einem Bildschirm (z. B. Textfeld) und bietet Interaktionspunkte (z. B. Schaltflächen, Eingabefelder). Das User Interface ist wie folgt definiert:

User interface: An interface for the exchange of information between a user and a system.

Beachten Sie, dass die Gestaltung eines User Interface sowohl Form und Funktion (insbesondere die Struktur und Dynamik des Informationsaustauschs) als auch Qualität (insbesondere Benutzbarkeit und User Experience) umfasst.

Hinweise zur Erstellung der Beschreibung

Die User Interface-Vorlage besteht aus einer Visualisierung des User Interface und einer textuellen Beschreibung. Abbildung 28 zeigt eine beispielhafte Vorlage für ein User Interface aus der YPRC-Fallstudie.

In einem Softwaredesignkonzept ist es sehr nützlich, die verschiedenen Elemente/Bildschirme des User Interface mit individuellen Vorlagen zu beschreiben. Dadurch kann jedes User Interface auf die erforderlichen Daten, Schnittstellen, Funktionen und Use Cases verweisen, die Teil eines speziellen User Interface sind.

Auf Foundation Level Niveau reicht dieses Grundverständnis aus, um typische Softwareelemente wie Apps und Unternehmenssoftware zu gestalten. Anspruchsvollere Technologien - beispielsweise Augmented-Reality-Schnittstellen (siehe Abschnitt 3.2) - erfordern möglicherweise andere oder zusätzliche Bausteine.

Hinweise für die Dokumentation von Beziehungen

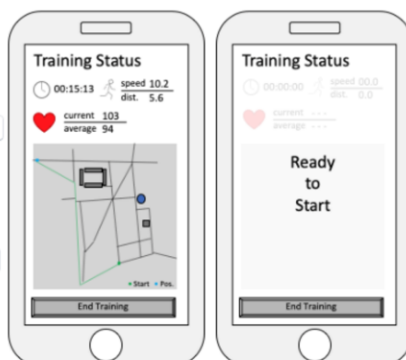
Das User Interface ist ein zentraler Bestandteil jeden Elements. Diese Rolle spiegelt sich auch in den verschiedenen Beziehungen wider. Beachten Sie die verschiedenen Beziehungen, die in der Beschreibung von Abbildung 28 erwähnt werden und sich auf zugehörige Vorlagen beziehen.

UI-5 Training Status Screen

Created by Kim Lauenroth
Last updated less than a minute ago • 1 min read

Relationships:

- E-4 Training Status
- E-5 Training Unit
- E-6 Track Data Point
- G-6 Inform the runner about current heart rate and performance with the app
- PHI-4 Smartphone Touch Screen
- SI-4 Map Data Interface (App)
- UC-2 The runner starts a training session with the watch
- UC-6 The runner initiates a solo training session on the app
- UC-11 The runner initiates a coached training session on the app



Description:

The training status screen presents the current training status to the runner during a training session (UC-6, UC-11). While the app is waiting for the runner to start the session with the watch (UC-2), the app shows an overlay message "ready to start" (right screen).

During an active training session, the screen shows the following information:

- Duration of the training (E-5.1)/current heart rate (E-4.2)/average heart rate (E-5.4)
- The current speed (E-4.3) and the running distance (E-5.5)
- A map with the start, the route and the current position (E-6)

The button "End Training" ends the training session (UC-6, UC-11).

Abkürzungen (nicht Teil der Vorlage)

E-x,y	Entität mit optionalem Verweis auf das Attribut y
G-x	Ziel (Goal)
PHI-x	Wahrnehmbare Hardware-Schnittstelle
SI-x	Software-Schnittstelle
U-x	Benutzer (User)
UC-x	Use Case

Abbildung 28 – Eine User Interface-Vorlage

Wir empfehlen, die folgenden Beziehungen für User Interfaces zu dokumentieren:

- Der Use Case oder die Use Cases, in denen die Schnittstelle verwendet wird, oder die Use Cases, die durch das User Interface ausgelöst werden können
- Die Funktionen, Hardware- oder Software-Schnittstellen, die Daten für das User Interface bereitstellen oder die Daten des User Interface verwenden
- Die Entitäten, die die Daten bereitstellen, die auf dem User Interface angezeigt und verarbeitet werden

Außerdem ist es sinnvoll, den Bezug zu einem bestimmten Ziel zu dokumentieren, das durch das User Interface erreicht wird.

Praxistipp

Diese Vielfalt an Beziehungen zwischen einem User Interface und anderen Bausteinen eines Designkonzepts kann für Einsteiger recht kompliziert erscheinen. Diese Beziehungen zu verstehen und zu pflegen ist jedoch ein zentrales Instrument, um selbst die komplizierteste digitale Lösung zu bewältigen.

Ein weiterer wichtiger Aspekt bei der Beschreibung von User Interfaces ist die visuelle Gestaltung des User Interface, die zusätzliche gestalterische Fähigkeiten erfordert (siehe Abschnitt 2.1.4). In den Bereichen Interaktionsdesign und visuelles Design ist es üblich, ein Designsystem zu definieren, um Designmaterialien zu dokumentieren und zu organisieren und um Richtlinien, Style Guides und „Leitplanken“ für das Design zu erstellen. Atomic Design [Fros2020] ist eine beispielhafte Technologie zur Erstellung eines Designsystems für User Interfaces. Dennoch betrachten wir Designsysteme als ein Thema für Fortgeschrittene. Leser, die an diesem Thema interessiert sind, können mit [Fros2020] als Einführung in Designsysteme beginnen.

2.2.5.6.3 Form: Software-Schnittstellenvorlage

Neben den User Interfaces sind Software-Schnittstellen ein wichtiger Baustein der zugrundeliegenden Form, um die Verbindung eines Softwareelements mit anderen Elementen oder Systemen zu beschreiben:

Software interface: An interface between a software element of a system and an element of the same system or of another system.

Diese Verbindung kann genutzt werden, um Daten aus einem bestehenden System zu beziehen (z. B. Kartendaten von einem Kartenserver) oder um externe Funktionen einzubinden, die von einem bestehenden System bereitgestellt werden (z. B. eine Zahlung von einem Zahlungsanbieter). Wird eine Software-Schnittstelle zur Verbindung mit einem anderen Element verwendet, das als Teil der digitalen Lösung realisiert wird, muss aus Gründen der Konsistenz eine entsprechende Schnittstelle im Elementdesignkonzept des anderen Elements definiert werden.

Hinweise zur Erstellung der Beschreibung

Auf Foundation Level Niveau ist eine textuelle Beschreibung der Software-Schnittstellen ausreichend. Abbildung 29 zeigt eine beispielhafte Software-Schnittstellenvorlage aus der YPRC-Fallstudie. In dieser Beschreibung wird der Zweck der Schnittstelle genannt und wie die Software-Schnittstelle mit anderen Teilen des digitalen Systems (hier die Smartwatch des Läufers) verbunden ist. Sehen Sie sich das Gerätedesignkonzept der Smartwatch für die entsprechende

Software-Schnittstelle auf der Smartwatch an, die den Status an die App sendet. In der YPRC-Fallstudie haben wir die Schnittstellen *Gesundheitsdaten an die App übertragen* für die Smartwatch des Läufers und *Gesundheitsdaten von der Smartwatch empfangen* in der App des Läufers als zwei entsprechende Bausteine definiert. Auf den ersten Blick mag dies wie eine überflüssige Information erscheinen. In der Praxis ist diese Beschreibung jedoch notwendig, da die Realisierung dieser Verbindung Implementierungsarbeit in beiden Elementen erfordert. Außerdem unterstützt diese explizite Beschreibung die Definition einer klaren Schnittstelle zwischen beiden Elementen.

SI-7 Watch-to-App Status Interface



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- **Con-2** Encrypt training data transfer
- **G-6** Inform the runner about current heart rate and performance with the app
- **UC-1** The runner connects the watch with the smartphone
- **UC-2** The runner starts a training session with the watch
- **UC-6** The runner initiates a solo training session on the app
- **UHI-7** Watch Bluetooth Interface
- **UHI-8** Smartphone Bluetooth Interface

Description:

The watch-to-app status interface is used to send status information from the watch to the app (UC-2). The watch signals the app that the runner has started his training. The watch sends the status "runner active" to the watch to start the data recording in the app (UC-6). When the runner ends his training or decides to make a break, the watch sends the status "runner inactive" to the app.

The connection between watch and app for this interface uses the Bluetooth module of the watch (UHI-7) and the Bluetooth module of the runner's smartphone (UHI-8). Establishing a Bluetooth connection is described in UC-1.

Abkürzungen (nicht Teil der Vorlage)

Con-x	Randbedingung (Constraint)
G-x	Ziel (Goal)
UHI-x	Zugrundeliegende Hardware-Schnittstelle
UC-x	Use Case

Abbildung 29 – Eine Software-Schnittstellenvorlage

Hinweise für die Dokumentation von Beziehungen

Ähnlich wie bei den User Interfaces kann die Software-Schnittstelle verschiedene Beziehungen zu anderen Bausteinen des Elementdesignkonzepts haben. Wir empfehlen, diese Beziehungen sorgfältig zu dokumentieren, um die Handhabung komplizierter digitaler Lösungen zu erleichtern.

Praxistipp

Die sorgfältige Dokumentation von Software-Schnittstellen ist Einsteigern im Digital Design meist fremd, da Software-Schnittstellen eher technischer Natur sind und als Konstruktionsaufgabe verstanden werden können. Nichtsdestotrotz muss ein DDP Software-Schnittstellen berücksichtigen, da Digital Design auch das Design der zugrundeliegenden Form umfasst, und definieren muss, welche Software-Schnittstellen ein Element benötigt, um seinen gewünschten Zweck zu erfüllen.

Auch Software-Schnittstellen sind ein wichtiger Bestandteil der gemeinsamen Arbeit mit Realisierungsexperten. Ein tiefgreifendes Verständnis der bestehenden Software-Schnittstellen

ermöglicht es einem DDP, die Experten für die Software-Realisierung zu verstehen und mit ihnen zu kommunizieren.

2.2.5.6.4 Form: Entitätsvorlage

Das letzte Element der zugrundeliegenden Form für den digitalen Teil ist die Entität. Sie ist wie folgt definiert:

Entity: A distinguishable structure of data stored by an element of a digital solution.

Der Begriff Entität stammt aus der Informatik und bezieht sich auf Datenstrukturen, die gespeichert und manipuliert werden.

Typische Beispiele für Entitäten in einem digitalen System sind:

- Benutzerdaten: Daten, die einen Benutzer identifizieren (Name, Passwort, Adresse)
- Statusdaten: Daten, die den Status des Elements beschreiben
- Transaktionsdaten: Daten, die eine bestimmte Transaktion beschreiben, die von der digitalen Lösung durchgeführt wurde (z. B. der Kauf eines Buchs)

Die explizite Dokumentation von Entitäten ist aus den folgenden Gründen wichtig:

- 1) Zur Dokumentation der gespeicherten Daten (die Tabelle in Abbildung 30 charakterisiert die Entität)
- 2) Zur Schaffung eines Bezugspunkts für die Beschreibung von User Interfaces (siehe Beziehung zu UI-5 in Abbildung 30)
- 3) Zur Schaffung eines Bezugspunkts für die Beschreibung von Funktionen

Hinweise zur Erstellung der Beschreibung

Es gibt verschiedene Ansätze zur Dokumentation von Datenstrukturen (z. B. UML-Klassendiagramme, Entity-Relationship-Diagramme). Fachleute werden die Integration dieser Techniken in das vorgestellte Softwaredesignkonzept verstehen. Auf Foundation Level Niveau reicht es aus, die einzelnen Entitäten und die Informationsattribute, die eine Entität definieren, anhand von Tabellen zu identifizieren und zu beschreiben. Abbildung 30 zeigt ein Beispiel für eine Entitätsvorlage aus der YPRC-Fallstudie.

Hinweise für die Dokumentation von Beziehungen

Die Beziehungen einer Entität zu anderen Bausteinen werden normalerweise nicht aus der Beschreibung der Entität abgeleitet. Stattdessen müssen die Beziehungen zur Beschreibung der verbundenen Bausteine hinzugefügt werden. Wenn zum Beispiel ein User Interface durch die Verwendung von Daten einer Entität beschrieben wird, dann sollte dieses User Interface in die Liste der Referenzen aufgenommen werden.

E-4 Training Status

KL Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- **Con-3** Encrypt training data storage
- **UC-11** The runner initiates a coached training session on the app
- **UC-6** The runner initiates a solo training session on the app
- **UI-5** Training Status Screen

Description:

This entity represents the status of the current training session and consists of the following attributes:

ID	Attribute	Description
1	Runner status	Indicates whether the runner is active solo, active coached, or inactive
2	Heart rate	Heart rate in beats per minute
3	Current speed	Speed in kilometers per hour
4	GPS position	Current GPS position of smart phone

Abkürzungen (nicht Teil der Vorlage)

Con-x	Randbedingung (Constraint)
UC-x	Use Case
UI-x	User Interface (Benutzerschnittstelle)

Abbildung 30 - Eine Entitätsvorlage

Praxistipp

Diese indirekte Art der Dokumentation von Beziehungen zu einer Entität erscheint auf den ersten Blick unpraktisch. Unserer Erfahrung nach ist dieser Ansatz jedoch sehr pragmatisch und ermöglicht eine effiziente Dokumentation der Beziehungen.

2.2.5.6.5 Form: Eine Vorlage für physische Teile

Der physische Teil ist ein Baustein im wörtlichen Sinne und wird für die Konstruktion von Geräten benötigt:

Physical part: The part that makes up a device.

Ein physisches Teil kann zu der wahrnehmbaren oder zugrundeliegenden Form eines Geräts gehören. Es kann technische Funktionalität bieten (z. B. eine CPU oder eine Batterie) oder Teil der eigentlichen Form des Geräts sein (z. B. das Gehäuse der Smartwatch in der YPRC-Fallstudie). Abbildung 31 zeigt eine beispielhafte Vorlage für ein physisches Teil.

Hinweise zur Erstellung der Beschreibung

Die Beschreibung gibt einen kurzen Überblick darüber, worum es sich bei dem physischen Teil handelt. In der Beschreibung eines physischen Teils wird häufig auf andere physische Teile und Hardware-Schnittstellen verwiesen, die von dem physischen Teil bereitgestellt werden. Diese Schnittstellen ermöglichen es dem Gerät, mit seiner Umgebung und dem Benutzer zu interagieren.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element aus dem Designkonzept, das in der Beschreibung erwähnt wird und einen Bezug zu dem beschriebenen physischen Teil hat, wird erwähnt.

PP-2 Watch Hardware



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- [PP-1 Watch Housing](#)
- [UHI-6 Watch Hardware Clock Interface](#)
- [UHI-7 Watch Bluetooth Interface](#)

Description:

The watch hardware is part of the watch housing (PP-1) and is a system-on-chip device with an ARM 1.0 Ghz CPU, 1024kb memory, and 4096kb flash storage for the operating system. The chip further provides a hardware clock module (UHI-6) and a Bluetooth module (UHI-7).

Abkürzungen (nicht Teil der Vorlage)

UHI-x Zugrundeliegende Hardware-Schnittstelle

Abbildung 31 - Eine Vorlage für physische Teile

Praxistipp

Bei der Beschreibung von physischen Bauteilen sollten Sie bedenken, dass die Vorlage des physischen Bauteils nur stellvertretend für einen ganzen Hardware-Entwicklungsprozess steht (siehe auch Abschnitt 2.2.2.5).

2.2.5.6.6 Funktion: Funktionsvorlage

Auf der Elementebene kann eine Funktion mit zwei Bausteinen beschrieben werden: Der Funktionsvorlage und der Use Case Vorlage.

Die Funktionsvorlage befasst sich mit der zugrundeliegenden Funktion eines Elements:

**Underlying function: The description of the transformation
of certain inputs into certain outputs.**

Wie die anderen Definitionen ist auch die Definition der zugrundeliegenden Funktion eher abstrakt, um ein breites Spektrum an möglichen Funktionen abzudecken. In der täglichen Arbeit wird der Begriff „zugrundeliegende Funktion“ meist synonym mit dem Begriff Funktion verwendet. In diesem Handbuch haben wir das Gleiche getan, um die Lesbarkeit zu verbessern. Wenn der Unterschied wichtig ist, verwenden wir den vollständigen Begriff „zugrundeliegende Funktion“.

Hinweise zur Erstellung der Beschreibung

Es gibt verschiedene Ansätze zur Beschreibung der zugrundeliegenden Funktionen eines digitalen Systems (z. B. UML-Aktivitätsdiagramme, UML-Zustandsmaschinen, algebraische Spezifikationen). Abbildung 32 zeigt eine beispielhafte Funktionsvorlage zur Beschreibung einer zugrundeliegenden Funktion aus der YPRC-Fallstudie, die einen Detaillierungsgrad aufweist, der für den DDP auf Foundation Level Niveau ausreichend ist.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element des Designkonzepts, das in der Beschreibung erwähnt wird und einen Bezug zu der beschriebenen zugrundeliegenden Funktion hat, wird erwähnt.

F-5 Calculate Heart Rate Status Parameter



Created by Kim Lauenroth
Last updated Jul 22, 2020 • 1 min read

Relationships:

- [E-3 User Data](#)
- [UI-2 Training Display](#)

Description:

This function calculates the parameter for the heart rate status display of the watch (see UI-2) for green, yellow, and red heart rate status. The function uses the date of birth (E-3.5), the height (E-3.6), and the weight (E-3.7) of the runner stored in the app.

Abkürzungen (nicht Teil der Vorlage)

E-x.y	Entität mit optionalem Verweis auf das Attribut y
UC-x	Use Case
UI-x	User Interface (Benutzerschnittstelle)

Abbildung 32 - Eine Funktionsvorlage

Praxistipp

Beachten Sie, dass der Text nicht die konkrete Formel für die Berechnung beschreibt. Diese Formel wird weggelassen, weil sie für das Verständnis des Beispiels nicht notwendig ist. Es genügt zu wissen, dass eine solche Formel existiert. Sie wird während der Realisierung der App hinzugefügt. Dieses Beispiel soll verdeutlichen, dass ein Designkonzept nicht bereits in einem frühen Stadium vollständig sein muss, sondern später ergänzt werden kann.

2.2.5.6.7 Funktion: Use Case Vorlage

Wir haben die Vorlage für den Use Case in unserer Erläuterung ans Ende gestellt, weil sie auf der Elementebene zwei wichtige Zwecke erfüllt. Erstens beschreibt der Use Case die Interaktion zwischen dem Benutzer und dem Element. Zweitens⁸ fasst der Use Case alle anderen Elemente der Elementebene zu einem kohärenten Interaktionsablauf zusammen.

Use case: A set of possible interactions between a user and an element of a system that provide a benefit for the user(s) involved.

⁸ Der zweite Aspekt ist ein erweitertes Verständnis der traditionellen Use Cases. Use Cases konzentrieren sich in der Regel nur auf die Benutzerperspektive. Im Rahmen der ganzheitlichen Perspektive des Digital Designs bevorzugen wir ein breiteres Verständnis von Use Cases, das andere Bausteine einbezieht und ausdrücklich auf diese verweist.

UC-6 The runner initiates a solo training session on the app



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

- [E-3 User Data](#)
- [E-4 Training Status](#)
- [F-2 Record training data and average values during solo training](#)
- [F-5 Calculate heart rate status parameter](#)
- [SI-6 App-to-Watch Status Interface](#)
- [SI-7 Watch-to-App Status Interface](#)
- [SI-9 Transfer Training Data Interface](#)
- [U-1 Runner](#)
- [UC-2 The runner starts a training session with the watch](#)
- [UHI-4 Smartphone Clock Interface](#)
- [UI-4 Training App Start Screen](#)
- [UI-5 Training Status Screen](#)
- [UI-6 Training Unit Summary Screen](#)

Main Scenario:

1. The runner (U-1) is viewing the start screen (UI-4).
2. The watch is connected (E-3.1) and the "Start Training" button is active.
3. The runner (U-1) selects the button "Start Training" (UI-4).
4. The app calculates the heart rate status parameter (F-5) and sends the parameter, the current data and time (UHI-4), and the training status "active" to the watch (SI-6).
5. The app shows the training status screen (UI-5) with the message "ready to start"
6. The app receives the signal "runner active" from the watch (see SI-7) and stores the status "active solo" in E-4.1. (see UC-2 of the runner's watch for details).
7. The app records the training data from the watch (F-2) and shows the training status on the screen (UI-5).
8. The runner (U-1) performs his training session. After a while, the app receives the status "runner inactive" from the watch (SI-7) and stores this status in E-4.1.
9. The app stops the recording of the training data (F-2).
10. The runner (U-1) decides to end the training and selects the "End Training" Button (UI-5).
11. The app sends the training status "inactive" to the watch (SI-6).
12. The app shows the training summary (UI-6).
13. The app transfers the training data to the portal (S-9).

Alternative Scenarios:

- 10a) The runner (U-1) decides to continue the training and the app receives a "runner active" signal from the watch. The status is stored in E-4.1.
- 11a) Continue with step 7) from main scenario.

Abkürzungen (nicht Teil der Vorlage)

E-x.y	Entität mit optionalem Verweis auf das Attribut y
F-x	Funktion
SI-x	Software-Schnittstelle
U-x	Benutzer (User)
UC-x	Use Case
UHI-x	Zugrundeliegende Hardware-Schnittstelle
UI-x	User Interface (Benutzerschnittstelle)

Abbildung 33 - Eine Vorlage für einen Use Case

Hinweise zur Erstellung der Beschreibung

Abbildung 33 zeigt einen Auszug aus einem Use Case der YPRC-Fallstudie. Die Beschreibung besteht aus einem Hauptszenario und einem Alternativszenario. Diese schrittweise Beschreibung in Textform ist für Use Cases üblich. Fachleute kennen weitere Dokumentationstechniken im Zusammenhang mit Use Cases (z. B. UML-Aktivitätsdiagramm, UML-Sequenzdiagramme oder die Beschreibung von Ausnahmeszenarien). Diese Techniken werden als fortgeschritten angesehen.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element des Designkonzepts, das in der Beschreibung erwähnt wird und einen Bezug zu dem beschriebenen Use Case hat, wird erwähnt. Use Cases haben in der Regel mehrere Beziehungen, da sie mehrere Aspekte einer digitalen Lösung integrieren. In Abschnitt 2.2.6.1 wird die Bedeutung von Use Cases und deren Beziehungen näher erläutert.

Praxistipp

Auf Foundation Level Niveau ist es wichtig, die schrittweise Beschreibung und die Bedeutung der Verweise auf andere Elemente des Softwaredesignkonzepts zu verstehen. Beachten Sie, dass fast jeder Schritt des Beispiels Verweise auf andere Elemente enthält (andere Use Cases, Entitäten, Funktionen, User Interfaces usw.).

Durch diese Art des Verweises auf andere Elemente werden die Use Case Vorlagen zur leistungsfähigsten Vorlage in Bezug auf Ausdruckskraft und Traceability. Die Use Case Vorlagen stellen alle anderen Elemente in einen situativen Kontext und beschreiben, wie und wann ein bestimmter Teil des Konzepts wichtig ist.

Mit anderen Worten: Die Use Case Vorlage ist zusammen mit der Funktionsvorlage der Teil des Softwaredesignkonzepts, in dem der DDP das digitale System buchstäblich in Textform programmiert.

Obwohl die Beschreibung langweilig und einfach erscheinen mag, ist es ziemlich schwierig, gute Use Cases zu erstellen, selbst auf dieser einfachen Ebene. Die YPRC-Fallstudie enthält mehrere Beispiele, die das Erlernen des Schreibens von Use Cases in dem vorgestellten Stil unterstützen.

2.2.5.7 Qualität: Qualitätsanforderungsvorlage

Qualität ist ein wichtiger Aspekt sowohl auf der Systemebene als auch auf der Elementebene. Die Beschreibung von Qualität ist ebenso anspruchsvoll wie die Beschreibung von Zielen. Wir empfehlen, für jedes einzelne Qualitätsmerkmal eine eigene Qualitätsanforderung zu schreiben.

Jede Qualitätsanforderung wird durch eine separate Vorlage dokumentiert. Abbildung 34 - Eine Vorlage für Qualitätsanforderungen zeigt eine beispielhafte Vorlage für Qualitätsanforderungen mit Inhalten aus der YPRC-Fallstudie. Die Überschrift enthält die Kennung (QR für Quality Requirement (Qualitätsanforderung)), die Nummer des Elements (hier 2) und den Titel. Der Titel einer Qualitätsanforderung sollte eine Zusammenfassung in einem Satz sein. Weitere Details können in der Beschreibung folgen.

Hinweise zur Erstellung der Beschreibung

Die Beschreibung von Qualitätsanforderungen ist ein wichtiges Thema im Requirements Engineering (vgl. [CPRE2020]). Einsteigern im Digital Design empfehlen wir, die Qualität des digitalen Systems mit textuellen Qualitätsanforderungen zu beschreiben. Wer mit dem Verfassen von Qualitätsanforderungen beginnt, sollte die folgenden Regeln beachten:

- 1) Verweisen Sie auf das Element des digitalen Systems, das die beschriebene Qualität aufweisen soll.
- 2) Vermeiden Sie Qualitätsanforderungen, die sich auf das gesamte digitale System beziehen.
- 3) Beschreiben Sie die Qualität wenn möglich in quantifizierbarer Form.

Hinweise für die Dokumentation von Beziehungen

Die Liste der Beziehungen kann direkt aus der Beschreibung abgeleitet werden. Jedes Element des Designkonzepts, das in der Beschreibung erwähnt wird und einen Bezug zu der beschriebenen Qualitätsanforderung hat, wird erwähnt.

Praxistipp



In Abschnitt 2.1.3 haben wir die Bedeutung von Qualität bei der Entwicklung digitaler Lösungen ausführlich erörtert. Die Qualitätsanforderungsvorlage ist ein wichtiges Werkzeug für Einsteiger im Digital Design, um die gewünschten Qualitätsmerkmale zu dokumentieren, die ein Element oder das gesamte System haben soll.

QR-2 Latency of heart rate status change and feedback to the runner



Created by Kim Lauenroth
Last updated just a moment ago • 1 min read

Relationships:

-  F-1 Update heart rate status
-  U-1 Runner

Description:

The latency time between a change of the heart rate status (F-1) of the runner (U-1) and a feedback to the runner (vibration alert and display on watch screen) must be below 1 second.

Abkürzungen (nicht Teil der Vorlage)

F-x	Funktion
U-x	Benutzer (User)

Abbildung 34 - Eine Vorlage für Qualitätsanforderungen

2.2.5.8 Übersichtsbilder zum Zweck der Visualisierung

Einsteigern im Digital Design empfehlen wir, ein Übersichtsbild zu erstellen, das die Gesamtform des Systems oder eines Elements darstellt. Abbildung 35 zeigt das Übersichtsbild der YPRC-Fallstudie. Sie verwendet eine sehr einfache Notation, die aus drei Elementen besteht:

- Benutzersymbol zur Darstellung des Benutzers
- Kästchen, die andere Elemente der Form darstellen; die unterschiedlichen Farben beziehen sich auf die vorhandenen und die zu gestaltenden Elemente
- Pfeile zur Darstellung des Datenflusses/der Interaktion zwischen den Elementen

Obwohl diese Notation recht einfach ist, reicht sie aus, um einen guten Überblick über das digitale System von YPRC zu erhalten.

In den frühen Phasen des konzeptuellen Schritts kann das Übersichtsbild als wichtigstes Artefakt der Arbeit am Design des digitalen Systems verwendet werden. Die Interaktionspfeile sind ausreichend, um die Ideen der Funktion zu vermitteln. Darüber hinaus können verschiedene Alternativen des digitalen Systems anhand unterschiedlicher Übersichtsbilder erarbeitet werden.

In späteren Phasen des konzeptuellen Schritts wird das Übersichtsbild relativ stabil und kann als Karte dienen, die die Gesamtform des digitalen Systems zeigt. Um dieser Übersichtsaufgabe gerecht zu werden, muss das Übersichtsbild während des gesamten Lebenszyklus der digitalen Lösung ständig gepflegt werden. Der Tätigkeitsbereich Design ist dafür verantwortlich, dass das Übersichtsbild immer auf dem neuesten Stand ist.

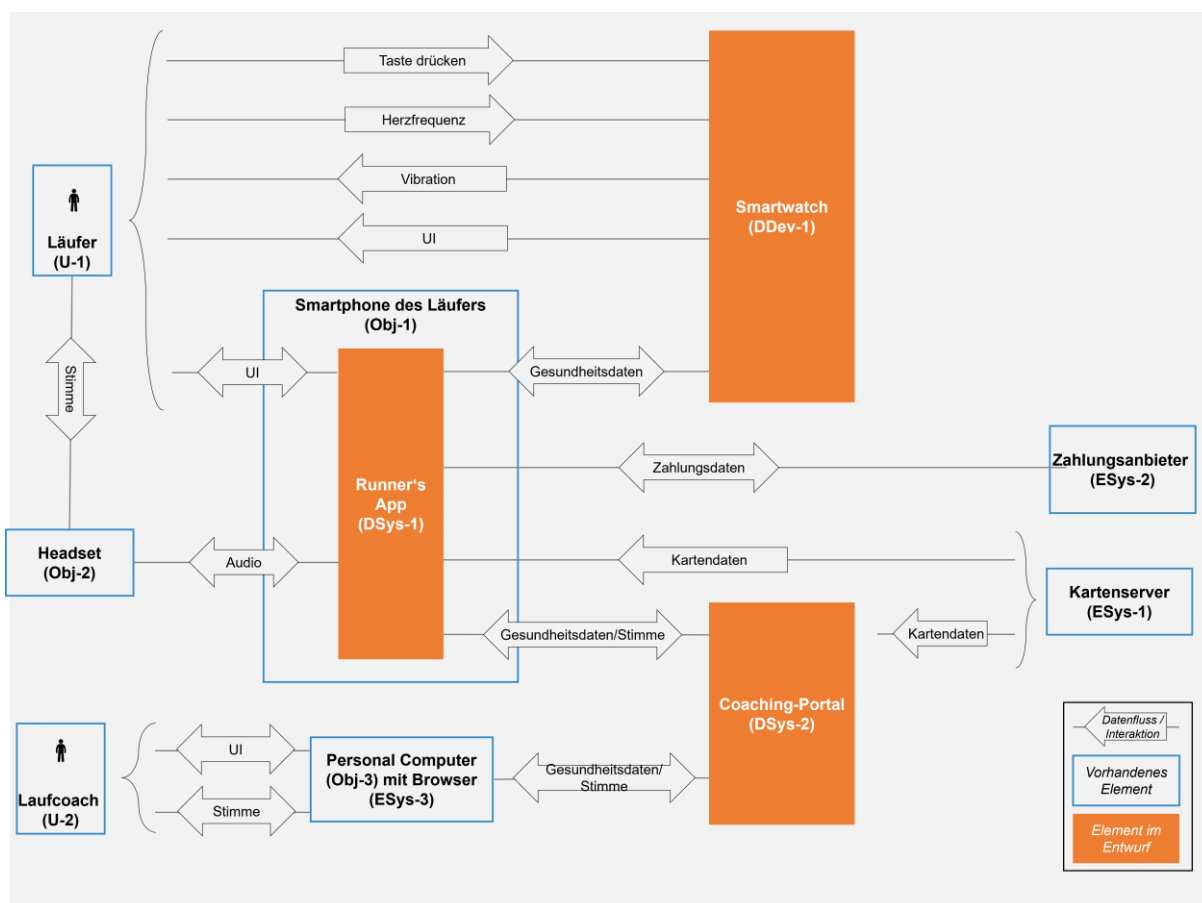


Abbildung 35 - Überblick über das digitale YPRC-System mit seinen Elementen

Fachleute kennen weitere Techniken zur Erstellung eines Übersichtsbildes (z. B. UML-Use-Case-Diagramme). Diese Techniken sind natürlich ebenfalls anwendbar. Wir betrachten diese Techniken jedoch als fortgeschritten.

2.2.6 Alles für das große Ganze wieder zusammenfügen

Bisher haben wir Dokumentvorlagen und Dokumentationstechniken für die Lösungsebene sowie die System- und Elementebene vorgestellt. Mit dieser Einführung haben wir eine Vielzahl von Details abgedeckt. Wir gehen davon aus, dass die erste Lektüre dieses Abschnitts, für manche Einsteiger in das Digital Design, möglicherweise etwas Verwirrung über all diese Details und Beziehungen zurücklässt.

In diesem letzten Abschnitt wollen wir alle Teile wieder zu einem großen Bild zusammenfügen. Wir beginnen mit der Elementebene und arbeiten uns dann zur Systemebene und zur Lösungsebene vor. Abschließend gehen wir noch einmal auf die Phasen und Tätigkeitsbereiche des Bauprozesses aus Kapitel 1 ein.

2.2.6.1 Beziehungen zwischen Bausteinen auf Elementebene

In Abschnitt 2.2.5 haben wir verschiedene Bausteinvorlagen für die Arbeit am Design eines bestimmten Elements einer digitalen Lösung vorgestellt. Abbildung 36 zeigt diese Bausteine und ihre Beziehungen zueinander, einschließlich der Bausteine auf der Systemebene. Die Beziehungen wurden bereits zusammen mit der Beschreibung der einzelnen Bausteine erörtert. Das Verständnis der Beziehungen zwischen den Bausteinen ist jedoch der Schlüssel zur praktischen Anwendung. Deshalb fassen wir sie in diesem Abschnitt noch einmal zusammen.

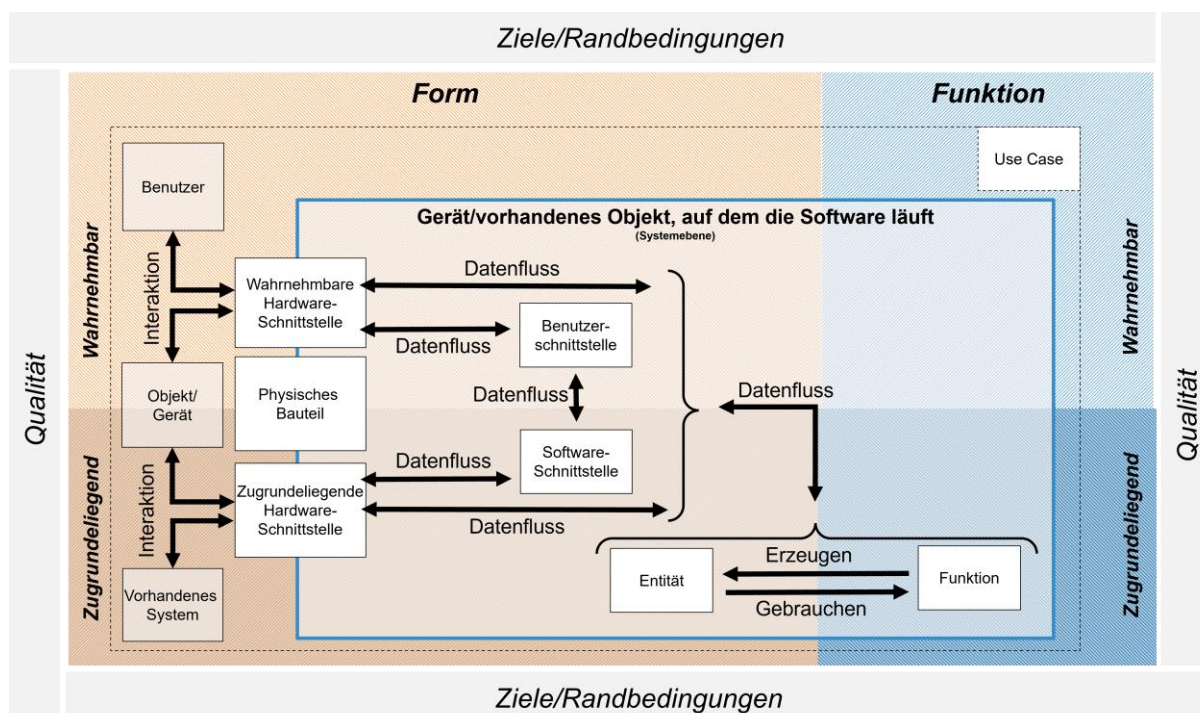


Abbildung 36 - Bausteine eines Software-/Gerätedesignkonzepts

Für Einsteiger im Digital Design auf der Elementebene empfehlen wir, die in Abschnitt 2.2.5 vorgestellten Vorlagen als Bausteine im wörtlichen Sinne zu verstehen. Sie können wie virtuelle Legosteine verwendet werden, um ein bestimmtes Element aufzubauen. Wie bei echten Steinen passen nur bestimmte Arten von Steinen zusammen. Die Beziehungen legen fest, welche Bausteine kombiniert werden können. Wir beschreiben diese im Folgenden.

Jedes Element benötigt Schnittstellen, um mit seiner Umgebung zu interagieren

Wir beginnen mit dem Gerät, das die Software betreibt. Bei diesem Gerät kann es sich um ein bereits vorhandenes Objekt handeln (z. B. ein Smartphone oder ein Computer) oder um ein Gerät, das speziell für die digitale Lösung gebaut wird (z. B. die Smartwatch aus der YPRC-Fallstudie). Jedes Gerät muss Schnittstellen bieten, die eine Interaktion zwischen der Umgebung und dem Gerät ermöglichen. Zu diesem Zweck haben wir Vorlagen für wahrnehmbare oder zugrundeliegende Hardware-Schnittstellen definiert. Wenn Sie mit der Arbeit an einem neuen Elementdesignkonzept beginnen, empfehlen wir, zunächst die Schnittstellen zu definieren. Typische Kandidaten, die die meisten Geräte bieten, sind Displays, physische Tasten,

Netzwerkanschlüsse und Eingabegeräte wie physische Tasten, eine Tastatur oder ein Touchscreen (siehe Kapitel 3 für weitere Details).

Schnittstellen erzeugen Datenflüsse

Wenn wir die Schnittstellen aus der Sicht der Elemente betrachten, können sie als Anbieter und Konsumenten von Datenströmen verstanden werden. Der Touchscreen eines Smartphones ist ein gutes Beispiel dafür. Der Bildschirm beschreibt das visuelle Erscheinungsbild des User Interface und benötigt einen Datenfluss, der visualisiert wird. Ein User Interface erzeugt auch einen Datenfluss, der die Eingaben des Benutzers darstellt (z. B. das Berühren einer bestimmten Schaltfläche oder die Eingabe von Daten). Obwohl dies aus gestalterischer Sicht recht technisch klingt, ist es wichtig, diesen Datenfluss auf einer angemessenen Abstraktionsebene zu definieren. In dieser Situation ist die Annahme einer perfekten Technologie sinnvoll. Aus gestalterischer Sicht müssen wir uns keine Gedanken darüber machen, wie der Bildschirm im Detail funktioniert, um den Datenfluss zu erzeugen oder zu nutzen. Wir müssen nur die Eigenschaften festlegen, die der Bildschirm aus gestalterischer Sicht haben muss (z. B. die Auflösung, die Farbtiefe usw.).

User Interfaces als Spezialfall von Schnittstellen

Da das User Interface ein besonderer Aspekt jeder digitalen Lösung ist, haben wir einen eigenen Baustein für diesen Zweck definiert. Mit diesem Baustein beschreiben wir die konkrete visuelle Form des Bildschirms. Außerdem ist für die visuelle Darstellung des User Interface für den Benutzer eine spezielle Hardwareschnittstelle erforderlich.

User Interfaces schaffen auch einen Datenfluss

Ein User Interface besteht aus einer Art statischer Form (z. B. Text, Bilder) und einer Form, die Interaktion ermöglicht (z. B. Eingabefelder, Textfelder, die bestimmte Daten anzeigen). Auch hier haben wir einen Datenfluss zwischen weiteren Bausteinen, nämlich Entitäten und Funktionen. Entitäten speichern Daten, die auf einem User Interface visualisiert werden können, und Funktionen können Daten erzeugen, die visualisiert werden können. Dies ist ein Datenfluss von Entitäten und Funktionen zum User Interface. Der Datenfluss ist auch in umgekehrter Richtung möglich. Ein Benutzer kann bestimmte Daten eingeben, die in einer Entität gespeichert werden oder einer Funktion für weitere Berechnungen zur Verfügung gestellt werden.

Entitäten und Funktionen sind der Kern jedes Elements eines digitalen Systems

Wir haben uns nun von der Hardware-Schnittstelle über das User Interface bis hin zu den Funktionen und Entitäten vorgearbeitet. Beide Bausteine bilden das Rückgrat für die Beschreibung dessen, was ein Element tatsächlich tun kann. Entitäten speichern die Daten, die Funktionen erzeugen und verbrauchen. Daher geht die Beschreibung von Funktionen und Entitäten beim Design eines Elements einer digitalen Lösung Hand in Hand.

Software-Schnittstellen sind das Gegenstück zu User Interfaces auf Ebene der zugrundeliegenden Form

Bislang haben wir das User Interface besprochen. Da Interaktion und Konnektivität zentrale Merkmale der Digitalisierung sind, sind Software-Schnittstellen als Teil der zugrundeliegenden Form von gleicher Bedeutung. Mit Software-Schnittstellen beschreiben wir technische Verbindungen zwischen dem eigentlichen Element und anderen Elementen der digitalen Lösung (z. B. Interaktion mit einem Zahlungsanbieter).

Wie User Interfaces benötigen auch Software-Schnittstellen eine Hardware-Schnittstelle, über die die Software-Schnittstelle mit anderen Elementen der digitalen Lösung verbunden werden kann. Im einfachsten Fall kann diese Hardware-Schnittstelle eine Art Internetverbindung sein. Die Dokumentation der Internetverbindung mag überflüssig als eine Art „No-Brainer“ erscheinen. Da es sich jedoch um eine Kernfunktion handelt, empfehlen wir Ihnen, sie nicht wegzulassen.

Der Datenfluss von Software-Schnittstellen bezieht sich auf Entitäten und Funktionen in der gleichen Weise wie bei User Interfaces.

Qualitätsanforderungen definieren qualitative Details

Wie bereits in Abschnitt 2.1 erläutert, ist Qualität ein übergreifender Aspekt jeder digitalen Lösung. Daher kann jeder Baustein eines Elements mit zusätzlichen Qualitätsanforderungen detailliert werden.

Use Cases für das Zusammenfügen von Bausteinen zu einem Gerüst

Bislang haben wir Bausteine beschrieben, die statische Aspekte beschreiben. Der Use Case ist eine andere Art von Baustein. Er definiert den dynamischen Rahmen, der die anderen Bausteine eines Elements durch Hauptszenarien und Alternativszenarien in Aktion setzt.

Abbildung 37 veranschaulicht dieses Zusammenspiel anhand eines Beispiels aus der YPRC-Runner-App: die Erstellung eines neuen Benutzerkontos.

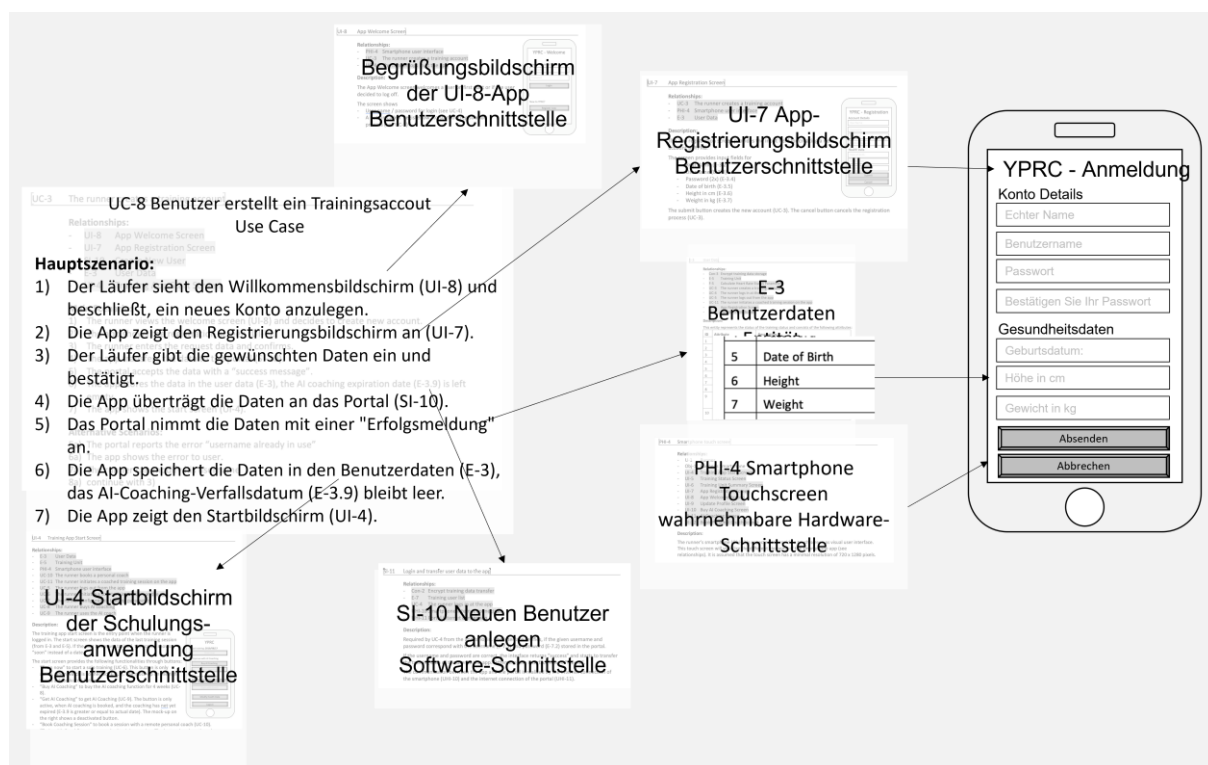


Abbildung 37 - Beispiel für einen Use Case, der andere Elemente einschließt

Die linke Seite dieser Abbildung zeigt das Hauptszenario des Use Case UC-8. Die Elemente, auf die verwiesen wird, werden als Icons dargestellt. Der Use Case beginnt mit der Darstellung des Begrüßungsbildschirms. Der Verweis zeigt auf UI-8, den Baustein des User Interface, der die Form des Begrüßungsbildschirms zeigt. Um ein neues Konto zu erstellen, muss der Benutzer auf die Schaltfläche *Jetzt registrieren* klicken. Diese Informationen sind in UI-8 enthalten und müssen

nicht unbedingt im Use Case erwähnt werden. Im nächsten Schritt zeigt die App den Registrierungsbildschirm UI-7 an.

Der Layout-Prototyp von UI-7 ist auf der rechten Seite der Abbildung als Beispiel für ein User Interface dargestellt. Hier können wir weitere Beziehungen sehen, die wir oben beschrieben haben: UI-7 hat eine Beziehung zu E-3, der Entität, die die in der App gespeicherten Benutzerdaten beschreibt. Die Abbildung zeigt einen Auszug aus der Tabelle mit drei Attributen: Geburtsdatum, Größe und Gewicht. Diese drei Attribute sind auch auf dem UI-Bildschirm vorhanden, da der Benutzer sie eingeben muss. Außerdem hat UI-7 einen Verweis auf die Hardware-Schnittstelle PHI-4, die den Touchscreen des Smartphones darstellt.

Wenn der Benutzer die Daten eingibt und die Eingabe bestätigt (Schritt 3 von UC-8), überträgt die App die Daten über eine Software-Schnittstelle (SI-10) an das Portal, um ein neues Benutzerkonto anzulegen. Wenn diese Aktion erfolgreich ist, werden die Benutzerdaten in der App gespeichert (Verweis auf E-3), mit einer Ausnahme: Dem Ablaufdatum des Coachings (die Bedeutung dieses Datums ist in den Details der Fallstudie zu finden). Nachdem die Daten gespeichert wurden, sieht der Use Case vor, dass die App den Startbildschirm anzeigt (UI-4). Dies ist das Ende des Hauptszenarios.

Obwohl die Beschreibung des Use Case selbst sehr kurz und knapp ist, bietet der Use Case eine ausführliche Beschreibung des Registrierungsprozesses zusammen mit den Elementen, auf die verwiesen wird, und dient somit als Rahmen, der mehrere Elemente zu einer kohärenten Beschreibung des Elements zusammenfasst.

2.2.6.2 Beziehungen zwischen Bausteinen auf Systemebene

Auf der Systemebene wird dann die größere Perspektive in Form von Beziehungen geschaffen. Abbildung 38 - Bausteine eines Systemdesignkonzepts bietet einen Überblick über die Bausteine eines Systemdesignkonzepts. Um die Form des digitalen Systems zu beschreiben, beginnen wir mit dem Baustein *Benutzer*. Hier sehen wir wieder Elemente, die die Form des digitalen Systems bilden (Benutzer, Objekt, bestehendes System, Gerät und Softwareelement).

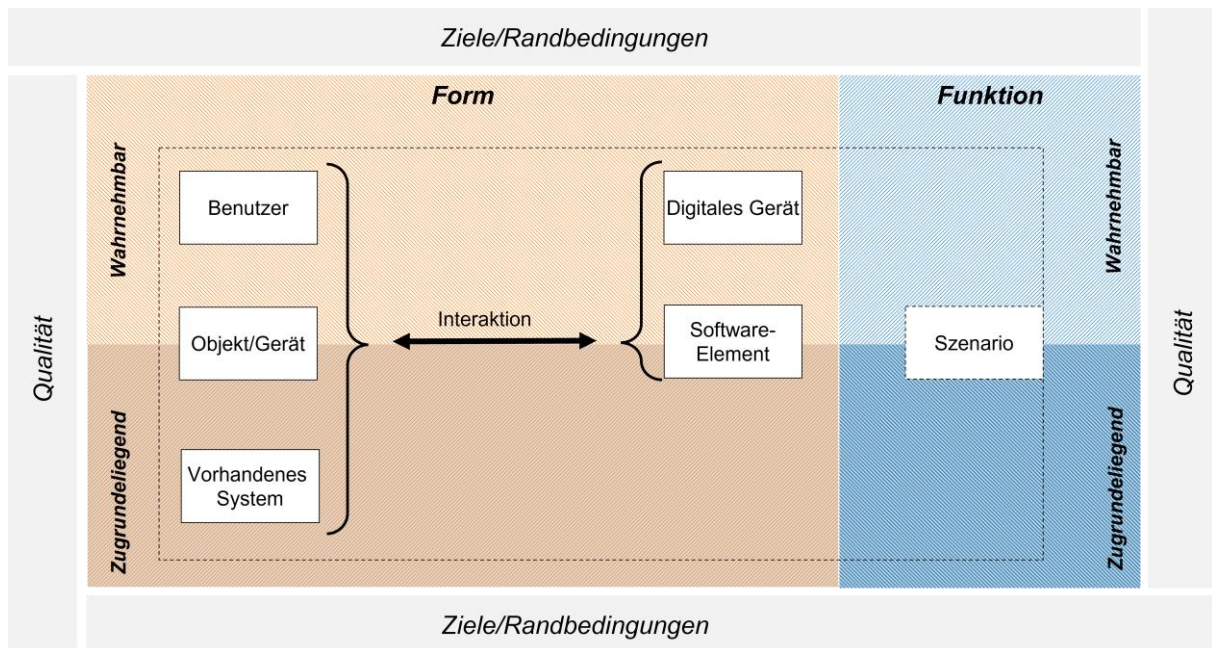


Abbildung 38 - Bausteine eines Systemdesignkonzepts

Diese Elemente stehen in Wechselwirkung zueinander. Um die Funktion des digitalen Gesamtsystems zu beschreiben, haben wir textuelle Szenarien gesehen, die eine beispielhafte Funktion des Systems beschreiben.

Die Beschreibung von Szenarien dient einem ähnlichen Zweck wie die Use Cases auf der Elementebene. Das Szenario ist jedoch nur ein beispielhafter Ablauf. Daher sind die Szenarien nur eine Teilbeschreibung der Funktion des gesamten Systems einschließlich des Benutzers und seiner Situation. Diese Teilbeschreibung ist der Zweck der Szenarien auf Systemebene und stellt keinen Nachteil dar, da die wichtigen Details durch die Use Cases auf der Elementebene abgedeckt werden.

2.2.6.3 Beziehung zwischen Elementebene und Systemebene

Die Beziehung zwischen Element- und Systemebene aus Sicht des Elements haben wir bereits in Abschnitt 2.2.6.1 erwähnt. Betrachtet man die Beziehungen zwischen den beiden Ebenen im Allgemeinen, so lassen sich zwei wichtige Erkenntnisse gewinnen.

Erstens kann eine Reihe von Bausteinen auf der Elementebene von der Systemebene abgeleitet werden. Die Beziehungen zwischen zwei Elementen auf der Systemebene führen zu zwei Schnittstellen auf der Elementebene. Darüber hinaus führt die durch ein Szenario auf der Systemebene beschriebene Funktion zu einem oder mehreren Use Cases und Funktionen für die verschiedenen Elemente auf der Elementebene.

Zweitens erfordert die Konsistenz zwischen der Elementebene und der Systemebene, dass die Details auf der Elementebene nicht im Widerspruch zur Beschreibung des Gesamtsystems stehen. Wenn ein Element eine Schnittstelle zu einem anderen Element definiert, muss diese Beziehung auch auf der Systemebene vorhanden sein, sonst besteht ein Widerspruch. Ebenso dürfen sich Ziele und Einschränkungen auf Element- und Systemebene nicht widersprechen.

Tabelle 9 gibt einen Überblick über die wichtigsten Abhängigkeiten zwischen der Systemebene und der Elementebene.

Tabelle 9 - Beziehungen zwischen Systemebene und Elementebene

Ebene	Form	Funktion	Qualität
System	Die Elemente des Systems, einschließlich der Beziehungen, beschreiben die statische Struktur.	Szenarien beschreiben die Funktion in beispielhafter Weise.	Qualitätsanforderungen, die das System erfüllen muss
Element	Jede Beziehung aus der Systemebene muss durch Schnittstellen erfasst werden. Alle Beziehungen auf der Elementebene müssen auf der Systemebene definiert sein.	Die beispielhafte Funktion aus der Systemebene muss durch die Use Cases abgedeckt sein. Die Use Cases dürfen nicht im Widerspruch zu den Szenarien auf der Systemebene stehen. Über die Szenarien auf Systemebene hinaus sind zusätzliche Verhaltensweisen erforderlich.	Die Qualitätsanforderungen für ein Element können von der Systemebene abgeleitet werden. Die für ein Element festgelegten Qualitätsanforderungen dürfen nicht im Widerspruch zu den Qualitätsanforderungen des Systems stehen. Zusätzliche Qualitätsanforderungen für ein Element sind möglich.

2.2.6.4 Beziehung zwischen Lösungsebene und System-/Elementebene

Betrachtet man die YPRC-Fallstudie, so ist der Unterschied zwischen den Designkonzepten auf der Lösungsebene und der System-/Elementebene erheblich. Auf der Lösungsebene geht es um Wertversprechen, Visionen und Geschäftsmodelle, während die System-/Elementebene Funktionen, Schnittstellen, Entitäten und Qualitätsanforderungen umfasst.

Tabelle 10 - Beispiele für Beziehungen zwischen der Lösungsebene und der System-/Elementebene

	Beispiel Kunde/ Benutzer	Beispiel Wert	Beispiel Einnahmen	Beispiel Kosten
Lösungsebene	Kundengruppen der digitalen Lösung	Wertversprechen für den Kunden	Einnahmequellen durch digitale Zahlungen	Die Kostenstruktur muss die Betriebskosten für ein Rechenzentrum enthalten.
Systemebene	Benutzertypen im Einklang mit Kundengruppen	Szenarien veranschaulichen das Wertversprechen als Beispiele.	Der Zahlungsdienstleister muss ein Bestandteil des Systems sein. Ein Szenario sollte die Einnahmequelle veranschaulichen.	Mindestens ein Element des digitalen Systems wird von einem Rechenzentrum betrieben.
Elementebene	Technische Schnittstellen und User Interfaces, die den Bedürfnissen der Benutzertypen entsprechen	Vollständige Umsetzung des Wertversprechens durch Use Cases und Funktionen	Mindestens ein Element muss eine Schnittstelle zum Zahlungsanbieter bieten. Mindestens ein Use Case muss die Zahlung detailliert beschreiben.	Das Element im Rechenzentrum kommuniziert mit anderen Elementen über eine Netzwerkverbindung.

Diese Lücke zu schließen, ist die Aufgabe von Digital Design. Es gibt keinen klaren Regelsatz oder Algorithmus, der es ermöglicht, ein Systemdesignkonzept aus einem Lösungsdesignkonzept abzuleiten. Es ist die Kreativität der beteiligten Personen (siehe Abschnitt 2.1), die sich eine digitale Lösung und das zugrundeliegende digitale System vorstellen. Es gibt mehrere Möglichkeiten, ein Wertversprechen zu realisieren, und gleichzeitig kann eine bestimmte Technologie verschiedene Möglichkeiten der Wertschöpfung bieten. Ein guter DDP wird die Stärke haben, diese Lücke mit kreativen Ideen für Lösungen und Systeme zu überbrücken. Der einzige Hinweis, den wir hier geben können, ist, dass die Konzepte auf beiden Ebenen einander nicht widersprechen dürfen.

Tabelle 10 zeigt einige beispielhafte Beziehungen zwischen der Lösungsebene und der System-/Elementebene, um die Komplexität zu veranschaulichen.

2.2.6.5 Beziehung zwischen Designkonzepten, Tätigkeitsbereichen und Schritten des Bauprozesses

In Kapitel 1 haben wir das Designkonzept und das Realisierungskonzept als allgemeine Arbeitsprodukte der Tätigkeitsbereiche Design und Konstruktion vorgestellt. In Abschnitt 2.1 haben wir drei wesentliche Schritte des Bauprozesses vorgestellt und weitere spezielle Arten von Designkonzepten für jeden Schritt des Bauprozesses eingeführt.

Tabelle 11 - Konzepttypen im Digital Design

Art	Schwerpunkt/Zweck	Kontext
Digital Design Brief	Im Mittelpunkt des Digital Design Briefs steht das Projekt, mit dem die digitale Lösung geschaffen werden soll. Er dient dazu, allen Projektteilnehmern und anderen relevanten Stakeholdern die Einzelheiten des Projekts zu erläutern.	Der Kontext des Digital Design Briefs ist die Organisation, die das Projekt vorantreibt, und die potenziellen Märkte oder Bereiche, die von der digitalen Lösung angesprochen werden sollen.
Lösungsdesignkonzept	Der Schwerpunkt des Lösungsdesignkonzepts liegt auf der digitalen Gesamtlösung. Ihr Zweck ist es, die Gesamtidee der digitalen Lösung (einschließlich des Geschäftsmodells) zu definieren und den relevanten Stakeholdern zu vermitteln.	Der Kontext des Lösungsdesignkonzepts ist der Markt/die Domäne, der/die für die jeweilige digitale Lösung gewählt wurde.
Systemdesignkonzept	Im Mittelpunkt des Systemdesignkonzepts steht das System, mit dem die digitale Lösung realisiert wird. Ziel ist es, das digitale System in Elementen zu definieren und zu strukturieren, die eine effiziente Realisierung ermöglichen.	Der Kontext des Systemdesignkonzepts sind die Elemente (Benutzer und andere Systeme) des Marktes/Bereichs, die direkt mit dem digitalen System interagieren.
Softwaredesignkonzept	Im Mittelpunkt eines Softwaredesignkonzepts steht ein bestimmtes Softwareelement (z. B. eine App oder ein Webportal). Es dient dazu, alle Details zu liefern, die zur Realisierung der beschriebenen Software notwendig sind.	Der Kontext eines Softwaredesignkonzepts ist die Umgebung, in der die Software verwendet wird (einschließlich des Benutzers), und das Gerät, mit dem das Softwareelement betrieben wird.
Gerätedesignkonzept	Im Mittelpunkt eines Gerätedesignkonzepts steht ein bestimmtes Gerät (z. B. die Smartwatch). Sie dient dazu, alle Details zu liefern, die zur Realisierung des beschriebenen Geräts notwendig sind.	Der Kontext eines Gerätedesignkonzepts ist die Umgebung, in der das Gerät eingesetzt wird (einschließlich des Benutzers).

Die Stärke der vorgestellten Konzepte besteht darin, dass sie es Ihnen ermöglichen, die große Menge an Informationen zu strukturieren, die für die Entwicklung einer digitalen Lösung erforderlich sind, bis hin zu den Details der Elemente. Zu diesem Zweck hat jede Art von Konzept in Digital Design einen speziellen Fokus und beschreibt einen bestimmten Aspekt der digitalen Lösung. Damit das Konzept richtig verstanden wird, müssen Sie den Kontext, in dem ein Konzept erstellt wurde, ausreichend erläutern.

Tabelle 11 gibt einen Überblick über alle Konzepttypen, einschließlich ihres Schwerpunkts und ihres Kontexts.

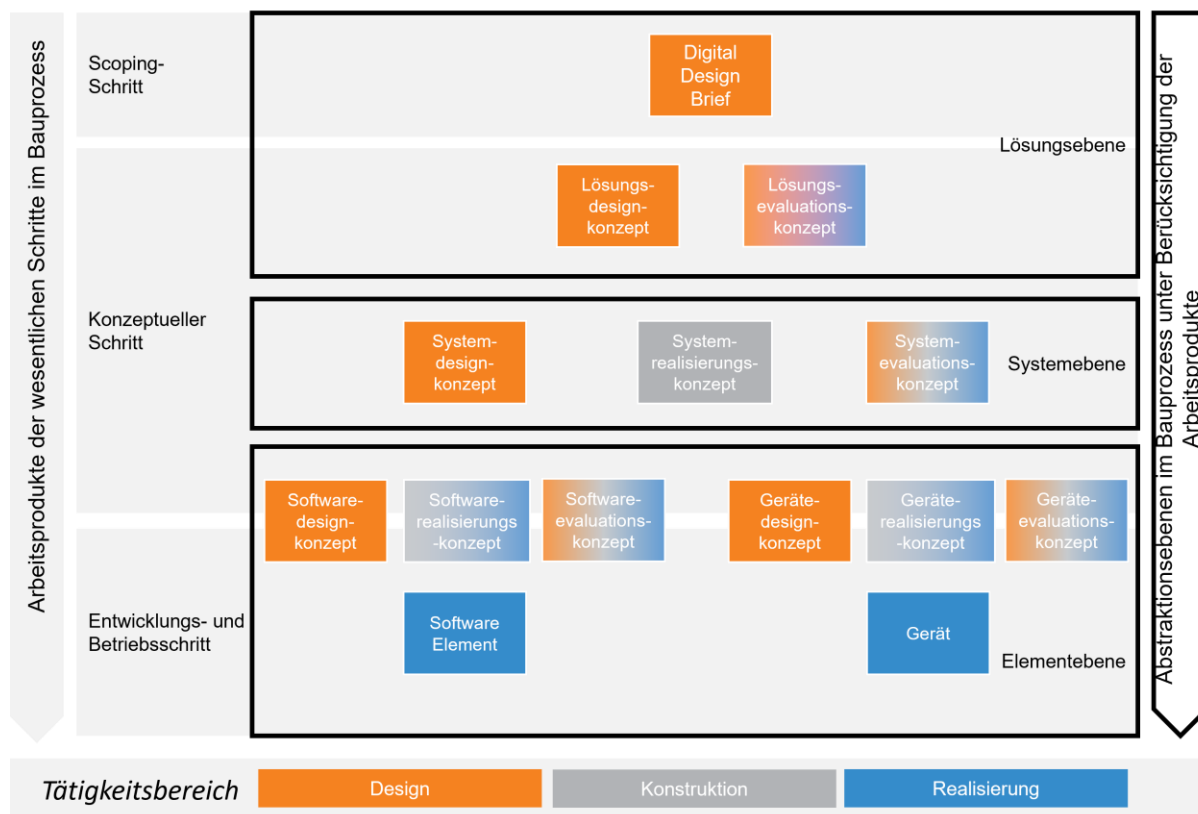


Abbildung 39 - Überblick über die Arten von Designkonzepten in den verschiedenen Phasen des Bauprozesses

Abbildung 39 bietet eine integrierte Visualisierung von Schritten, Arbeitsprodukten und Ebenen. Die Abbildung zeigt jedes Designkonzept in seinem Hauptschritt des Bauprozesses sowie die Beziehungen zwischen den Ebenen und dem Designkonzept. Beispielsweise gehört der Digital Design Brief zur Lösungsebene und wird hauptsächlich im Scoping-Schritt erstellt. Das Konzept des Lösungsdesigns gehört ebenfalls zur Lösungsebene, während das Konzept des Systemdesigns und der Realisierung zur Systemebene gehört. Beide Konzepte werden hauptsächlich in der konzeptuellen Phase erstellt.

Es ist wichtig, daran zu denken, dass die Definition eines Systems und seiner Elemente eine kontextabhängige Entscheidung ist (siehe Kapitel 1). Bei komplizierten Systemen mit vielen Elementen (z. B. besteht die Software-Landschaft eines großen Unternehmens nicht selten aus Dutzenden Elementen) ist es ratsam, die Definition dedizierter Teilsysteme zu erwägen, die als gemeinsame Elemente betrachtet werden. Wir betrachten diese mehrstufige Systemstruktur jedoch als ein fortgeschrittenes Thema.

Diese Zuordnung zu Stufen und Ebenen bedeutet nicht, dass die Konzepte nach Abschluss der jeweiligen Stufe abgeschlossen sind. Es bedeutet lediglich, dass eine erste Version des jeweiligen Designs (und Realisierungskonzepts) nach diesem Schritt fertig ist. Die oben beschriebenen Zusammenhänge zeigen, dass es von Vorteil ist, kontinuierlich und iterativ an allen Designkonzepten zu arbeiten und sie gegebenenfalls zu ändern.

Die Linien zwischen den verschiedenen gestalterischen (und technischen) Konzepten zeigen die wichtigsten Konsistenzabhängigkeiten an. Zum Beispiel muss das Lösungsdesignkonzept mit

dem Systemdesignkonzept übereinstimmen. Andere transitive Beziehungen sind möglich. Die Elementdesignkonzepte sind zwischen dem konzeptuellen Schritt und dem Entwicklungs-/Betriebsschritt angesiedelt. Der Hauptgrund dafür ist, dass einige Prozessmodelle bereits vor Beginn der Entwicklungsarbeit Teil-Elementdesignkonzepte erstellen.

Abbildung 39 zeigt auch die verschiedenen Arten von Design-, Realisierungs- und Evaluationskonzepten. Die Erstellung und Pflege dieser Konzepte erfordert zusätzliche Kompetenzen in den Tätigkeitsbereichen Konstruktion und Realisierung. Die verschiedenen Farben dieser Kästchen zeigen die Beiträge der einzelnen Tätigkeitsbereiche an.

2.2.7 Schlussfolgerung: Konzeptarbeit lernen

In diesem Abschnitt haben wir einen detaillierten Überblick über die Konzeptarbeit im Bereich des Digital Designs gegeben. Wir haben eine große Anzahl verschiedener Konzepttypen und Technologien eingeführt. Ziel dieses Abschnitts war es, die Vielfalt der Konzeptarbeit und die wichtigsten Beziehungen zwischen den verschiedenen Designkonzepten und ihren Bausteinen vorzustellen. Eine Zusammenfassung dieser Beziehungen wurde im vorherigen Abschnitt 2.2.6 gegeben.

Wir gehen nicht davon aus, dass das Lesen dieses Abschnitts ausreicht, um ein Experte für Konzeptarbeit zu werden. Die Absicht dieses Abschnitts war es, einen Ausgangspunkt für die Arbeit mit Designkonzepten zu schaffen. Die ergänzenden Materialien zu diesem Handbuch, insbesondere die YPRC-Fallstudie, bieten zusätzliches Material zum Erlernen von Konzeptarbeit.

Darüber hinaus hoffen wir, dass Leser und Leserinnen, die bereits über Erfahrungen im Bereich Digital Design verfügen, den breiten Überblick nützlich finden werden, um die Beziehungen zwischen ihren bevorzugten Konzepten und dem Blick des Digital Designs auf die Konzeptarbeit zu verstehen.

2.3 Anwendung von Prototypen im Digital Design

Dieser Abschnitt befasst sich mit der Anwendung von Prototypen im Digital Design. Er beginnt mit den Grundlagen von Prototypen in Form einer Definition (Abschnitt 2.3.1), den Zielen (Abschnitt 2.3.2) und der beispielhaften Anwendung von Prototypen in verschiedenen Disziplinen (Abschnitt 2.3.3).

Abschnitt 2.3.4 bietet einen umfassenden Überblick über die verschiedenen Arten von Prototypen, und Abschnitt 2.3.5 stellt verschiedene Werkzeuge zur Erstellung von Prototypen vor. Unter 2.3.6 werden Papierprototypen anhand eines Beispiels als einsatzbereite und einfache Technologie für Einsteiger vorgestellt. Abschnitt 2.3.7 schließt diesen Abschnitt mit einer Schlussfolgerung über die Anwendung von Prototypen im Digital Design ab.

2.3.1 Definitionen für Prototypen

Die Entwicklung einer digitalen Lösung kann ein breites Spektrum an Disziplinen und Berufen umfassen (siehe Abschnitt 1.3), und Digital Design beinhaltet Konzepte, Methoden und Techniken aus verschiedenen Disziplinen, z. B. Industriedesign, Interaktionsdesign, Mensch-Computer-Interaktion, Softwarearchitektur und Requirements Engineering. Um eine eindeutige Kommunikation zwischen Fachleuten aus diesen verschiedenen Bereichen zu gewährleisten, ist es für den DDP auf Foundation Level Niveau wichtig zu verstehen, dass es keine einheitliche und allgemein akzeptierte Definition des Begriffs Prototyp gibt. Fachleute und Stakeholder haben

möglicherweise ein unterschiedliches Verständnis des Begriffs Prototyp, weil sie je nach Disziplin einen anderen Hintergrund haben.

Industriedesigner verwenden beispielsweise den Begriff Prototyp für eine Vorabversion eines Serienprodukts. Andererseits erstellen sie viele frühe Versionen des Produkts in Form von Skizzen, (Computer-)Modellen oder einfachen Attrappen, um Entwürfe zu verbessern - Fachleute aus anderen Bereichen würden diese als Prototypen bezeichnen. Interaktionsdesigner verwenden Skizzen, interaktive Mock-ups, User Flows, Wireframe-Prototypen oder Papierprototypen, um vorläufige Versionen eines (grafischen) User Interface darzustellen. Diese vorläufigen Versionen können zusammen mit und von den Stakeholdern erforscht werden. Software-Designer oder Software-Architekten verwenden frühe Implementierungen eines bestimmten Teils der digitalen Lösung (einen funktionalen Prototyp), um die Machbarkeit dieser speziellen Software-Funktion zu untersuchen.

Um Kommunikationsprobleme mit den anderen am Bau einer digitalen Lösung beteiligten Disziplinen zu reduzieren, bietet die folgende Definition ein breites Verständnis des Begriffs Prototyp für die Verwendung im Kontext von Digital Design und dem Bau digitaler Lösungen.

Prototype: A preliminary, partial instance of a design solution.

Prototypes can be used as

- 1) A manifestation of an idea for a future digital solution in a format that communicates the idea to others or that can be tested with customers or users**
- 2) A model for later stages or for the final, complete version of the digital solution**
- 3) A means of obtaining early feedback on a concept by providing a working model of the expected digital solution before actually building it**

Based on [McEI2017] and [IEEE2017]

Die Verwendung eines Prototyps oder von Prototypen wird als *Prototyping* bezeichnet. Wie in [Dick2019] ausgeführt, wird der Begriff *Prototyp* (das Objekt) manchmal verwendet, wenn *Prototyping* (der Prozess der Arbeit an und mit einem Prototyp) gemeint ist.

Der weite Blick auf Prototypen, wie er oben definiert wurde, könnte uns dazu verleiten, ein Digital Design-Konzept selbst als Prototyp zu betrachten. Dies mag abstrakt betrachtet gültig sein. In der obigen Definition wird die Verknüpfung jedoch zu einem bestimmten Ziel hergestellt, z. B. der Kommunikation mit Teammitgliedern oder Tests mit Kunden oder Benutzern. Solche Tests sind insbesondere mit einer Spezifikation schwierig durchzuführen - zum Beispiel nur mit dem Digital Design-Konzept. Ein weiteres Beispiel: Es ist schwierig, die Idee der zukünftigen digitalen Lösung mit den Benutzern zu testen, wenn die Idee in Form einer Liste von User Storys in einem Arbeitsblatt präsentiert wird. Solche Zwecke erfordern eine sehr konkrete Version der frühen digitalen Lösung, des Systems oder des Elements. Ein konkreter Prototyp kann entweder physisch anfassbar (Tangible) oder nicht anfassbar sein. Ein Beispiel für einen anfassbaren Prototyp ist ein interaktives Mock-up, das es dem Kunden oder Benutzer ermöglicht, mit Teilen der geplanten digitalen Lösung zu interagieren. Beispiele für physisch nicht anfassbare Prototypen sind einfache Skizzen oder Storyboards, die die (frühen) Ideen viel konkreter darstellen, als es ein abstrakter Text einer Spezifikation kann. Um den Kontext und das geplante Kunden- oder Benutzererlebnis zu verdeutlichen, können kurze Erzählungen in Form von

Storyboards oder fiktiven (Werbe-)Videos Klarheit über die anvisierte digitale Lösung schaffen. Siehe Abschnitt 2.3.2 für eine Diskussion über die Gründe und Ziele des Prototypings und Abschnitt 2.3.4.5 für typische Bezeichnungen von Prototypen, die im Digital Design verwendet werden.

2.3.2 Ziele von Prototypen

Die Verwendung von Prototypen ist die Schlüsseltechnik, die es dem DDP ermöglicht, mehrere (sich teilweise überschneidende) Ziele zu erreichen (vgl. [McEI2017]):

- Erkunden des Problems, der Bedürfnisse und Anforderungen
- Kommunikation von Lösungsideen und Konzepten
- Testen und verbessern von Konzepten und Lösungsideen
- Eine Lösung oder eine Lösungsidee durchsetzen

In all diesen Fällen unterstützt die Erstellung von Prototypen die Iteration von Problemen, Anforderungen, Konzepten, Lösungsideen und Lösungen. Der DDP lernt durch das Erstellen eines Prototyps und kann den Prototyp nutzen, um Feedback von Stakeholdern einzuholen und auf Basis dieses Feedbacks eine anschließende Verbesserung erzielen. Solche Iterationszyklen sind ein wesentlicher Bestandteil jedes Bauprozesses (siehe Kapitel 1 und Abschnitt 2.1).

Das Erstellen eines Prototyps ist mit einem gewissen Aufwand verbunden. Dieser Aufwand ist jedoch gerechtfertigt, wenn das gesammelte Feedback dazu beiträgt, Entscheidungen auf mehr und validen Informationen zu stützen und somit bessere Entscheidungen zu treffen. Wenn man mit Hilfe von Prototypen Lösungsideen in vielen grundlegend verschiedenen Richtungen untersucht, werden viele dieser Ideen scheitern, aber es werden auch neue Ideen für bessere Lösungen generiert. Das bedeutet, dass der DDP bereit sein muss, einen Prototyp nur für einen einzigen Zweck zu erstellen, um diesen danach zu verwerfen. Daher sollte das DDP den Anwendungsbereich für den Prototyp gezielt auswählen und den Aufwand für dessen Erstellung sorgfältig abwägen.

Die Leistungsfähigkeit des iterativen Prototypings kann die Suche nach innovativen und exzellenten Konzepten und Lösungen erleichtern und die Wahrscheinlichkeit deutlich erhöhen, dass aus einer Lösungsidee ein echtes innovatives Produkt oder eine Dienstleistung wird.

Die oben genannten Ziele werden in den folgenden Abschnitten im Einzelnen erläutert. Ein Teil des Inhalts basiert auf [McEI2017].

2.3.2.1 Erkundung des Problems, der Benutzerbedürfnisse und der Anforderungen

Das Hauptziel der Problemerkundung besteht darin, die Sichtweise des Benutzers in Bezug auf die zu entwickelnde digitale Lösung zu verstehen. Prototypen sind ein gutes *Denkwerkzeug*, um die Ursachen für die Probleme der Benutzer oder die zentralen Bedürfnisse der Benutzer zu erforschen. Sie helfen auch dabei, den Umfang einer potenziellen digitalen Lösung auf der Grundlage verfügbarer oder neuer Technologien auszuloten. Oft zeigt sich bei der Erkundung durch früh entwickelte Prototypen (frühe Prototypen) ein umfassenderes und anderes Problem als ursprünglich angenommen wurde.

Beispiel YPRC. In der Anfangsphase der Entwicklung der digitalen YPRC-Lösung könnte man zunächst davon ausgehen, dass der Remote-Coach die aktuelle Pulsfrequenz des Läufers sehen möchte, um ihm Trainingsempfehlungen geben zu können. Die Skizzierung früher Entwürfe der Bildschirme des User Interface ermöglicht es, die geplante zukünftige digitale Lösung mit professionellen Lauftrainern zu diskutieren. Das Ergebnis einer solchen Diskussion könnte sein, dass nur die Veränderungen des Pulses während einer bestimmten Zeitspanne und nicht der aktuelle Wert wichtig sind. Außerdem muss der Trainer den Ruhepuls des Läufers und die Maximalwerte aus früheren Trainingseinheiten kennen. Diese Erkenntnis verändert das User Interface für den Coach und damit auch wichtige zugrundeliegende Teile der digitalen Lösung.

Ein weiteres Beispiel, das sich auf die YPRC-Fallstudie bezieht, ist die anfängliche Annahme, dass der Läufer in der Lage sein muss, seine Position auf der Strecke auf einer Karte zu sehen. Benutzerstudien mit Läufern anhand von Papierprototypen (siehe Abschnitt 2.3.4.5 und Abschnitt 2.3.6) könnten zeigen, dass sie ihre Laufroute im Allgemeinen kennen und die Kartenansicht auf dem Smartphone nur für neue Routen benötigen. Deshalb tragen sie ihr Smartphone meist in einer Tasche oder am Oberarm. Es kann auch sein, dass Läufer über das Verlassen eines vordefinierten Pulsbereichs durch einen anderen Informationskanal informiert werden möchten, z. B. durch eine Vibration oder einen akustischen Alarm anstatt über das Display des Smartphones.

Die obigen Beispiele zeigen, wie Prototypen helfen, alternative Probleme im Problembereich zu beleuchten. Die Lösung jedes Problems führt letztlich zu einer besseren digitalen Lösung. Da dies das grundlegende Konzept einer digitalen Lösung in Frage stellen und möglicherweise verändern könnte, ist es sehr wichtig, solche Erkenntnisse bereits zu Beginn des Entwicklungsprozesses zu gewinnen (siehe Abschnitt 2.1).

Praxistipp

Nach dem Verstehen des Problems ist der DDP besser in der Lage, eine echte Innovation zu schaffen und darauf aufbauend erste Designkonzepte zu entwickeln. Dieses Verständnis ist der Ausgangspunkt für die Entwicklung von Lösungsvarianten für das Problem. Prototypen sind sehr wertvoll für das iterativ Erkunden von Lösungen. Sobald solche prototypischen Lösungen - die grundlegend unterschiedliche Lösungsansätze darstellen können - erstellt worden sind, sollten Tests mit Benutzern stattfinden, um diese Alternativen zu vergleichen (siehe Abschnitt 2.3.2.3). Dieser Vergleich und die Auswahl der besten Alternative sollte erfolgen, sobald ein erstes Konzept vorliegt, aber dennoch in einem frühen Stadium des Bauprozesses.

Prototypen können dazu beitragen, Geschäftsstrategien im Zusammenhang mit digitalen Wettbewerbslösungen, der Ausrichtung von Produktportfolios und den Zielen der Zielbenutzer zu verstehen. Prototypen können eine Geschäftsausrichtung in ein physisch anfassbares oder zumindest sichtbares Objekt verwandeln, so dass Präsentationen und Diskussionen mit den Stakeholdern sehr konkret werden (siehe Abschnitt 2.3.2.4). Diskussionen auf der Grundlage solcher konkreten Beispiele sind sehr effektiv und führen zu Verbesserungsideen für den nächsten Iterationszyklus. Diese Form des Prototyping ist in allen Phasen des Bauprozesses anwendbar.

Im weiteren Verlauf des Entwicklungsprozesses helfen Prototypen bei der Gestaltung der User Experience und der interaktiven Elemente des User Interface. Präsentationen und Tests solcher

Lösungsmanifestationen unterstützen das Verständnis der Benutzerabläufe und das Entdecken von neuen Design-Aspekten.

Das Verständnis des Benutzers im Allgemeinen und der Benutzerinteraktion mit der digitalen Lösung ist in allen Phasen des Bauprozesses hilfreich. Der wichtigste Teil ist jedoch das Verständnis der Benutzerbedürfnisse, das auch im ersten Teil des menschenzentrierten Designprozesses enthalten ist (siehe [McEI2017], Abschnitt 2.1 und [ISO2019]). Darüber hinaus werden innerhalb dieses Prozesses nicht nur die Bedürfnisse der Benutzer abgedeckt, sondern es können auch die Bedürfnisse anderer Stakeholder, wie z. B. der Auftraggeber und Kunden der digitalen Lösung, berücksichtigt werden. Wenn die zentralen Probleme klar sind und eine gute Lösung für diese Kernbedürfnisse entwickelt wurde, wird die digitale Lösung wahrscheinlich ein Erfolg auf dem Markt sein.

Neue technologische Entwicklungen können zu digitalen Lösungen führen, an die bisher kein Stakeholder gedacht hat. Diesem Problem kann der DDP mit Hilfe des Dual-Mode-Modells des Designs begegnen (siehe Abschnitt 2.1.1.2). Die Erstellung von Prototypen für Lösungsideen, die sich aus neuen technologischen Entwicklungen ergeben, und deren Präsentation für Benutzer und andere Stakeholder hilft, neue Anforderungen zu entdecken und Inspiration für innovative digitale Lösungen zu erhalten.

2.3.2.2 Kommunikation von Lösungsideen und Konzepten

Prototypen manifestieren Gedanken in einem physischen oder digitalen Medium. Sie verwandeln unscharfe oder allgemeine Gedanken in ein konkretes beispielhaftes Objekt. Bei einem solchen Objekt kann es sich um ein anfassbares physisches Objekt handeln, z. B. ein Prototyp auf Papier, oder um ein immaterielles Objekt, z. B. eine Story Map oder ein Storyboard (siehe Abschnitt 2.3.4.5). Diese Objekte dienen als Grundlage für die Kommunikation zwischen allen Stakeholdern, wie z. B. Kunden, Teamkollegen, Entwicklern, Testexperten und Projektmanagern. Mit einem Prototyp als konkreter Diskussionsgrundlage werden die Treffen mit diesen Stakeholdern effektiver und effizienter, insbesondere wenn die Teilnehmer konkretes Feedback zum Prototyp geben können und dies dürfen. Durch die iterative Anwendung dieser Feedback-Schleife werden Verbesserungsideen immer deutlicher, und das Verständnis aller Beteiligten wächst mit jeder Iteration.

Im Allgemeinen hängt die Wahl einer geeigneten Prototyp-Kategorie (siehe Abschnitt 2.3.4) von den Kommunikationspartnern und dem Ziel des jeweiligen Treffens ab. Für die Kommunikation mit Fachkollegen sind oft Low-Fidelity-Prototypen ausreichend, da Kollegen ähnliche mentale Modelle haben. High-Fidelity-Prototypen sind dann angebracht, wenn Feedback zu visuellen Designergebnissen oder komplexen Interaktionen erforderlich ist. Dies geschieht meist in späteren Phasen des Bauprozesses. Iterationen mit Fachkollegen sind in allen Prozessphasen nützlich, aber besonders in späteren Phasen, um zu vermeiden, dass sich der DDP zu sehr auf seine eigenen Lösungen und sein eigenes Denken fokussiert.

Praxistipp

Bei der Kommunikation mit projektverantwortlichen Stakeholdern., wie z. B. Projekt- und Produktmanager oder Kunden, hängt die Wahl des Prototyps von den jeweiligen Gesprächszielen in den einzelnen Phasen des Prozesses ab. Sie hängt auch vom aktuellen Kenntnisstand der Teilnehmer und ihrer Fähigkeit ab, sich die angestrebte digitale Lösung vorzustellen. Visuelle oder anfassbare Elemente in Form eines Prototyps können dazu beitragen, ein besseres Verständnis zu entwickeln. Außerdem können kurze Erzählungen in Form von Storyboards oder

fiktiven (Werbe-)Videos (siehe Abschnitt 2.3.4.5) Klarheit über die Vision hinter der geplanten digitalen Lösung schaffen. Generell gilt, dass in frühen Phasen des Entwicklungsprozesses Low-Fidelity-Prototypen für die frühe Abstimmung und Genehmigung von Benutzerabläufen (User Flows), Use Cases und Funktionen nützlich sind. Im weiteren Verlauf des Prozesses kann der Detailgrad parallel zum Reifegrad der digitalen Lösung steigen. Es besteht allerdings häufig das Risiko, dass der Detailgrad zu hoch ist, so dass die Entscheidungsträger oder Kunden denken könnten, die Entwicklung der digitalen Lösung sei bereits abgeschlossen. Um später im Prozess eine Entscheidung über bestimmte Details herbeizuführen, sind ein Mid-Fidelity-Prototypen mit mittlerem Detaillierungsgrad oder ein Mixed-Fidelity-Prototypen die richtige Wahl.

Softwareentwickler oder Hersteller, die (einen Teil) der digitalen Lösung umsetzen sollen, benötigen detaillierte Spezifikationen oder Styleguides, um den Umfang und die Details ihrer Arbeit zu bestimmen. High-Fidelity-Prototypen können die Spezifikationen ergänzen. Die Präsentation von Prototypen, um zukünftige Investoren oder Mitarbeiter zu überzeugen, trägt dazu bei, das Vertrauen in die digitale Lösungsidee oder das Digital Design-Konzept zu stärken.

2.3.2.3 Testen und Verbessern von Konzepten und Lösungsvorschlägen

Ein Schwerpunkt des Einsatzes von Prototypen ist das Testen einer Verbesserung (von Teilen) des Digital Design-Konzepts und der digitalen Lösung. Sie sind sehr nützlich für Machbarkeitsstudien. Iterative benutzerbasierte Tests mit diesen Prototypen helfen sicherzustellen, dass die geplante Lösung tatsächlich funktioniert und um Verbesserungsideen zu sammeln. Selbst die Validierung oder Entkräftung weniger, detaillierter Annahmen - zumindest bei einer kleinen Anzahl von Benutzern - ist wertvoll, um den Bauprozess voranzubringen. Auch eine technische Bewertung der Machbarkeit oder Qualitätssicherung ist möglich. Solche Prototypen werden in der Regel nur für den einmaligen Gebrauch entwickelt (Wegwerfprototypen), was in den meisten Fällen den Aufwand wert ist, wenn die daraus resultierende digitale Lösung erfolgreich wird.

2.3.2.4 Empfehlung für eine Lösung oder eine Lösungsidee

Prototypen sind ideal, um für bestimmte Gestaltungsmöglichkeiten oder Lösungen zu werben. Zusammen mit den Erkenntnissen aus vorangegangenen Tests können Produktmanager, Business Stakeholder und Projektmanager von der richtigen Richtung des Digital Design Konzepts oder der digitalen Lösung überzeugt werden. In großen Projektteams ist Überzeugungsarbeit notwendig, um bestimmte Lösungsansätze, die von den ursprünglichen Ideen des Teams abweichen, zu motivieren. In großen Projektteams ist das Werben für bestimmte Entwicklungslösungen, die von den ursprünglichen Ideen des Teams abweichen, notwendig, um Motivation zu schaffen. Die richtigen Daten aus iterativen (Benutzer-)Tests helfen dabei, das Team zu überzeugen.

Die frühzeitige Einbindung von Entscheidern und Entwicklern in den iterativen Feedback-Prozess zu alternativen prototypischen Gestaltungsmöglichkeiten oder Lösungen ist für die digitale Lösung genauso wertvoll wie das frühzeitige Commitment der Beteiligten für die vorgestellten Gestaltungsmöglichkeiten oder Lösungen. Darüber hinaus sollten Stakeholder die Möglichkeit haben, Test-Sessions der digitalen Lösung zu beobachten, um sich selbst davon zu überzeugen, ob die Lösungen funktionieren oder nicht. Dies führt in der Regel zu einem besseren Verständnis für bestimmte Gestaltungsmöglichkeiten und -lösungen.

Praxistipp

Die Erfahrung aus Softwareentwicklungsprojekten zeigt jedoch, dass bei einigen Projekten die Tendenz besteht, zu früh mit der Implementierung von Software zu beginnen, wodurch häufig eine konkrete (Teil-)Lösung in einem frühen Stadium des Projekts festgelegt wird. Beobachtungen aus solchen Projekten zeigen die Tendenz, Klick-Prototypen zu erstellen, die zu anspruchsvoll sind. Beispielsweise wird zu früh im Prozess mit einem zu hohen Fidelity-Level gearbeitet. Dies schränkt in der Regel die Erkundung von (grundlegend) anderen Konzepten oder Lösungswegen ein, da Designer und Entwickler zögern, Lösungen zu verwerfen, in die sie viel Arbeit investiert haben. Außerdem fokussieren Prototypen mit einem zu hohen Detailgrad den Betrachter oder Benutzer dieses Prototyps zu sehr auf eher kleine Details, die zu Beginn des Projekts normalerweise nicht im Fokus stehen. Dies schränkt die Möglichkeit ein, das Gesamtproduktkonzept zu diskutieren oder zu bewerten, weil die Gesprächspartner durch Details regelrecht abgelenkt werden.

2.3.3 Beispiele für die Verwendung von Prototypen in unterschiedlichen Disziplinen

Prototypen spielen in unterschiedlichen Disziplinen eine wichtige Rolle. Die folgende Liste enthält Beispiele aus der digitalen Industrie und anderen Disziplinen.

- Gebäudearchitekten arbeiten beispielsweise mit Grundrissen, Luftströmungsmodellen für Lüftungen, Heizungen und Kühlungen, Tageslichtsimulationen zur Optimierung des Lichteinfalls durch Fenster, Materialstudien, ästhetischen Modellen und Gebäudesimulationen, die es ermöglichen, virtuell durch das geplante Gebäude zu gehen (siehe [McEI2017]).
- Industriedesigner haben eine lange Tradition in der umfassenden Verwendung von Prototypen. Sie verwenden Skizzen (z. B. erstellt mit echten oder digitalen Stiften), Schaumstoffmodelle oder Modelle aus der additiven Fertigung (z. B. 3D-Druck); sie führen Materialstudien durch, verwenden ästhetische Modelle, fertigen maßstabsgetreue Modelle an und erstellen endgültige Formen als Prototypen, bevor sie das Designergebnis an die Fertigung weitergeben (siehe [McEI2017]).
- Filmemacher verwenden Storyboards und Filmvorschauen, um den Ablauf eines Films zu visualisieren, bevor die Szenen tatsächlich gedreht werden.
- Bei der Entwicklung elektronischer Geräte entwerfen die Designer in der Regel Industrie- und Elektronikdesigns parallel. Die Platzierung der elektronischen Komponenten beeinflusst das Industriedesign. Prototypen, die aus ausgewählten, montierten elektronischen Komponenten bestehen, sind nützlich, um die Auswirkungen auf das Industriedesign zu untersuchen und dieses zu optimieren (siehe [McEI2017]).
- Interaction Designer, die User Interfaces von Softwareanwendungen entwickeln, verwenden Prototypen wie Skizzen, Wireframes, in Software programmierte Prototypen und visuelle Entwürfe, um eine Lösung iterativ zu verbessern (vgl. [McEI2017]).
- Softwarearchitekten und Softwareentwickler verwenden Teilimplementierungen von Software als funktionale Prototypen, um die Machbarkeit bestimmter Lösungen zu untersuchen, Anforderungen zu verifizieren oder bestimmte Qualitätsaspekte alternativer Softwarelösungen zu untersuchen.

Die obigen Beispiele zeigen, dass die Erstellung und Verwendung von Prototypen gängige und erfolgreiche Techniken in unterschiedlichen Disziplinen sind. Die Beispiele sollen als Inspiration und Motivation für den DDP dienen, verschiedene Arten und Kategorien von Prototypen - die

nicht unbedingt Prototypen in einem digitalen Format sein müssen - zu verwenden, um digitale Lösungen zu bauen.

2.3.4 Kriterien für die Kategorisierung von Prototypen

Es gibt verschiedene Ansätze zur Kategorisierung von Prototypen. Die Auseinandersetzung mit solchen Klassifizierungen hilft dem DDP, das breite Spektrum von Prototypen zu verstehen und bei der Auswahl eines geeigneten Prototyps in einer bestimmten Situation und für eine bestimmte Prozessphase.

Die linke Spalte von Tabelle 12 enthält eine Übersicht über die Kriterien zur Klassifizierung von Prototypen, die in den folgenden Abschnitten näher erläutert werden. Die rechte Spalte dieser Tabelle enthält die Kategorien, die auf den Kriterien der linken Spalte basieren. Die Tabelle wird nachstehend im Detail erläutert. In Abschnitt 2.3.4.6 wird abschließend eine Zuordnung zwischen Kategorien und Bauprozessschritten vorgestellt.

Tabelle 12 - Kriterien für die Kategorisierung von Prototypen

Kriterium für die Kategorisierung	Kategorien
<i>Interaktionsebene</i>	<i>horizontal, vertikal</i>
<i>Ziel des Prototypings</i>	<i>explorativ, experimentell, evolutionär</i>
<i>Detailgrad (Level of Fidelity)</i>	<i>niedrig, mittel, hoch</i>
Detailgrad für jede Dimension eines Prototyps (Fidelity-Profil)	
<i>Sensorische Vollkommenheit (sensory refinement)</i>	<i>niedrig, mittel, hoch</i>
<i>Funktionsbreite (breadth of functionality)</i>	<i>niedrig, mittel, hoch</i>
<i>Funktionstiefe (depth of functionality)</i>	<i>niedrig, mittel, hoch</i>
<i>Interaktionsumfang (richness of interactivity)</i>	<i>niedrig, mittel, hoch</i>
<i>Datenmodellumfang (richness of data model)</i>	<i>niedrig, mittel, hoch</i>

2.3.4.1 Interaktionsebene

Eine sehr einfache, aber häufig verwendete Kategorisierung von Prototypen nach der Ebene der Interaktion, unterscheidet zwischen horizontalen und vertikalen Prototypen. Horizontale Prototypen bestehen aus einem umfassenden User Interface mit wenig oder keiner Funktionalität. Vertikale Prototypen stellen eine Teilmenge der Zielfunktionalität in der Tiefe bereit, decken aber nicht viele Funktionen oder das komplette User Interface ab.

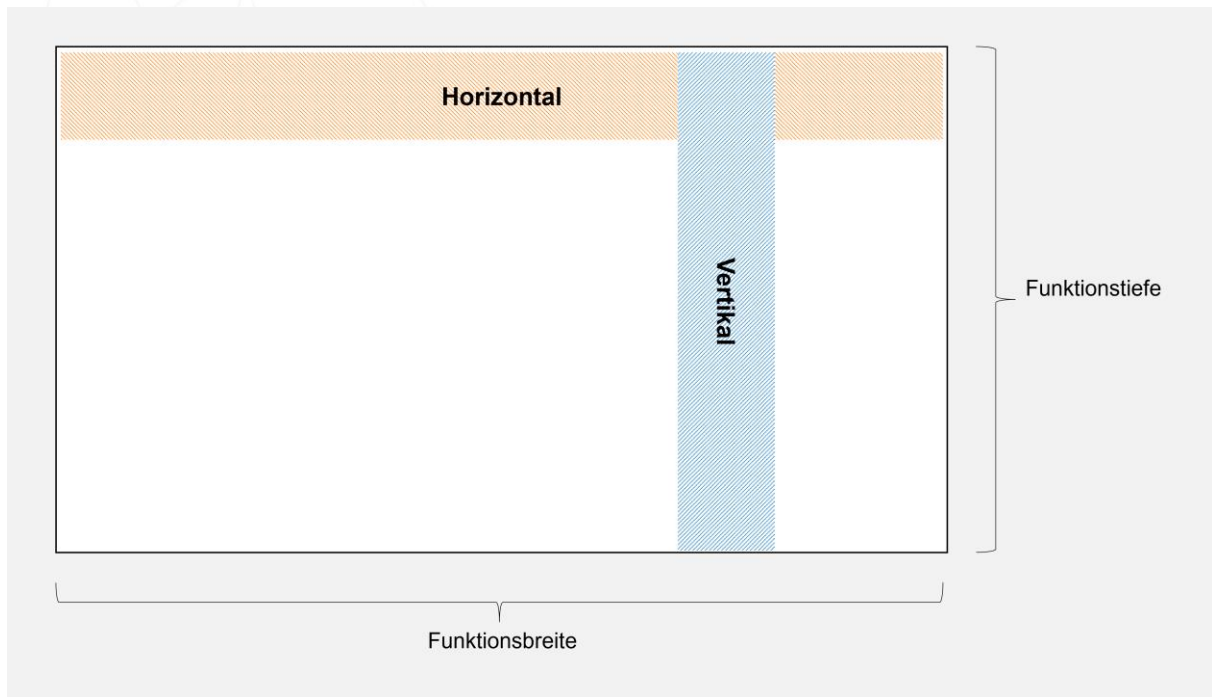


Abbildung 40 – Interaktionsumfang

Abbildung 40 visualisiert die Korrelation dieser Prototypenkategorien mit der Anzahl der Funktionen und der Funktionstiefe. Diese einfache, aber häufig verwendete Kategorisierung wird auch durch die Kategorisierung anhand der fünf Dimensionen von Prototypen (Fidelity-Profil) abgedeckt, die in Abschnitt 2.3.4.4 vorgestellt wird.

2.3.4.2 Ziel des Prototypings

Floyd [Floy1984] unterscheidet zwischen Prototyping für die Exploration, das Experimentieren und die Evolution von Lösungen, in Abhängigkeit vom Ziel des Prototypings innerhalb der gegebenen Projektsituation.

Explorative Prototypen unterstützen die Kommunikation zwischen Benutzern und Entwicklern des Zielsystems in frühen Phasen des Entwicklungsprozesses. Sie sind nützlich für das Sammeln von Ideen, das Erheben von Anforderungen und die Validierung. Außerdem schafft die Visualisierung der Anforderungen durch solche Prototypen Transparenz bezüglich der Funktionen des Zielsystems. Exploratives Prototyping ist bereits in frühen Phasen des Entwicklungsprozesses sinnvoll, wobei der Schwerpunkt auf der Erkundung von (grundlegend) unterschiedlichen alternativen Konzepten und Lösungen liegt. Solche Prototypen werden oft nur für diese Zwecke verwendet (Wegwerfprototypen) und sind horizontale Prototypen (siehe [Floy1984]).

Experimentelle Prototypen unterstützen die Evaluierung der Machbarkeit oder Nützlichkeit einer bestimmten Lösung, bevor erheblicher Aufwand in die Realisierung der digitalen Lösung gesteckt wird. Die Überprüfung von Ideen (zur Umsetzung) liefert Erfahrungen für eine spätere Implementierung. Das experimentelle Prototyping ist in frühen Phasen des Bauprozesses nützlich, nachdem eine erste Spezifikation, wie z. B. der Design Brief oder ein erstes Konzept (siehe Abschnitt 2.2.3), erstellt wurde. Solche Prototypen verbessern die Spezifikation des Zielsystems, indem sie entweder die Spezifikation ergänzen, (einen Teil) der Spezifikation verfeinern oder als Zwischenschritt zwischen Konzepterstellung und Realisierung dienen. Bei solchen Prototypen handelt es sich meist um Wegwerfprototypen und vertikale Prototypen (siehe [Floy1984]).

Wenn neue Ideen, Anforderungen oder Funktionen erst später im Prozess auftauchen, kann es sich lohnen, vor der eigentlichen Konstruktion oder Realisierung dieser Elemente explorative oder experimentelle Prototypen zu erstellen.

Das Ziel von **evolutionären Prototypen** ist die kontinuierliche Entwicklung und Weiterentwicklung von Prototypen zusammen mit der angestrebten digitalen Lösung. Am Ende des Realisierungsprozesses werden die Prototypen zu einer endgültigen digitalen Lösung integriert. Wenn nur ein Prototyp verwendet wird, wird dieser im Projektverlauf zur endgültigen digitalen Lösung entwickelt. Die Verwendung dieser Prototypen zu Testzwecken ermöglicht eine kontinuierliche Validierung der in Entwicklung befindlichen digitalen Lösung. Indem der Prototyp oder die Prototypen nicht verworfen, sondern in das endgültige System integriert werden, werden sie zu einem festen Bestandteil der endgültigen digitalen Lösung. Floyd [Floy1984] hat bereits festgestellt, dass ein evolutionärer Prototyping-Ansatz eher eine *Entwicklung in Versionen* ist. Bei diesem Ansatz wird ein linearer, wasserfallartiger Entwicklungsprozess in aufeinanderfolgende, iterative Zyklen von (Re-)Design, (Re-)Implementierung und (Re-)Evaluation unterteilt (siehe [Floy1984]). Solche Iterationen sind die grundlegende Basis der heutigen modernen agilen Entwicklungsparadigmen.

2.3.4.3 Detailgrad (Level of Fidelity)

Eine Kategorisierung von Prototypen, die häufig angewendet wird, verwendet den Detailgrad als Kriterium. Der Detailgrad eines Prototyps beschreibt auf einer kontinuierlichen Skala von niedrig bis hoch, wie nahe der Prototyp an der endgültigen digitalen Lösung ist. Dies gilt sowohl für die visuelle Darstellung als auch für das Verhalten des Prototyps.

Diese einfache Kategorisierung hilft beim Verständnis des allgemeinen Konzepts der Charakterisierung eines Prototyps. In vielen praktischen Fällen ist es jedoch sinnvoller, den Detailgrad auf verschiedene Dimensionen eines Prototyps einzeln anzuwenden. Eine Kategorisierung anhand von fünf Dimensionen von Prototypen (Fidelity-Profil) wird in Abschnitt 2.3.4.4 vorgestellt. Diese dort vorgestellte Kategorisierung schließt die in diesem Abschnitt dargestellte Kategorisierung von Prototypen auf der Grundlage des allgemeinen Detailgrads ein. Der verbleibende Teil dieses Abschnitts konzentriert sich auf eine allgemeine Betrachtung des Detailgrads eines Prototyps.

Low-Fidelity-Prototypen sind weit von der endgültigen digitalen Lösung entfernt. Ein Prototyp dieser Kategorie wird gewöhnlich in einem anderen Medium erstellt, d. h. es wird ein anderes Material verwendet als bei der endgültigen digitalen Lösung. Dieser Prototyp verfügt in der Regel über eine begrenzte Anzahl von Funktionen und hat normalerweise kein (ausgearbeitetes) visuelles Designergebnis. Beispiele sind Papierprototypen, bei denen es sich in ihrer einfachsten Form um Zeichnungen auf Papier handelt, die z. B. bestimmte Smartphone-Bildschirme zeigen. Low-Fidelity-Prototypen sind die einfachste und kostengünstigste Prototypenkategorie, erfordern den geringsten Zeitaufwand und zudem nur ein geringes Maß an Fachkenntnis für die Erstellung. Low-Fidelity-Prototypen eignen sich am besten zum Testen von Kernkonzepten und zum Ausloten verschiedener alternativer Ideen. Außerdem helfen sie, mögliche Probleme der endgültigen digitalen Lösung frühzeitig zu erkennen. Da Low-Fidelity-Prototypen leicht zu erstellen sind, lassen sie sich schnell und kostengünstig realisieren (siehe [McEl2017]).

Low-Fidelity-Prototypen eignen sich gut zum Testen und Evaluieren von Kernannahmen oder Hypothesen über die digitale Lösung. Benutzerinteraktionen, Informationsarchitekturen und mentale Modelle der Benutzer können untersucht werden. Bei der Verwendung dieser

Prototypenkategorie liegt der Schwerpunkt in der Regel auf der allgemeinen Nutzung und dem allgemeinen Ablauf der digitalen Lösung. Beispiele für Low-Fidelity-Prototypen sind Skizzen, Papierprototypen, Storyboards, Wireframe-Prototypen, Moodboards und einfache Baugruppen elektronischer Komponenten (Komponentenprototyp) (siehe [McEI2017]).

High-Fidelity-Prototypen befinden sich am anderen Ende der Fidelity-Skala und sind nahe an der endgültigen digitalen Lösung. In der Regel handelt es sich bei einem Prototyp dieser Kategorie um das endgültige Medium, d. h. es wird das gleiche Material wie für die digitale Lösung verwendet. Diese Art von Prototyp hat ein hochwertiges visuelles Designergebnis und enthält reale Inhalte, wie z. B. Produktbilder in einem Prototyp für eine Shop-App oder Musikdateien in einem Prototyp für eine Musik-App. Die meisten Funktionen und die meisten Interaktionspfade sind verfügbar. Da der Detaillierungsgrad sehr hoch ist, sind Zeit- und Kostenaufwand für die Erstellung eines High-Fidelity-Prototypen ebenfalls hoch. Für die Erstellung solcher realitätsnahen Prototypen ist in der Regel ein hohes Maß an Fachwissen erforderlich (siehe [McEI2017]), z. B. Kenntnisse im Interaktionsdesign oder in der Softwareentwicklung.

High-Fidelity-Prototypen sind gut geeignet, um kleine Details zu testen und eine abschließende Gesamtbewertung der digitalen Lösung kurz vor der ersten Markteinführung vorzunehmen. Es können Icons, Animationen, detaillierte Benutzerabläufe und die Größe der endgültigen Schaltflächen und Tasten untersucht werden. Diese Prototypen-Kategorie eignet sich auch, um die Lesbarkeit von Schriftarten zu testen und Langzeitstudien durchzuführen, wie gut sich ein Gerät (als Teil einer digitalen Lösung) längere Zeit am Körper tragen lässt. High-Fidelity-Prototypen sind nützlich, um die Haltbarkeit oder langfristige Stabilität des physischen Teils der angestrebten digitalen Lösung zu testen. Beispiele für realitätsnahe Prototypen sind eine frühe Vorabversion einer Smartwatch (Vorserienprototyp), eine vollständig kodierte Smartphone-Anwendung oder ein vollständig visuell gestaltetes digitales Erlebnis (siehe [McEI2017]).

Manchmal wird auch der Begriff **Mid-Fidelity** zur Kategorisierung von Prototypen verwendet. Wie der Name schon nahelegt, liegen Mid-Fidelity-Prototypen in vielerlei Hinsicht zwischen Low-Fidelity- und High-Fidelity-Prototypen. Mid-Fidelity-Prototypen sehen der endgültigen digitalen Lösung ähnlicher als Low-Fidelity-Prototypen, sind aber immer noch einfacher als High-Fidelity-Prototypen. Mid-Fidelity-Prototypen verwenden mehr oder bessere visuelle Designs, mehr Interaktionspfade und mehr Funktionen als Low-Fidelity-Prototypen. Solche Prototypen können bereits im endgültigen Medium sein, d. h. im gleichen Material wie die endgültige digitale Lösung. Dies bedeutet natürlich einen höheren Aufwand als bei einem Low-Fidelity-Prototyp, aber die Zielgruppe, z. B. Benutzer oder Kunden, erhält mehr Kontext für die Bewertung. Kosten und Zeitaufwand können im Vergleich zum maximalen Nutzen des Prototyps optimiert werden. Beispiele für Mid-Fidelity-Prototypen sind Klick-Prototypen auf Zielgeräten, Style Tiles, einfache in Software programmierte Prototypen und einfache elektronische Prototypen (siehe [McEI2017]).

Diese Kategorisierung bietet eine grobe Einteilung des allgemeinen Detailgrads von Prototypen. Obwohl die Stufen *niedrig*, *mittel* und *hoch* für die meisten Zwecke ausreichen, ist es möglich, diese Skala zu verfeinern, z. B. durch die Verwendung einer kontinuierlichen, auf Zahlen basierenden Skala (z. B. zwischen eins und zehn), um den Fidelity-Grad noch genauer zu kategorisieren.

2.3.4.4 Detailgrad für jede Dimension eines Prototyps (Mixed-Fidelity-Prototypen)

Abschnitt 2.3.4.3 befasst sich mit dem (allgemeinen) Detailgrad des vollständigen Prototyps. Diese Kategorisierung hilft, das allgemeine theoretische Konzept von Low-Fidelity und High-Fidelity zu verstehen und Prototypen auf einer groben Ebene zu klassifizieren. In der Praxis ist es jedoch oft notwendig, den Detailgrad für bestimmte Dimensionen des Prototyps getrennt zu betrachten. Dies gilt insbesondere für größere Projekte und für die Arbeit in Iterationen.

Zu diesem Zweck wird vorgeschlagen, den Detailgrad für die folgenden fünf Dimensionen eines Prototyps zu betrachten: (1) sensorische Vollkommenheit, (2) Funktionsbreite, (3) Funktionstiefe, (4) Interaktionsumfang und (5) Datenmodellumfang. Dies ist eine Erweiterung des Modells von McCurdy et al. [MCPKV2006], die die *visuelle Verfeinerung* als erste Dimension betrachten. Dieses Modell erweitert diese Idee um weitere sensorische Fähigkeiten, wie z. B. den Hörsinn, den Tastsinn oder den Gleichgewichtssinn.

Die Dimensionen sensorische Vollkommenheit, Funktionsbreite und Funktionstiefe gehören zum Umfang des DDP Foundation Level, während die letzten beiden Dimensionen - Interaktionsumfang und Datenmodellumfang - über diese Stufe hinausgehen.

Ein Prototyp kann für jede dieser Dimensionen, die das Fidelity-Profil eines Prototyps definieren, verschiedene Fidelity-Stufen (auf einer Skala von niedrig bis hoch) aufweisen. Wie bereits in Abschnitt 2.3.4.3 erwähnt, sind die Fidelity-Stufen niedrig, mittel und hoch für die meisten Zwecke ausreichend. Es ist jedoch auch hier eine kontinuierliche Skala, z. B. von eins bis zehn, möglich, um eine feinere Kategorisierung zu erreichen, falls dies notwendig erscheint.

Sensorische Vollkommenheit (sensory refinement)

Der Detailgrad der Dimension *sensorische Vollkommenheit* bezieht sich auf die Qualität der sensorischen Darstellung des Prototyps, etwa die visuelle Darstellung, die auditive Verfeinerung oder die Qualität des haptischen Outputs. Eine Low-Fidelity-Darstellung könnte eine handgezeichnete Skizze der User Interface Elemente sein, z. B. einfache kastenförmige Wireframes. Zu den High-Fidelity-Produkten gehören pixelgenaue Visualisierungen oder zumindest hochwertige Visualisierungen des User Interface, die der endgültigen digitalen Lösung sehr nahe kommen.

Beispiel YPRC. Einer der ersten Prototypen, die im Rahmen der YPRC-Fallstudie erstellt wurden, ist der Prototyp für die Trainerperspektive. Das Ziel dieses Prototyps ist es, zu untersuchen, ob es möglich ist, den Läufer über eine Sprachverbindung zu coachen. Tabelle 13 zeigt das Fidelity-Profil dieses Prototyps.

Tabelle 13 - Fidelity-Profil des YPRC-Prototyps für die Perspektive des Coaches in Bezug auf die fünf Dimensionen eines Prototyps

Sensorische Vollkommenheit	Funktionsbreite	Funktions-tiefe	Interaktions-umfang	Datenmodell-umfang
niedrig	niedrig	hoch	mittel	niedrig

Abbildung 41 stellt verschiedene Stufen der sensorischen Vollkommenheit anhand der visuellen Ausarbeitung eines Bildschirms aus der YPRC-Fallstudie vor. Bild a) in dieser Abbildung zeigt einen handgezeichneten Bildschirm eines Papierprototyps als Beispiel für ein qualitativ geringwertiges Bildmaterial. Die Bilder b) und c) zeigen einen Wireframe Prototyp von geringer und mittlerer visueller Qualität. Bild d) ist eine hochwertige Grafik, die zu einer realitätsnahen sensorischen Vollkommenheit beitragen kann.

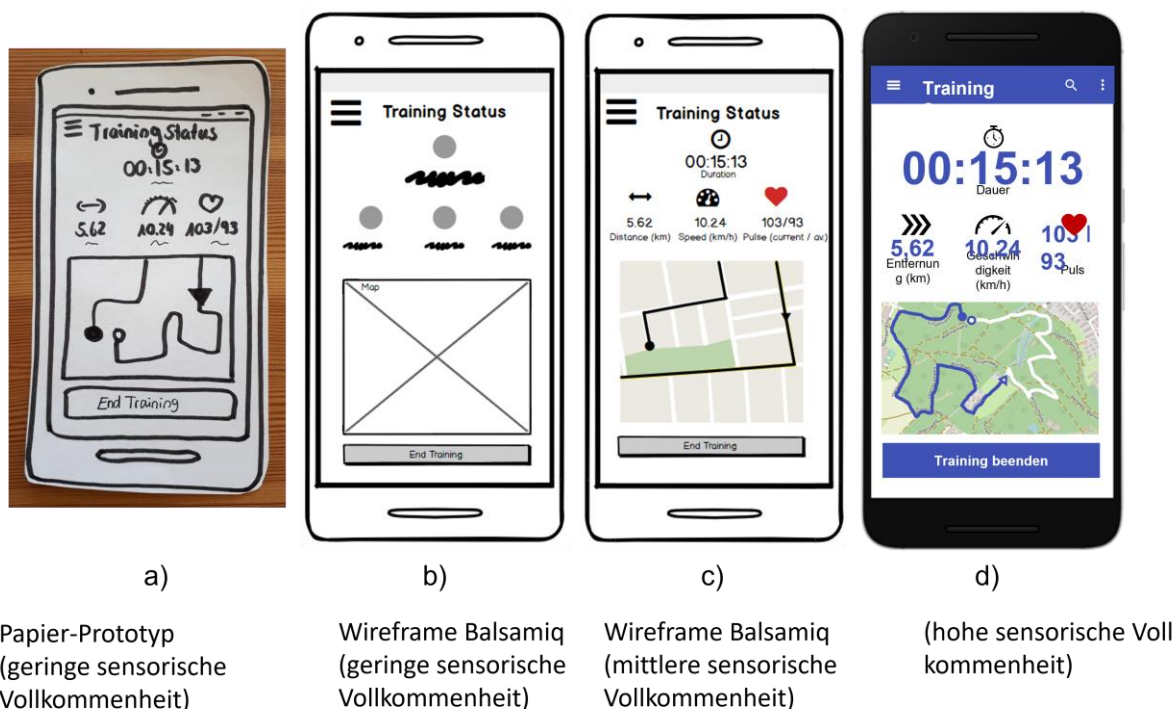


Abbildung 41 - Ein Beispiel eines Bildschirms aus den Prototypen der YPRC-Fallstudie, das verschiedene visuelle Qualitätsstufen zeigt, die zu unterschiedlichen Realitätsgraden der Dimension der sensorischen Vollkommenheit führen.

Karte im Bild ganz rechts: © OpenStreetMap-Mitwirkende, Lizenz: www.openstreetmap.org/copyright.

Funktionsbreite (breadth of functionality)

Der Realitätsgrad der Dimension *Funktionsbreite* bezieht sich auf die Anzahl der Funktionen, die im Prototyp verfügbar sind.

Ein Prototyp für die YPRC-Fallstudie mit einer hohen Funktionsbreite verfügt über alle geplanten Funktionen, wie z. B. die Verbindung mit der Smartwatch, dem Kopfhörer und dem Portal, die Überwachung des Trainingsstatus, Anrufe an den Trainingsassistenten und das Hochladen der Trainingsdaten.

Funktionstiefe (depth of functionality)

Der Realitätsgrad der Dimension *Funktionstiefe* definiert den Detaillierungsgrad eines im Prototyp verfügbaren Merkmals.

In der YPRC-Fallstudie könnte eine simulierte Anruffunktion für den Trainingsassistenten, die durch ein einfaches Bild auf dem Bildschirm auf einen bestehenden Anruf hinweist, eine wenig realitätsnahe Funktionstiefe darstellen. Im Gegensatz dazu könnte eine hohe Funktionstiefe alle Funktionen zur Kontaktaufnahme mit dem Assistenten bieten, wie z. B. das Herstellen eines Gesprächs mit einer Person, deren Stimme zuvor aufgenommen wurde, das Hochladen von

Trainingsdaten und das Erhalten von Trainingsempfehlungen. Dieses Beispiel befasst sich hauptsächlich mit den wahrnehmbaren Funktionen der digitalen Lösung und könnte nur mit Dummy-Daten funktionieren. Eine andere Form einer hohen Funktionstiefe für die YPRC-Fallstudie könnte ein voll funktionsfähiges Empfehlungssystem sein, das eine voll funktionsfähige künstliche Intelligenz (KI) einsetzt, die Daten auf der Grundlage der vom Benutzer übermittelten tatsächlichen Trainingsdaten vorschlägt. Dies würde (auch) die zugrundeliegenden Teile der digitalen Lösung betreffen. Ein weiteres Beispiel aus der YPRC-Fallstudie ist das Prototyping-Projekt, bei dem ein bestehendes Fitness-Tracker-Gerät mit einer Open-Source-Software-Schnittstelle Trainingsdaten wie die Pulsfrequenz liefert. Als funktionaler Prototyp wird eine Smartphone-App programmiert, um diese Daten zu sammeln, und eine andere PC-App wird programmiert, um diese Daten zu empfangen und sie als Text auf der Konsole der PC-Entwicklungsumgebung anzuzeigen. Obwohl der Grad der sensorischen Vollkommenheit gering ist, deckt er die gesamte Datenübertragungskette vom Fitness-Tracker zum PC ab. Sie befasst sich daher mit der gesamten zugrundeliegenden Funktion, um die Machbarkeit dieser speziellen Funktion für die endgültige Lösung zu bestätigen. Für dieses Beispiel kann die Funktionstiefe als hoch bezeichnet werden.

Horizontale Prototypen haben eine High-Fidelity Funktionsbreite und eine Low-Fidelity Funktionstiefe, während vertikale Prototypen eine Low-Fidelity Funktionsbreite und eine High-Fidelity Funktionstiefe (mindestens einer Funktionalität) haben. Folglich wird die in Abschnitt 2.3.4.1 vorgestellte Kategorisierung von Prototypen in horizontale und vertikale Prototypen (Interaktionsebene) durch die in diesem Abschnitt vorgestellte Kategorisierung nach dem Detailgrad der Dimensionen abgedeckt.

Interaktionsumfang (richness of interactivity)

Der Detailgrad der Dimension *Interaktionsumfang* bezieht sich auf die simulierte Präzision der interaktiven Elemente - zum Beispiel, wie genau der Übergang von einer Bildschirmseite zur anderen implementiert ist und wie nahe die Reaktionszeiten auf Benutzereingaben an der endgültigen digitalen Lösung liegen.

In der YPRC-Fallstudie würde ein Low-Fidelity-Prototyp in dieser Dimension abrupt von einer Bildschirmseite zur nächsten wechseln. Ein äußerst niedriger Detailgrad liegt vor, wenn der Übergang von einer handgezeichneten Bildschirmseite eines Papierprototyps, während einer Benutzertest-Session mit diesem Papierprototypen, zur nächsten handgezeichneten Bildschirmseite manuell von einem Moderator durchgeführt wird. Ein hoher Detailgrad wäre ein flüssig animierter Übergang von einer Bildschirmseite zur nächsten in einem in Software programmierten Prototyp.

Datenmodellumfang (richness of data model)

Der Detailgrad in der Dimension *Datenmodellumfang* drückt aus, wie repräsentativ die im Prototyp verwendeten Daten für die tatsächliche Zieldomäne sind.

Beispielsweise können mögliche Probleme beim Scrollen in einer Liste von Trainingsdaten der YPRC-Fallstudie nur untersucht werden, wenn der verwendete Datensatz groß genug ist. Einfache Datenmodelle mit einem oder zwei Beispiel-Trainingsdatensätzen - wie sie üblicherweise in Low-Fidelity- oder Mid-Fidelity-Prototypen verwendet werden - ermöglichen zwar das Testen verschiedener Aspekte der Liste, wie z. B. die Lesbarkeit der Zeichen oder die Platzierung der Bilder der Laufstrecke, lassen aber keine Probleme beim Scrollen in einer Liste mit 100 Einträgen auf einem Touchscreen erkennen. Ein hoher Detailgrad in dieser Dimension

würde daher eine große Anzahl von Trainingsdatensätzen mit einer hohen Variation der einzelnen Dateneinträge erfordern.

Praxistipp

Die Wahl des Detailgrads für jede Dimension ermöglicht es sich mit Hilfe des Prototyps auf bestimmte Aspekte zu konzentrieren, die vom Benutzer getestet oder auf ihre Machbarkeit hin untersucht werden sollen. Dies ist in der Praxis äußerst nützlich, um Prototypen bewusst an die tatsächlichen Prototyping-Ziele anzupassen. Wenn der Detailgrad zwischen diesen Dimensionen unterschiedlich ist, nennen McCurdy et al. [MCPKV2006] einen solchen Prototyp einen *Mixed-Fidelity-Prototyp*.

Es gibt Überschneidungen zwischen dem Kriterium Interaktionsebene (siehe Abschnitt 2.3.4.1), dem Detailgrad für jede Dimension eines Prototyps und dem gesamten Detailgrad (siehe Abschnitt 2.3.4.3). Wie bereits erwähnt, wird die Kategorisierung nach der Interaktionsebene bei horizontalen und vertikalen Prototypen (siehe Abschnitt 2.3.4.1) durch die Kategorisierung nach den Dimensionen eines Prototyps (Funktionstiefe etc.) abgedeckt. Die Klassifizierung anhand der fünf Dimensionen eines Prototyps umfasst auch die Kategorisierung nach dem (allgemeinen) Detailgrad. Daher ist die Verwendung von Fidelity-Profilen auf der Grundlage der fünf Dimensionen eines Prototyps präziser und umfasst die beiden anderen Kategorisierungen.

Alle in diesem Handbuch vorgestellten Ansätze zur Kategorisierung von Prototypen werden durch die Schemata des explorativen, experimentellen und evolutionären Prototyping (siehe Abschnitt 2.3.4.2) und die in diesem Abschnitt vorgestellte Betrachtung von Fidelity-Profilen auf der Grundlage der fünf Dimensionen eines Prototyps abgedeckt.

2.3.4.5 Fidelity-Profile von Prototypen, die typischerweise während des Bauprozesses verwendet werden

Wie in Abschnitt 2.3.1 erläutert, haben verschiedene Disziplinen ein unterschiedliches Verständnis davon, was ein Prototyp ist. Daher können Prototypbezeichnungen auch in verschiedenen Disziplinen unterschiedlich verwendet werden oder sogar innerhalb einer Disziplin unterschiedliche Bedeutungen haben. Daher werden wichtige Prototypnamen, für die Verwendung durch den DPP vorgeschlagen und in Tabelle 14 durch ihr Fidelity-Profil nach dem Verständnis und der Erfahrung der Autoren dieses Handbuchs charakterisiert. Die linke Spalte von Tabelle 14 zeigt Prototypennamen, die typischerweise im Digital Design verwendet werden.

Die folgende Liste enthält eine kurze Beschreibung der einzelnen Prototypen, die unter Tabelle 14 aufgeführt sind:

- Ein **Appearance-Prototyp** stellt das endgültige Erscheinungsbild des Produkts dar, bietet aber keine Funktionalität. Sie wird in der Regel verwendet, um das visuelle Erscheinungsbild des Produkts bei Kunden oder Benutzern zu validieren.
- Ein **Experience-Prototyp** stellt das endgültige Erscheinungsbild des Produkts dar und bietet nahezu echte Interaktivität.
- Eine **Skizze** ist eine einfache handgefertigte Zeichnung (eines Aspekts) der angestrebten digitalen Lösung.

- Ein **Paper-Prototyp** besteht in der Regel aus handgezeichneten Bildschirmseiten des Zielsdisplays auf Papier und einer Beschreibung der Abfolge zwischen diesen Bildschirmseiten - zum Beispiel, was passiert, wenn der Benutzer eine bestimmte Taste drückt. In Abschnitt 2.3.6 finden Sie Einzelheiten zu diesem leistungsstarken Werkzeug für den DDP.
- Ein **Cardboard-Prototyp** kann als eine Erweiterung eines Papierprototyps in die dritte Dimension betrachtet werden. Es handelt sich um Karton oder ähnliches Material, das die Form eines physischen Produkts darstellt.
- Ein **Wireframe-Prototyp**, in der Regel nur Wireframe genannt, besteht aus Skizzen von Bildschirmen in einem digitalen Format, die das Layout der jeweiligen Bildschirme zeigen, d. h. wo sich Textfelder, Schaltflächen, Bilder oder Eingabefelder auf dem Bildschirm befinden. In der Regel werden keine echten Bilder und kein echter Text gezeigt.
- Ein **Klick-Prototyp** bzw. klickbarer Prototyp oder interaktives Mock-up zeigt (handgezeichnete oder digital erstellte) Bildschirmseiten, die angeklickt werden können. Nach einem Klick wird eine andere Bildschirmseite angezeigt. In der Regel ist die Interaktivität begrenzt, und es wird ein spezielles Programm zur Erstellung benötigt (vgl. Abschnitt 2.3.5.3).
- Ein **Mock-up** ist ein einfaches Modell der physischen Form eines (neuen) Geräts, das für eine digitale Lösung verwendet wird, und in der Regel in einem anderen Medium wie Knetmasse, Schaumstoff, Holz oder einem Material für die additive Fertigung (z. B. 3D-Druck, siehe Abschnitt 2.3.5.2) erstellt wird. Ein Mock-up, das nicht die Zielmaße hat, wird als skaliertes Mock-up bezeichnet.
- Ein **funktionaler Prototyp** enthält alle relevanten Elemente zur Realisierung einer bestimmten Funktion, hat aber nicht die endgültige Form des Produkts. Er wird in der Regel erstellt, um die jeweilige zugrundeliegende Funktion einer digitalen Lösung zu validieren.
- Ein **Coded-Prototyp** ist eine prototypische Softwareimplementierung des Softwareteils der digitalen Lösung (vgl. Abschnitt 2.3.5.1).
- Ein **Storyboard** ist eine handgezeichnete oder digital erstellte Abfolge von Bildern zur Visualisierung einer beispielhaften Storyline, die einen ersten Eindruck für die digitale Lösung vermittelt.
- Eine **Story Map** ist eine zweidimensionale Visualisierung einer Abfolge von (textuellen) User Storys. Sie beschreibt den narrativen Fluss einer digitalen Lösung oder den Gesamtprozess, der durch die digitale Lösung bereitgestellt wird. Es handelt sich nicht nur um eine Form des Prototyps, sondern auch um ein Managementinstrument (siehe Kapitel 5).
- Ein **Video-Prototyp** oder **fiktives Video** visualisiert die (zukünftige) interaktive Nutzung einer digitalen Lösung in Form eines Videos. Der Zweck kann sein, (1) die interaktive Nutzung zu zeigen, indem den Stakeholdern bestimmte Funktionen oder Nutzungsszenarien vorgeführt werden, oder (2) die Stakeholder zu überzeugen. Im letzteren Fall ist der Prototyp eher ein Werbevideo oder ein Prototyp für ein Werbevideo.

- Ein **Interface-Prototyp** (Schnittstellenprototyp) ist eine andere Bezeichnung für einen horizontalen Prototyp (siehe Abschnitt 2.3.4.1).
- Ein **Future-Press-Release** (Pressemitteilung aus der Zukunft) ist ein möglicher Artikel in einer zukünftigen Ausgabe einer Zeitung oder eines Nachrichtenportals. Sie erzählt eine emotionale Zukunftsgeschichte über die digitale Lösung, die in ihrem Anwendungskontext vorgesehen ist.
- Ein **High-Fidelity Mock-up** ist eine andere Bezeichnung für einen High-Fidelity-Prototyp (siehe Abschnitt 2.3.4.3).
- Ein **Vorserienprototyp** ist ein vollständig realisiertes Muster des Produkts. Es wird in der Regel für abschließende und umfassende Qualitätssicherungsmaßnahmen verwendet.
- Ein **Solution-Prototyp** (Lösungsprototyp) ist eine andere Bezeichnung für einen Mid-Fidelity- oder High-Fidelity-Prototyp (siehe Abschnitt 2.3.4.3), je nachdem, wie nahe er an der endgültigen digitalen Lösung ist.

Tabelle 14 - Fidelity-Profilbereich typischer Prototypen im Digital Design

Typische Namen von Prototypen	Sensorische Vollkommenheit	Funktionsbreite	Funktions-tiefe	Interaktionsumfang	Datenmode llumfang
Appearance-Prototyp	hoch	niedrig-hoch	niedrig	niedrig	niedrig
Experience-Prototyp	hoch	niedrig-hoch	mittel-hoch	mittel-hoch	mittel
Skizze	niedrig	niedrig	niedrig	niedrig	niedrig
Paper-Prototyp	niedrig	niedrig-hoch	niedrig-hoch	niedrig-mittel	niedrig
Cardboard-Prototyp	niedrig	niedrig-hoch	niedrig-hoch	niedrig-mittel	niedrig
Wireframe	niedrig-mittel	niedrig-hoch	niedrig-hoch	niedrig-mittel	niedrig
Klick-Prototyp	niedrig-hoch	niedrig-hoch	niedrig-hoch	niedrig-mittel	niedrig
Mock-up	niedrig-hoch	niedrig-mittel	niedrig-mittel	niedrig	niedrig
Funktionaler Prototyp	niedrig	niedrig	hoch	niedrig	mittel-hoch
Coded-Prototyp	niedrig-hoch	niedrig-hoch	niedrig-hoch	niedrig-hoch	niedrig-hoch
Storyboard	niedrig	niedrig	niedrig-mittel	k.A. ⁹	k.A.
Story Map	niedrig	niedrig	niedrig-mittel	k.A.	k.A.
Videoprototyp	mittel-hoch	niedrig-mittel	niedrig-mittel	k.A.	k.A.
Future-Press-Release	niedrig	niedrig	niedrig	k.A.	k.A.
Schnittstellenprototyp	mittel-hoch	hoch	niedrig-mittel	mittel-hoch	niedrig-hoch
High-Fidelity Mock-Up	hoch	hoch	hoch	niedrig-hoch	niedrig-hoch
Vorserienprototyp	hoch	hoch	hoch	hoch	hoch
Solution-Prototyp	mittel-hoch	mittel-hoch	mittel-hoch	mittel-hoch	mittel-hoch

⁹ Die Abkürzung *k.A.* bedeutet *keine Anwendung* und weist darauf hin, dass diese Kategorie bei diesem Prototyp nicht berücksichtigt wird.

2.3.4.6 Prototyp-Kategorien in bestimmten Phasen des Bauprozesses

Das **explorative Prototyping** (siehe Abschnitt 2.3.4.2) ist in allen Phasen des Bauprozesses nützlich, wenn alternative Ideen, Konzepte oder Lösungen in Betracht gezogen werden. Solche Prototypen werden in der Regel nicht wiederverwendet und sind oft horizontale Prototypen.

Das **experimentelle Prototyping** (siehe Abschnitt 2.3.4.2) ist während der Scoping- und Konzeptionsphase sinnvoll. Solche Prototypen ergänzen oder verfeinern Konzepte und können einen Zwischenschritt im Prototyping zwischen Design, Konstruktion und Realisierung darstellen. Daher ist diese Art von Prototyping besonders in den Konsolidierungsphasen der Design- und Konstruktionsaktivitäten und während der Realisierungsphase des Bauprozesses nützlich. Dieser Typ ist in der Regel ein vertikaler Wegwerfprototyp.

Evolutionäre Prototypen (siehe Abschnitt 2.3.4.2) werden ausgearbeitet und verfeinert, bis sie das Zielsystem darstellen. Sie sind also während aller Schritte des Bauprozesses präsent, wenn ein solcher Prototyping-Ansatz verwendet wird.

Der Detailgrad der Prototypen insgesamt (siehe Abschnitt 2.3.4.3) nimmt mit dem Fortschritt bei der Erstellung der digitalen Lösung zu. Zu Beginn des Bauprozesses ist es nützlich, viele Low-Fidelity-Prototypen zu haben. Sie werden im Laufe des Projekts immer detaillierter (High-Fidelity) und in ihrer Anzahl geringer. Prototypen, die zwischen Low-Fidelity und High-Fidelity liegen, sind für geschäftliche Entscheidungen nützlich. Es besteht jedoch die Gefahr, dass der Detailgrad für einen bestimmten Prototypen zu hoch gewählt wird. In diesem Fall könnten die Stakeholder eine ausgereifere digitale Lösung vermuten, als tatsächlich vorhanden ist. Daher muss der DDP den richtigen Detailgrad sorgfältig auswählen, um keine falschen Erwartungen zu wecken. High-Fidelity-Prototypen - die häufiger in späteren Stadien des Bauprozesses eingesetzt werden - sind nützlich, um Details zu testen, um die vollständige digitale Lösung zu erleben oder um als Schnittstellen-Spezifikation für Softwareentwickler oder Herstellern zu dienen (siehe [McEl2017]).

Abbildung 42 veranschaulicht eine typische Verwendung von Prototypenkategorien während der Laufzeit eines Projekts zum Aufbau einer digitalen Lösung. Die steigende Tendenz bei explorativem und experimentellem Prototyping zeigt eine zunehmende Nutzung dieser Prototyping-Ansätze, bevor die Nutzung einen Höhepunkt erreicht. Die tatsächliche Verwendung hängt jedoch sehr stark von der angestrebten digitalen Lösung und dem Projekt ab. Ähnliche Muster der Verwendung von Prototypen können auch während der Design-, Konstruktions- und Realisierungsaktivitäten des Bauprozesses auftreten.

Zusätzlich zu diesen allgemeinen Prototyp-Nutzungsprofilen erfordern bestimmte Teilbereiche der meisten Projekte die Erprobung einer bestimmten Lösung, selbst in einem späten Projektstadium. So kann es beispielsweise notwendig sein, alternative Prototypen zu erstellen, die verschiedene visuelle Design-Themes zeigen, um eine Entscheidung für das beste Design vorzubereiten. Die Erhebungen und Spitzen in der oberen dreieckigen Form in Abbildung 42 stellen solche Aktivitäten dar.

Möglicherweise muss eine bestimmte kritische Funktion der digitalen Lösung prototypisch implementiert werden, um das Risiko eines Totalausfalls zu minimieren, der zu einem sogenannten *Showstopper* führt. Nehmen wir als beispielsweise einen KI-Algorithmus für einen Sprachausgabe-Assistenten, der auf dem kleinen Gerät, für das eine digitale Lösung geplant ist, leistungskritisch ist. Um sicherzustellen, dass dieser Algorithmus schnell genug notwendige Ergebnisse liefert, und um den möglichen Optimierungsbedarf zu ermitteln, kann ein

experimenteller funktionaler Prototyp erstellt werden. Diese Tätigkeit sollte so früh wie möglich im Projekt durchgeführt werden, um die Arbeit mit einem geringeren Risiko fortsetzen zu können.

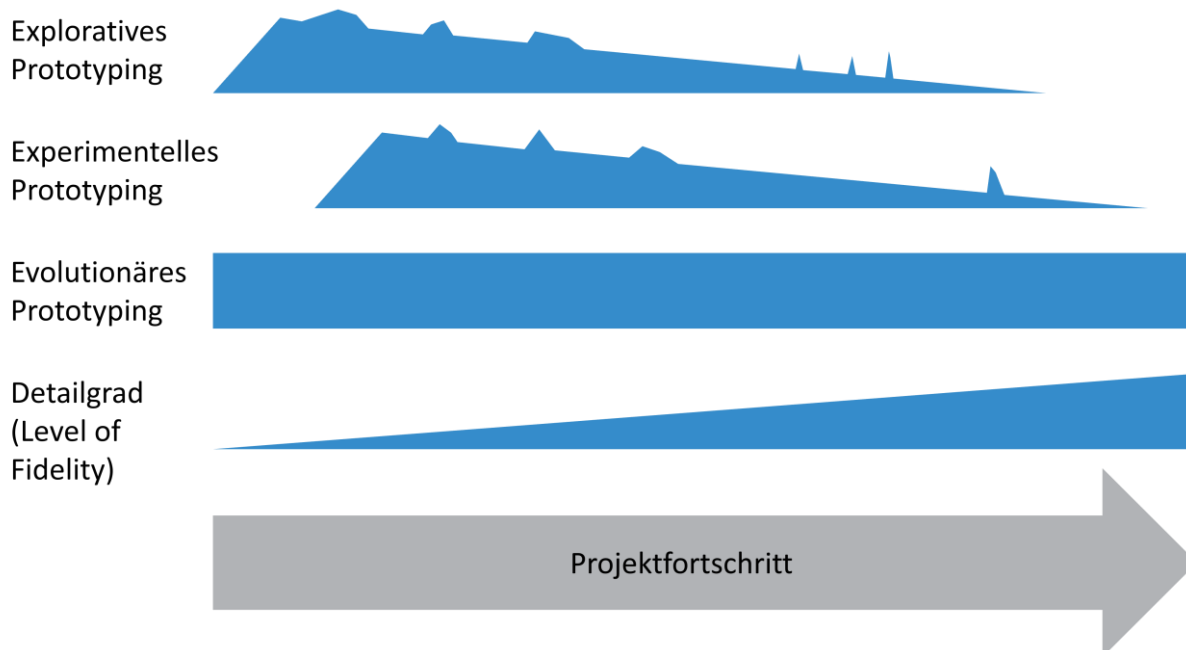


Abbildung 42 - Typische Verwendung von Prototypen (Aktivitäten) im Rahmen eines Digital Design Projekts

In einem anderen Beispiel wird zum Abschluss der Realisierungsaktivität ein experimenteller funktionaler Prototyp der Zahlungsfunktion für eine digitale Lösung erstellt, um zu testen, ob der Abrechnungsprozess im Detail zusammen mit anderen Systemen funktioniert, zu denen die digitale Lösung eine Schnittstelle bilden muss. Diese Art von prototypischen Aktivitäten stellen die Erhebungen und Spitzen in der zweiten Dreiecksform von Abbildung 42 dar.

Im Allgemeinen sind Mixed-Fidelity-Prototypen (siehe 2.3.4.4) sehr nützlich, um sich auf einzelne Aspekte des konzeptuellen Designs, des technischen Designs oder der realisierten Lösung zu konzentrieren, wie die drei in den vorherigen Abschnitten beschriebenen Beispiele. Daher sind sie in der Mitte des Bauprozesses am hilfreichsten. Es ist nicht möglich, einen allgemeinen Leitfaden für die Definition des *richtigen* Detailgrads der verschiedenen Dimensionen eines Prototyps, d. h. des idealen Fidelity-Profiles in einer bestimmten Projektphase oder Abstraktionsebene, zu geben. Dies liegt daran, dass das geeignete Fidelity-Profil von der spezifischen Zielsetzung innerhalb der jeweiligen Projektsituation abhängt, für die der Prototyp erstellt wird - beispielsweise von den Evaluationszielen eines Usability-Tests. Nichtsdestotrotz zeigen die Beispiele die hohe Flexibilität und Nützlichkeit dieses Kategorisierungsansatzes unter Verwendung von Fidelity-Profilen für die verschiedenen Dimensionen eines Prototyps.

2.3.4.7 Grad der Immersion in Bezug auf das Fidelity-Profil von Prototypen

Technische Systeme zur Erzeugung einer virtuellen Realität (VR) bieten einen gewissen Grad an *Immersion*¹⁰, der die technische Voraussetzung dafür ist, den psychologischen Zustand der *Präsenz* für den Benutzer zu ermöglichen. Präsenz bezeichnet das Gefühl einer Person in der Rolle eines Benutzers, in der virtuellen Umgebung *angekommen zu sein*, während diese Person

¹⁰ Nach [Jera2016] (zitiert nach [SIWi1997]) ist „Immersion der objektive Grad, in dem ein VR-System und eine VR-Anwendung Reize auf die Sinnesrezeptoren der Benutzer in einer Weise projiziert, die umfassend, passend, umgebend, lebendig, interaktiv und handlungskonform ist.“

die Tatsache vergisst, sich gleichzeitig physisch noch in einer anderen (der realen) Umgebung zu befinden.

Die Autoren dieses Handbuchs sind der Meinung, dass das Konzept der Immersion und Präsenz aus der virtuellen Realität für die Anwendung von Prototypen im Digital Design nützlich ist. Je nach den geplanten Zielen, die mit einem Prototyp erreicht werden sollen, kann ein bestimmter Grad der Immersion angestrebt werden. Wenn beispielsweise ein Designerkollege schnelles Feedback zu einem bestimmten Benutzerinteraktionsdetail für eine digitale Lösung geben soll, ist ein geringerer Immersionsgrad ausreichend als beispielsweise bei einem Prototyp, der für eine Kundenpräsentation oder eine umfangreiche Usability-Untersuchung verwendet werden soll.

Die folgende Beschreibung der Immersionselemente stammt aus dem VR-Bereich (vgl. [Jera2016], [SIWi1997]) und wurde für die Verwendung im Digital Design angepasst. Die wichtigsten Anpassungen wurden im Hinblick auf Anschaulichkeit, Interaktivität und Handlung vorgenommen.

- Der **Umfang** (Extensiveness) gibt die Anzahl der sensorischen Modalitäten an, die dem Benutzer präsentiert werden, z. B. wenn visuelle Ausgabe, auditive Ereignisse und haptische Stimulation durch physische Kräfte vorhanden sind.
- **Übereinstimmung** (Matching) beschreibt den Grad, in dem verschiedene Modalitäten zusammenpassen und dem Benutzer ein konsistentes Feedback vermitteln - zum Beispiel, wie gut die Vibration des Smartphones zeitlich zum Drücken der virtuellen Taste passt.
- **Umgebenheit** (Surroundedness) beschreibt, wie umfassend die Merkmale sind - beispielsweise ein großer Bildschirm, eine räumliche Audioausgabe oder ein großer Bereich, in dem Gesten erkannt werden, die zu diesem Element beitragen.
- **Lebendigkeit** (Vividness) bezeichnet die Fähigkeit, eine sensorisch reichhaltig vermittelte Umgebung zu erzeugen, die beispielsweise durch die Auflösung des Bildschirms, die Helligkeit und den Farbbereich sowie die Bild- oder Bitrate gekennzeichnet ist (vgl. [Steu1992]).
- **Interaktivität** (Interactability) ist die Fähigkeit, den Prototyp zu beeinflussen und das Ausmaß der Reaktionen des Prototyps auf der Grundlage der Aktionen des Benutzers.
- **Handlungskonform** beschreibt, wie gut ein Nutzungsszenario in der digitalen Lösung implementiert ist. Sie geht über einen einzelnen Interaktionsablauf hinaus und wird daher nicht vollständig von der Interaktivität abgedeckt. Ein Beispiel aus der YPRC-Fallstudie: Der Läufer muss ein Gerät zur Messung der Herzfrequenz anbringen und die Smartwatch und das Smartphone einrichten, bevor der Lauf beginnen kann. Nach einer Reihe von Läufen - zum Beispiel mehrere Läufe innerhalb einer Woche - kann der Läufer die Laufstatistiken auf dem Smartphone oder dem heimischen PC auswerten. Die Einbeziehung der Funktion eines Trainers mit menschlicher oder künstlicher Intelligenz würde das Kontext-Szenario und damit die Handlung komplexer machen.

Tabelle 15 stellt die Beziehung zwischen den Immersionselementen und den fünf Dimensionen eines Prototyps dar (siehe Abschnitt 2.3.4.5). Diese Zuordnung kann verwendet werden, um wichtige Dimensionen eines Prototyps zu identifizieren, um so den gewünschten Grad der Immersion zu erreichen. Eine starke Verbindung (gekennzeichnet durch ein „X“) zwischen einer Prototyp-Dimension und einem bestimmten Immersions-Element zeigt an, wie der Realitätsgrad dieser Prototyp-Dimension das entsprechende Immersions-Element und damit den Gesamt-

Immersion-Grad beeinflusst. Die Wahl eines Prototyps mit einem bestimmten Fidelity-Profil kann den Schwerpunkt auf bestimmte Immersionselemente legen.

Tabelle 15 - Beziehung zwischen den Dimensionen des Prototyps und den Immersionselementen. „X“ steht für eine starke und „~“ für eine schwache Beziehung.

Immersionselemente	Dimensionen eines Prototyps (Fidelity-Profil)				
	Sensorische Vollkommenheit	Funktionsbreite	Funktions-tiefe	Interaktionsumfang	Datenmodellumfang
Umfang	X				
Übereinstimmung	X	~	X		
Umgebenheit	X	~			
Lebendigkeit	X				
Interaktivität		X	X	X	X
Handlungskonformität				X	X

2.3.5 Werkzeuge zur Erstellung von Prototypen

Um in der frühen Phase von Digital Design-Projekten einen großen Lösungsraum für die Erkundung (grundsätzlich) unterschiedlicher Lösungsansätze offen zu halten, sind einfache, aber leistungsfähige Prototyping-Technologien, wie z. B. das Paper Prototyping (siehe Abschnitt 2.3.6), oft die richtige Wahl.

Die Auswahl der am besten geeigneten Kategorie eines Prototyps (siehe Abschnitt 2.3.4) und des Werkzeugs zur Erstellung von Prototypen in einer bestimmten Projektsituation erfordert ein gewisses Maß an Erfahrung. Die Ziele der konkreten Untersuchung bestimmen die am besten passende Prototypenkategorie. Ein Einsteiger in diesem Bereich sollte mit einfachen Techniken beginnen, wie Skizzen, Papierprototypen oder Storyboards. Um unnötigen Aufwand zu vermeiden, sollten komplexere Technologien nur nach sorgfältiger Abwägung eingesetzt werden. Die folgenden Abschnitte geben einen Überblick über die breite Palette von Werkzeugen zur Erstellung von Prototypen, die für die Entwicklung digitaler Lösungen hilfreich sind.

2.3.5.1 Software-Design- und Entwicklungswerkzeuge und Technologien für die Erstellung von Prototypen

Standard-Software-Entwicklungsumgebungen können für die Erstellung von programmierten Prototypen verwendet werden, insbesondere wenn Prototypen mit tiefgreifender Funktionalität benötigt werden. Um einen kostengünstigen Prototyp zu erstellen, sind viele Vereinfachungen im Vergleich zur Zielentwicklungsumgebung möglich.

Webtechnologie als leistungsstarkes Werkzeug für die Erstellung von Prototypen

Webtechnologien wie HTML5, CSS und JavaScript sind nützlich, um schnell Prototypen zu erstellen, die mehr Funktionen bieten als Klick-Prototypen (siehe Abschnitt 2.3.5.3). Neben der Werkzeugunterstützung für die Erstellung von Webseiten und Webanwendungen gibt es auch Werkzeuge, die die automatische Erzeugung von Webanwendungen für verschiedene Smartphone-Betriebssysteme ermöglichen. Es existieren beispielsweise Werkzeuge, die mithilfe von Webtechnologien Anwendungen für Android und iOS durch eine einzige Codebasis erzeugen

können. Solche *hybriden Anwendungen*, wie sie manchmal genannt werden, könnten sogar eine Option für die endgültige digitale Lösung sein (siehe auch Abschnitt 3.2.3).

Wenn ein programmierter Prototyp benötigt wird, sind grafische User Interface-Builder (GUI-Builder), die Grafikbibliotheken des Zielbetriebssystems verwenden, hilfreiche Werkzeuge. Dies beschleunigt in der Regel die Entwicklung im Vergleich zur Implementierung eines vollständig angepassten User Interfaces bzw. einer User-Interface-Bibliothek.

Programmierung von Prototypen für komplexe oder tiefgreifende Funktionalität

Für eine komplexere oder umfassendere Funktionalität der Prototypen ist die Programmierung von Teilen der Softwareanwendung eine zusätzliche Option und manchmal auch notwendig. Um diesen Erstellungsprozess zu beschleunigen, ist eine einfachere Programmiersprache oder Entwicklungsumgebung als für die angestrebte digitale Lösung eine sinnvolle Option. Prototypische Programme bleiben einfach, indem sie Fehlerbehandlungsroutinen oder Sonderfälle des Programmkontrollflusses weglassen. Im Fachjargon der Softwareentwicklung wird diese Art der Vereinfachung manchmal als *Hack* bezeichnet.

Wenn eine digitale Lösung auf Standard-Hardware läuft, z. B. auf einem Smartphone, reicht es in der Regel aus, die oben beschriebenen Prototypen-Werkzeuge zu verwenden, um den Prototyp auf einer (Standard-)Bildschirmgröße und -ausrichtung laufen zu lassen. Wenn jedoch eine neue digitale Lösung einen neuen physischen Formfaktor oder eine neue Art von Hardware verwendet, sollten die endgültige Form und die Fähigkeiten des Geräts, auf dem die Software läuft (z. B. Bildschirmgröße, Rechenkapazität), berücksichtigt werden, um einen realistischen Prototyp zu erstellen.

Praxistipp

Für den DDP auf Foundation Level Niveau wird empfohlen, die in diesem Abschnitt vorgestellten Möglichkeiten des Prototypings zu berücksichtigen. Wenn tatsächlich ein programmierter Prototyp benötigt wird, muss ein erfahrener Experte für einen solchen Prototyp mit der Entwicklung des Prototyps beauftragt werden. Es ist sehr wichtig, dass dieser Experte erfahren und qualifiziert genug ist, um den Prototyp so weit wie möglich zu vereinfachen und keine weitere alternative Implementierung für die endgültige digitale Lösung zu schaffen. Der Aufwand für solche Prototypen muss so begrenzt sein, dass diese Arbeit verworfen werden kann, wenn der Prototyp seinen Zweck erfüllt hat.

Die in diesem Abschnitt vorgestellten Prototyping-Werkzeuge eignen sich für die Erstellung von Prototypen für die wahrnehmbaren und zugrundeliegenden Ebenen (siehe Abschnitt 1.2) des Softwareteils der digitalen Lösung.

2.3.5.2 Industriedesign-Werkzeuge für die Erstellung von Prototypen

Industriedesigner haben eine lange Tradition im Prototyping und verwenden Prototyping-Werkzeuge, um die Interaktion des Benutzers mit einem physischen Produkt zu gestalten und die Machbarkeit eines solchen Produkts zu untersuchen. Es können die folgenden drei Typen unterschieden werden (vgl. [IDSA2020]):

- Werkzeuge zum Erstellen von Skizzen und Illustrationen
- Werkzeuge für dreidimensionale Darstellungen (Rendering)
- Werkzeuge für die additive Fertigung von Tangible Mock-Ups

Zeichenwerkzeuge und Rendering-Software

Zeichenwerkzeuge werden verwendet, um Skizzen und Illustrationen zu erstellen, z. B. für Ideenskizzen, Studienskizzen, Referenzskizzen, Gedächtnisskizzen, kodierte Skizzen, Informationsskizzen, Sketch Renderings, perspektivische Skizzen, Szenarien, Storyboards, Schaubilder, perspektivische Darstellungen, Anordnungspläne, Detailzeichnungen und technische Illustrationen.

Rendering-Software wird verwendet, um dreidimensionale Renderings zu erstellen, z. B. Layout-Renderings und Präsentations-Renderings. Für anspruchsvolle dreidimensionale Illustrationen können Werkzeuge für computergestütztes Design (CAD) verwendet werden.

Tangible Mock-Ups und physische Prototypen

Der Bau von anfassbaren Mock-Ups und physischen Prototypen, oft auch Modellierung genannt, hat im Industriedesign eine lange Tradition, um Ideen zu erforschen und Produkte im physischen Raum zu optimieren. Die Untersuchung bestimmter Aspekte kann Appearance-Modelle, Montagemodelle, Produktionsmodelle, Servicemodelle, experimentelle Prototypen, Alpha- oder Beta-Prototypen, Systemprototypen, endgültige Hardware-Prototypen, werkfreie Komponenten (off-tool components), Appearance-Prototypen und Vorserienprototypen umfassen.

Um einen Eindruck über die Möglichkeiten dieser Art von Prototyping zu vermitteln, die über das Foundation Level hinausgeht, zeigt Abbildung 43 Beispielprototypen aus diesem Bereich.



Abbildung 43 - Beispielhafte Industriedesign-Prototypen (Copyright: Neuland/GENERATIONDESIGN)

Auf der linken Seite sind handgefertigte Skizzen (1) mit Markern und auf der rechten Seite Photoshop-Skizzen (2) zu sehen. Die Mitte der Abbildung zeigt Prototypen, die durch additive Fertigung (3) auf der Grundlage eines in Solidworks (CAD-Anwendung) erstellten CAD-Modells

hergestellt wurden. Für die additive Fertigung wurde das Fused-Filament-Verfahren eingesetzt. Dieses Verfahren ist auch unter dem geschützten Begriff *Fused Deposition Modeling* bekannt. Diese Prototypen werden verwendet, um die ergonomischen Eigenschaften des Stifts zu überprüfen. Der schwarze Vorserienprototyp (4) wurde vom Endhersteller im Spritzgussverfahren hergestellt. Der transparente Stift (5) ist das Endprodukt der Neuland FineOne® Empty Variante. Dreidimensionale Renderings sind im rechten und unteren Teil der Abbildung zu sehen. Diese werden für Kommunikationszwecke verwendet.

Industriedesigner verwenden manchmal den Begriff *Modell*, wenn ein Prototyp gemeint ist, während in der Softwaretechnik ein anderes Verständnis zum Begriff Modell vorherrscht (vgl. [IEEE2017]). Unter 2.3.1 befindet sich eine Definition des Begriffs Prototyp im Kontext des Digital Designs und eine Diskussion darüber, wie er sich zum Begriff Modell verhält.

Praxistipp

Für den DDP auf Foundation Level Niveau wird empfohlen, die Fähigkeiten zum Skizzieren und Illustrieren zu trainieren. Ein Grundverständnis für die Möglichkeiten des Prototyping mit dreidimensionalen Renderings und Tangible Mock-ups ist ausreichend. Wenn es tatsächlich notwendig ist, einen anspruchsvolleren Prototyp mit solchen Werkzeugen zu erstellen, muss ein erfahrener Experte hinzugezogen werden.

Die in diesem Abschnitt vorgestellten Prototyping-Werkzeuge eignen sich für die Erstellung von Prototypen für die wahrnehmbare Ebene (siehe Abschnitt 1.2) des physischen Teils der digitalen Lösung.

2.3.5.3 Interaktionsdesign-Werkzeuge für die Erstellung von Prototypen

Bei digitalen Lösungen spielen die Schnittstellen zu den Benutzern eine wichtige Rolle. Interaktionsdesign ist die Disziplin, die sich mit der Gestaltung von User Interfaces beschäftigt (vgl. [Coop2004]). Daher sind alle Interaktionsdesign-Werkzeuge für die Erstellung eines User Interface-Prototyps nützlich. Dazu gehören Zeichen-Werkzeuge für Skizzen, Storyboarding und Wireframing, Papier und Bleistift für das Prototyping auf Papier und Rendering-Software für hochwertige Renderings.

Praxistipp

Skizzen, Wireframe-Prototypen oder hochwertige Renderings von User Interfaces können zur Integration in einen klickbaren Prototyp verwendet werden. Viele Softwarewerkzeuge für die Erstellung solcher Prototypen ermöglichen die Verwendung des physischen Zielgeräts (Formfaktor), wie z. B. Smartphones, PCs oder Tablets. Dies macht den Prototyp realistischer, indem er dem Benutzer zum Beispiel verdeutlicht, wie klein der Bildschirm des Geräts tatsächlich ist. Es gibt viele Klick-Prototypen-Werkzeuge, die für verschiedene Arten von Prototypen geeignet sind. Zu den existierenden Produktnamen für solche Werkzeuge gehören Adobe XD, Axure, Balsamiq, Figma, Flinto, InVision, Mockups oder Sketch. Für den DDP auf Foundation Level wird jedoch empfohlen, sich auf einfache Werkzeuge zu beschränken, die vorzugsweise auf Papier und Bleistift basieren, oder einen Experten hinzuzuziehen, der Erfahrung mit diesen Werkzeugen hat.

Die in diesem Abschnitt vorgestellten Prototyping-Werkzeuge eignen sich für die Erstellung von Prototypen für die wahrnehmbare Ebene (siehe Abschnitt 1.2) des Softwareteils der digitalen Lösung.

2.3.5.4 Andere Werkzeuge

Darüber hinaus können auch Prototyping-Werkzeuge und -Technologien aus der Produktions- und Elektrotechnik zum Einsatz kommen, z. B. die Erstellung eines Geräteprototyps mittels additiver Fertigung (z. B. 3D-Druck) oder einer frühen Version einer kundenspezifischen Leiterplatte (Printed Circuit Board). Diese Prototyping-Werkzeuge sind nützlich für die Erstellung von Prototypen für die zugrundeliegende Ebene (siehe Abschnitt 1.2) des physischen und Hardware-Teils der digitalen Lösung.

2.3.6 Erstellung und Verwendung einfacher Low-Fidelity-Prototypen

Papierprototypen und Prototypen aus Pappe sind einfach zu bauen und praktisch für die Erstellung von Low-Fidelity-Prototypen (siehe 2.3.4.5). Ein Papierprototyp besteht in der Regel aus (1) handgezeichneten Bildschirmseiten des Zielbildschirms der digitalen Lösung und (2) einer Beschreibung der Abfolge dieser Bildschirmseiten, wenn der Benutzer mit der digitalen Lösung interagiert (Interaktionsablauf). Bei dieser Beschreibung kann es sich um ein Storyboard oder eine andere Art der Spezifikation des logischen Ablaufs von Bildschirmseiten handeln, z. B. was passiert, wenn der Benutzer eine Schaltfläche berührt oder wenn ein bestimmtes Ereignis eintritt.

Beispiel YPRC. Abbildung 44 zeigt drei Papier-Prototyp-Bildschirme der Smartphone-App der YPRC-Fallstudie. Das linke Bild zeigt die Anzeige auf dem Smartphone, wenn die Smartwatch mit dem Smartphone verbunden ist. Die anderen beiden Bildschirme zeigen die Inhalte während des Lauftrainings an, wie z. B. die verstrichene Zeit, die gelaufene Distanz, das durchschnittliche Lauftempo, die aktuelle und durchschnittliche Pulsfrequenz sowie die aktuelle Position des Läufers auf der Karte der geplanten Laufstrecke.

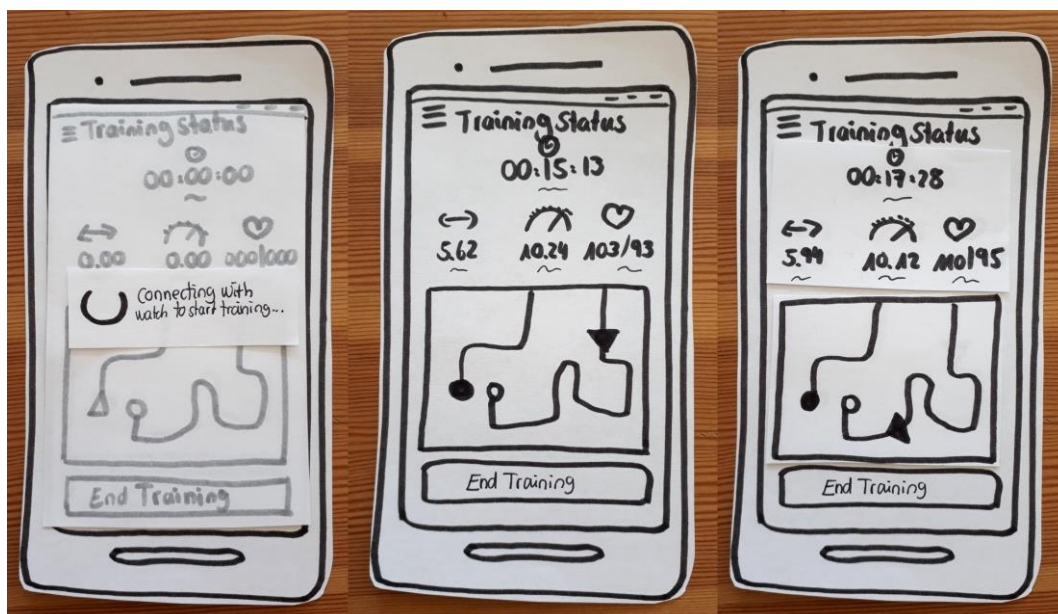


Abbildung 44 - Beispielbildschirme eines Papierprototyps für die YPRC-Fallstudie

Die Bildschirme lassen sich auf kleinen Zetteln oder auf Haftnotizen zeichnen, so dass sie in ein Modell des Zielgeräts (Hardware-Display) passen. Es können mehrere dieser gezeichneten Bildschirmseiten verwendet werden, um durch den Austausch dieser Bildschirmhalte den Ablauf bei einer Interaktion durch einen Benutzer darzustellen. Ein großes Stück Papier oder Karton kann dazu als Modell dienen.

Beispiel YPRC. Abbildung 45 illustriert ein solches Modell mit Haftnotizen aus der YPRC-Fallstudie. Die gezeichneten Bildschirmseiten auf der rechten Seite der Abbildung können bei einer Diskussion mit den Stakeholdern oder einem Usability-Test mit den Benutzern auf den Smartphone-Bildschirm auf der linken Seite der Abbildung geklebt werden (siehe unten).

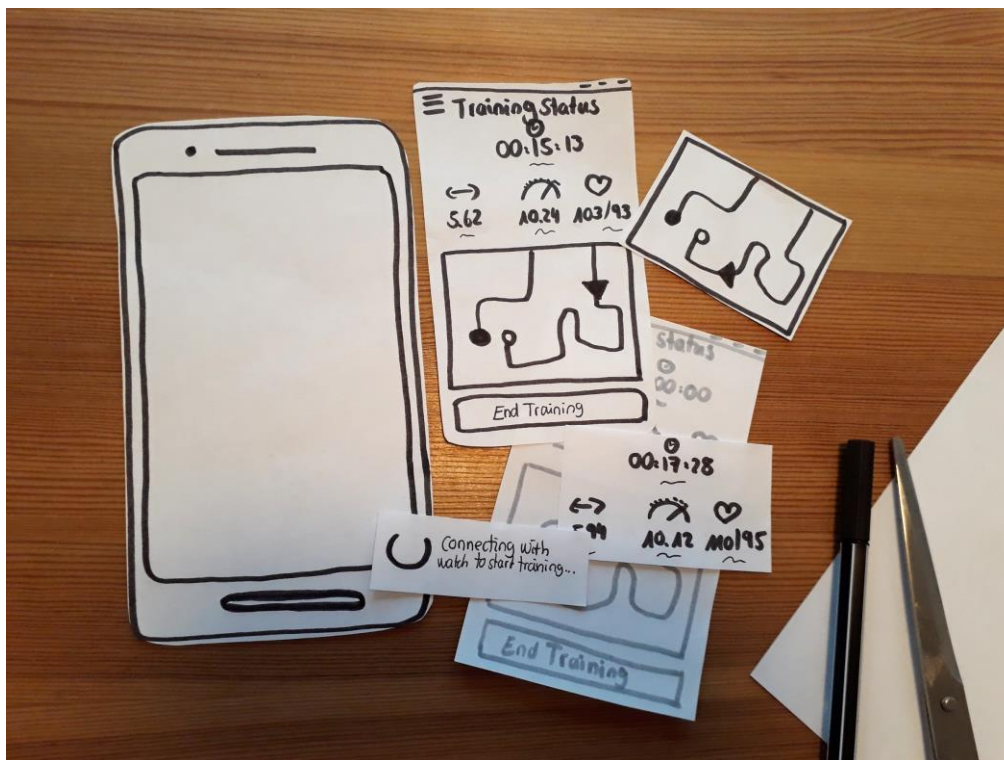


Abbildung 45 - Beispiel eines Papierprototyps für die Fallstudie YPRC mit mehreren skizzierten Bildschirmseiten

Ein Prototyp kann für fast jedes Gerät erstellt werden, z. B. für einen PC, ein Tablet, ein Smartphone oder eine Smartwatch. In Erweiterung dieses Ansatzes sind auch Prototypen aus Pappe oder einem anderen Material möglich, die eine frühe Version eines physischen Geräts darstellen.

Beispiel YPRC. Abbildung 46 zeigt einen Prototyp der Smartwatch aus der YPRC-Fallstudie mit verschiedenen alternativen Display-Inhalten. Die oberen Bilder zeigen die Warnung vor einer zu hohen Pulsfrequenz und die unteren Bilder zeigen gleichzeitig die aktuelle und die durchschnittliche Pulsfrequenz. Wie die Bilder auf der rechten Seite zeigen, kann ein solch einfacher Prototyp dazu verwendet werden, mit einem Läufer zu untersuchen, wie geeignet solche Anzeigen sind und wie gut die physischen Abmessungen des Geräts passen, wenn es beispielsweise am Handgelenk getragen wird.



Abbildung 46 - Beispiel eines Kartonprototyps der Smartwatch der YPRC-Fallstudie

Ein Storyboard¹¹ visualisiert, wie der Benutzer mit der digitalen Lösung interagiert. Es visualisiert den Interaktionsablauf, der beispielsweise festlegt, was passiert, wenn der Benutzer eine

¹¹ Ein Storyboard ist sehr leistungsfähig, wenn es mit einem Informationsszenario kombiniert wird (vgl. [RoCa2003]).

Schaltfläche berührt oder wenn ein bestimmtes Ereignis eintritt, wie z. B. das Überschreiten der oberen Pulsfrequenzgrenze.

Beispiel YPRC. Abbildung 47 zeigt ein Beispiel für ein Storyboard aus der YPRC-Fallstudie. Ausgehend vom oberen linken Bild über das untere linke bis hin zum oberen rechten Bild zeigt das Storyboard, wie ein Läufer sein Smartphone und seine Smartwatch für ein Lauftraining vorbereitet. Es zeigt dann den Läufer beim Laufen und schließlich eine Situation, in der die Herzfrequenz 180 Schläge pro Minute erreicht und damit die persönliche Obergrenze des Läufers überschreitet. Wie bereits angedeutet, zeigt dieses Storyboard nur einen Teil einer größeren Geschichte, die die Interaktion eines Läufers mit der digitalen YPRC-Lösung visualisiert.



Abbildung 47 - Beispiel für ein Storyboard der YPRC-Fallstudie

Beim Testen eines bestimmten Interaktionsablaufs mit einem Papierprototyp enthält der Prototyp eine Reihe von skizzierten Bildschirmseiten mit dem grundlegenden Layout der Anzeigekomponenten und einer knappen Darstellung des grundlegenden Inhalts - beispielsweise der Schaltflächen und Menüpunkte. In der Regel werden verschiedene Ablaufalternativen für die Erledigung einer Aufgabe des Benutzers vorbereitet.

Usability-Test mit Papierprototypen

Als Beispiel wird in den folgenden Abschnitten die Verwendung eines Papierprototyps für einen Usability-Test beschrieben. Bei einem solchen Usability-Test berichten Testbenutzer anhand des Papierprototyps über die Benutzbarkeit (Usability) des Designs, das durch den Prototyp dargestellt wird. Um solche Tests vorzubereiten, werden der Anwendungskontext und die Spezifikation der Benutzeranforderungen, auf denen der Prototyp basiert, benötigt. Zu den Werkzeugen, um dies zu definieren, gehören Personas, (Kontext-)Szenarien oder Use Cases. Anschließend werden konkrete (Benutzer-)Aufgaben für den Einsatz des Prototyps ausgewählt -

zum Beispiel aus vorhandenen Vorgaben wie einem Digital Design Brief oder dem Digital Design Konzept.

Sobald der Usability-Test vorbereitet ist, versucht ein Benutzer, die ausgewählten Aufgaben mit dem Prototyp zu erledigen. Ein Moderator führt den Benutzer und erklärt das System. Eine zweite Person simuliert das System, indem sie den Papierprototyp manipuliert. Wenn der Benutzer beispielsweise eine Schaltfläche auf dem Papier berührt, erscheint eine neue Bildschirmseite oder es öffnet sich ein Menü, indem z. B. ein Klebezettel entfernt wird, der einen Teil der Bildschirmseite verdeckt. Der Benutzer kann durch ein Menü blättern, was zu einer Verschiebung des Papierstreifens mit dem skizzierten Menü führt. In der Regel sind Benutzer sehr gut in der Lage, sich solche Interaktionen mit der künftigen digitalen Lösung vorzustellen, selbst wenn nur eine Interaktion mit einem Papierprototyp stattfindet. Auf diese Weise kann wertvolles Feedback der Benutzer gesammelt werden. Ein oder mehrere Beobachter zeichnen die Kommentare der Benutzer auf und beobachten das Verhalten der Benutzer. Sowohl Kommentare als auch Beobachtungen der Benutzer können erfolgreiche Designs oder aber (Usability-)Probleme des Designs aufzeigen. Je nach Bedarf oder bei begrenzten Ressourcen, können ein oder zwei Personen den Usability-Test durchführen. Entweder nimmt eine Person alle Moderations-, Simulations- und Beobachtungsaufgaben allein wahr, oder zwei Personen teilen sich diese Aufgaben untereinander auf.

Nach Abschluss des Usability-Tests führt die Analyse der gesammelten Daten in der Regel zu Verbesserungsideen für das Design. Diese Änderungsvorschläge werden ausgewählt, um das Design oder die Realisierung der digitalen Lösung zu verbessern. Je nach Erfüllung der Anforderungen kann ein neuer (Papier-)Prototyp erstellt und für eine weitere Studie mit Benutzern verwendet werden. Der in diesem Abschnitt beschriebene Prozess ist eine Umsetzung des User-Centered Designprozesses, wie er in [ISO2019] beschrieben ist.

Praxistipp

Prototyping auf Papier ist eine einfache, flexible, aber wirkungsvolle Methode, bei der Papierprototypen in iterativen Schritten (Iterationsschleifen) verwendet werden, um schnell wertvolles Feedback von Benutzern und anderen Stakeholdern zu erhalten. Interaktionsdesigner verwenden häufig Papierprototypen, um die Gestaltung von User Interfaces zu verbessern. In vielen Fällen bevorzugen sie diese Art von Prototypen aufgrund ihrer Flexibilität sogar gegenüber Klick-Prototypen. Papierprototypen eignen sich gut für Low-Fidelity-Prototypen. Wenn die Interaktionsabläufe jedoch komplex werden oder High-Fidelity-Prototypen erforderlich sind, sollten andere Prototyping-Werkzeuge verwendet werden (siehe Abschnitt 2.3.4 und 2.3.4.7). Unter [Snyd2003] findet sich eine ausführliche Beschreibung der leistungsstarken Methode des Papier-Prototypings.

2.3.7 Fazit zum Prototyping

Wie dieser Abschnitt über die Einsatzmöglichkeiten des Prototypings zeigt, ist der Bau von Prototypen zur Erstellung neuer Designs und die anschließende Verwendung dieser Prototypen zur Evaluierung ein leistungsfähiges Instrument für den DDP. Prototypen sind eine vorläufige, partielle Instanz einer Designlösung. Sie stellen somit eine Manifestierung einer Idee, eines Konzepts, eines Systems oder einer Lösung dar.

Der Einsatz von Prototypen in einem frühen Stadium des Bauprozesses ermöglicht es, in grundlegend verschiedene Richtungen zu forschen und die Vor- und Nachteile verschiedener

Alternativen frühzeitig zu erkennen. Wenn die Prototypen einfach sind und einen geringen Detailgrad haben, ist es einfach, die Alternativen zu verwerfen, die für die angestrebte digitale Lösung nicht geeignet sind. Dies führt in der Regel nicht zu hohen Kosten, sondern vor allem zu einem zusätzlichen Verständnis und einem erheblichen Fortschritt im Bauprozess. Der DDP muss jedoch bereit sein, Prototypen nur zu diesem Zweck zu erstellen, um sie anschließend zu verwerfen. Später im Bauprozess helfen Prototypen dabei, genauer herauszufinden, ob ein bestimmter kritischer Teil der digitalen Lösung funktioniert oder nicht, oder welche Probleme tatsächlich noch bestehen.

Der Einsatz von Prototypen hat ein breites Anwendungsspektrum in allen Stufen des Bauprozesses (Scoping, Konzeption, Entwicklung und Betrieb) und auf den verschiedenen Abstraktionsebenen (Lösungsebene, Systemebene, Elementebene). Die Anwendung des geeigneten Prototyps, insbesondere in Bezug auf seinen Detailgrad, hängt vom Risiko bestimmter Teile der digitalen Lösung, dem verfügbaren (planbaren) Budget und der Erfahrung des beteiligten Personals, insbesondere des DDP, ab. Der DDP sollte jedoch die mächtige Methode des Prototyping nutzen, sich für ein Budget für diese Aktivität einsetzen und Erfahrungen mit dem breiten Einsatzbereich und den Vorteilen dieser Technik sammeln. Schließlich sollte der DDP die Stärke des Prototyping auf Papier nicht unterschätzen, da dies zur schnellen Beseitigung von Unsicherheiten und zur Verringerung von Risiken bei der Entwicklung innovativer digitaler Lösungen eingesetzt werden kann.

3 Digitales Material

Die technologischen Möglichkeiten (von digitalem Material) im Bereich der Hard- und Software zur Realisierung einer digitalen Lösung sind in den letzten Jahren enorm gestiegen und werden auch in Zukunft weiter steigen (siehe [Kell2016]). Der DDP auf Foundation Level Niveau muss diese Vielfalt an Möglichkeiten kennen und bereit sein, mit der ständigen technischen Weiterentwicklung Schritt zu halten. In diesem Kapitel ist eine Einführung in das breite Spektrum des zur Verfügung stehenden digitalen Materials dargestellt.

Das Kapitel beginnt mit einer Einführung in das Technologieverständnis von Digital Design (Abschnitt 3.1). Die wahrnehmbaren und die zugrundeliegenden Ebenen einer digitalen Lösung beeinflussen sich gegenseitig stark (siehe Abschnitt 1.2). Daher müssen ihre gegenseitigen Abhängigkeiten berücksichtigt werden, um eine gute digitale Lösung anbieten zu können. In den Abschnitten 3.2 und 3.3 wird ein Überblick über die wahrnehmbaren und zugrundeliegenden Technologien auf der Grundlage der beiden in Abschnitt 1.2 vorgestellten Ebenen des digitalen Materials vermittelt. In Abschnitt 3.4 werden drei technologieorientierte Wissensbereiche vorgestellt, die für einen DDP auf Foundation Level Niveau für wichtig sind. Abschließend wird in Abschnitt 3.5 die Technologie aus der Perspektive des Digital Design betrachtet.

3.1 Technologie verstehen

Der DDP versteht Technologie als gestaltbares Material für den Bau einer digitalen Lösung (siehe Abschnitt 1.2.1). Technologie als Material zu verstehen, bedeutet im Wesentlichen, dass nur ein profundes Wissen über Technologie es dem DDP ermöglicht, hervorragende digitale Lösungen zu schaffen. Der DDP kennt die Möglichkeiten und Grenzen z. B. von Verarbeitungs- und Datenübertragungstechnologien sowie von Speichermedien. Durch die Kombination dieser verschiedenen Aspekte schafft der DDP eine digitale Lösung, die dem Benutzer oder dem Kunden einen Mehrwert bietet.

Das Modell von Form, Funktion und Qualität als Grundlage für das Verständnis von Technologie

Um den Zugang zu diesem Wissen zu strukturieren und das Wissen über Technologie weiterzugeben, verwendet der DDP das Modell von Form, Funktion und Qualität des digitalen Materials. Wie in Abschnitt 1.2.1 beschrieben, wird durch dieses Modell zwischen der wahrnehmbaren und der zugrundeliegenden Ebene unterschieden. Die wahrnehmbare Ebene umfasst die Form, Funktion und Qualität, die von den Stakeholdern wahrgenommen werden kann. Die zugrundeliegende Ebene ist der Wahrnehmung durch Stakeholder verborgen, aber nur durch sie wird die wahrnehmbare Ebene ermöglicht. Abbildung 48 veranschaulicht diese Unterscheidung zwischen den beiden Ebenen in Bezug auf Form und Funktion und liefert Beispiele für diese Sichtweise auf Technologie.

Wahrnehmbare Technologie

Die linke Seite der Abbildung zeigt Beispiele für Hardwaretechnologien. Die Form eines Endgeräts, z. B. eines Smartphones oder eines Tablet-Computers, kann vom Benutzer direkt wahrgenommen werden. Wurde bei der Herstellung des Geräts besonderes Material verwendet, z. B. Magnesium oder Glas, kann der Benutzer dies direkt fühlen. Der Benutzer kann alle Tasten des Geräts sehen und fühlen. Dies sind Beispiele für die wahrnehmbare Form. Die Art und Weise, wie die Benutzer die Tasten eines Geräts verwenden, um beispielsweise die Lautstärke zu regeln oder den Ton stumm zu schalten, ist Teil des sogenannten Interaktionsflusses und kann ebenfalls

direkt wahrgenommen werden. Wenn es eine spezielle Stummschalttaste gibt, kann der Benutzer die Stummschaltung aktivieren, den Druckpunkt und die Bewegung spüren, das Abschalten des Tons hören und eventuell durch eine farbige Anzeige auf der Taste, sogar auf der Taste sehen, dass das Gerät stummgeschaltet ist. Dieses Beispiel für Lautstärkeregelung und Stummschaltung ist Teil der wahrnehmbaren *Remote-Voice-Coaching-Funktion* der YPRC-Fallstudie.

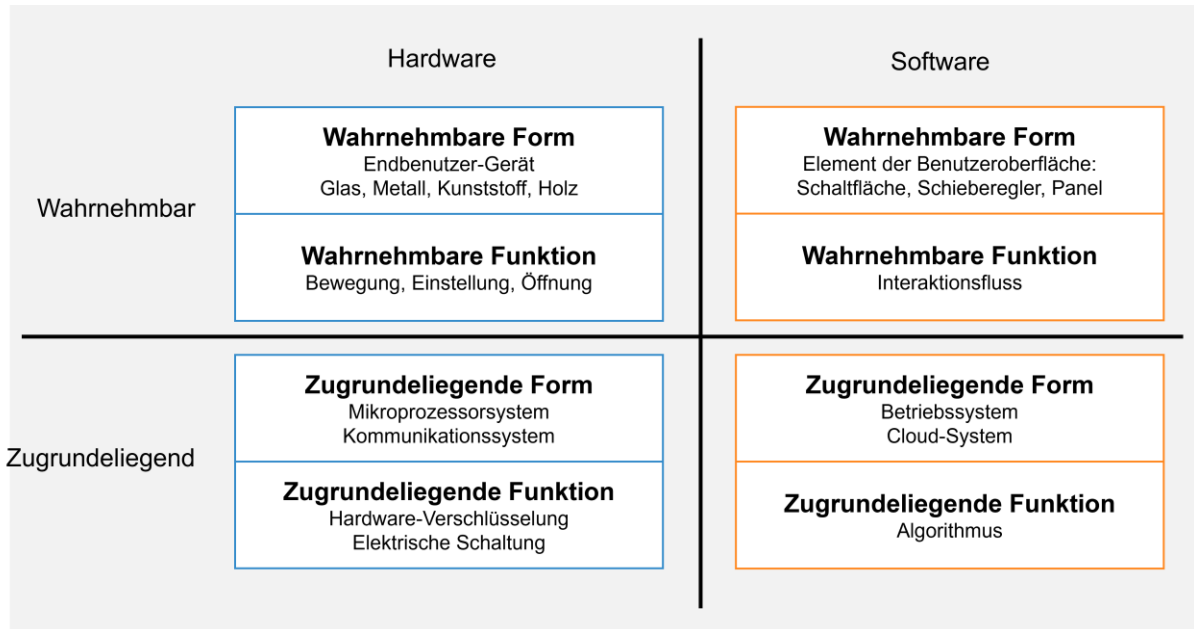


Abbildung 48 - Beispiele für Technologie, unterteilt in (1) wahrnehmbare Form und Funktion für Hardware und Software (oberer Teil) und (2) zugrundeliegende Form und Funktion für Hardware und Software (unterer Teil)

Bei einer digitalen Lösung ist die Hardware-Funktion nur ein Teil der Lösung. Um das Beispiel fortzusetzen: Um nach dem Drücken der (Hardware-)Lautstärketaste die Einstellung der Lautstärke anzuzeigen, ist eine Software erforderlich. Ein Element des User Interface, das auf dem Bildschirm des Smartphones erscheint und den Status oder die Änderung der Lautstärke anzeigt, ist ein Beispiel für die wahrnehmbare Form des Softwareteils (siehe den oberen rechten Teil von Abbildung 48).

Der wahrnehmbare Softwareteil des Interaktionsflusses bestimmt beispielsweise, wie viele Tastendrucke nötig sind, um die Lautstärke von der niedrigsten auf die höchstmögliche Einstellung zu ändern. Ein komplexeres Beispiel für eine wahrnehmbare Funktion der Software ist die Interaktion, die zum Entsperren eines Smartphones durch Eingabe des PIN-Codes erforderlich ist. Unter 3.2 befinden sich weitere Einzelheiten zur wahrnehmbaren Technologie.

Zugrundeliegende Technologie

Neben diesen wahrnehmbaren Elementen, die im oberen Teil von Abbildung 48 dargestellt sind, verwendet eine digitale Lösung in der Regel weitere Technologien, die nicht direkt wahrnehmbar sind, d. h. die zugrundeliegende Technologie (unterer Teil von Abbildung 48). Im Beispiel der Lautstärkeregelung ist eine zugrundeliegende Software der Teil des Betriebssystems, der von der Anwendung zur Lautstärkeregelung verwendet werden muss, um die Lautstärke tatsächlich zu ändern. Ein Betriebssystem bietet in der Regel eine Anwendungsprogrammierschnittstelle (API) in Form einer Menge von Funktionen oder Methoden, auf die mithilfe einer Programmiersprache zugegriffen werden kann. Die Art und Weise wie diese Funktionen oder

Methoden verwendet werden müssen und wie sie funktionieren, ist eine zugrundeliegende Funktion der Software.

Um die Lautstärke vollständig abzuschalten, muss die Anwendung möglicherweise zunächst das API verwenden, um die Lautstärke um eine Stufe zu verringern, um dann durch eine andere Funktion zu prüfen, ob die Lautstärke bereits auf null gesetzt wurde. Dies wird so lange wiederholt, bis die Lautstärke tatsächlich null ist. In einer alternativen Implementierung könnte das API eine spezielle Funktion bereitstellen, die die Lautstärke in einem Schritt stummschaltet oder den Ton langsam ausblendet, wodurch die beschriebene iterative Verwendung der API-Funktionen überflüssig wird. Möglicherweise gibt es eine API-Funktion, mit der ein Audiosignal wieder eingeblendet werden kann. Wie bereits erwähnt, erübrigen sich damit aus Sicht der User Interface-Anwendung die schrittweise Anpassung und Kontrolle der Lautstärke.

Die zugrundeliegende Form der Hardware eines Smartphones besteht zum Beispiel aus allen elektronischen Komponenten, die nicht direkt sichtbar oder mit anderen menschlichen Sinnen erkennbar sind (nicht wahrnehmbar). Als Beispiel für die zugrundeliegende Hardware verfügt ein typisches Smartphone über einen (kleinen) Lautsprecher und einen Verstärker, um akustische Signale wiederzugeben und um (Freisprech-)Telefonate zu führen. Zum Telefonieren wird ein Kommunikationsprozessor (Signalprozessor) benötigt. Ein Anwendungsprozessor ist für die Ausführung von (heruntergeladenen) Anwendungen zuständig. Die zugrundeliegenden Hardwareelemente des Smartphones können miteinander kommunizieren und ermöglichen dadurch komplexe Use Cases, wie z. B. das Streaming von Musik mit Hilfe des Anwendungsprozessors, die über die mobile Datenübertragungsfunktion des Funkmoduls heruntergeladen wird.

Die zugrundeliegende Funktion der Hardware beschreibt, wie die Hardware-Komponenten (zugrundeliegende Form) zusammenarbeiten. Im obigen Beispiel muss der Verstärker den Lautsprecher auf Basis der Signale des Kommunikationsprozessors ansteuern, um die Funktion des (Freisprech-)Telefonierens zu ermöglichen. Sollen auch andere Anwendungen akustische Signale abspielen können, muss der Verstärker ebenfalls an den Anwendungsprozessor angeschlossen werden.

Bei der Betrachtung dieser einfachen Beschreibung der Hardware-Architektur stellt sich folgende Frage: Was passiert, wenn die Musikanwendung einen Musiktitel abspielt und gleichzeitig ein Telefonanruf eingeht? Es liegt auf der Hand, dass der Anruf auch dann dem Benutzer angezeigt werden muss, wenn Musik gespielt wird. Dies kann durch einen sogenannten Mischer (engl. mixer, eine zusätzliche zugrundeliegende Hardware) realisiert werden, der das Mischen von Signalen aus dem Kommunikationsprozessor und dem Anwendungsprozessor ermöglicht. Abschnitt 3.3 enthält weitere Einzelheiten zur zugrundeliegenden Technologie.

Die in den obigen Beispielen beschriebenen Formen und Funktionen sind heute Teil der Standard-Smartphone-Architektur und in allen modernen Geräten verfügbar. Das Beispiel veranschaulicht, wie Überlegungen zur zugrundeliegenden Form und Funktion die Möglichkeiten auf der wahrnehmbaren Ebene beeinflussen. Wäre der Mischer (engl. mixer) nicht verfügbar, wäre das reibungslose Überblenden der akustischen Signale - gesteuert durch das Telefonat oder die Musikanwendung - nicht möglich.

Der Einfluss der zugrundeliegenden Ebene auf die wahrnehmbare Ebene ist generell ein wichtiger zu betrachtender Faktor, um digitale Lösungen zu gestalten, die Bedürfnisse der Benutzer und neue Technologien optimal miteinander verbinden. Insbesondere durch die

Verfügbarkeit moderner Anwendungen der künstlichen Intelligenz (KI) ist das Verständnis der zugrundeliegenden Form und Funktion und ihres Einflusses auf die wahrnehmbare Form und Funktion von entscheidender Bedeutung.

Vorteile des Technologieverständnisses

Ein tiefes Verständnis über Technologie und technologische Entwicklungen bietet wichtige Vorteile für den DDP:

- *Vermeidung von nicht realisierbaren Lösungen:* Ein fundiertes Verständnis über verfügbare Technologien verhindert die Definition von nicht realisierbarer Form, Funktion und Qualität einer digitalen Lösung. So ist beispielsweise die Realisierung einer 4k-Videochat-Funktion derzeit nicht praktikabel, da dies eine Bandbreite von mindestens 25 Mbit/s erfordern würde.
- *Inspiration für neue Lösungen:* Innovative Technologien bieten Möglichkeiten, die neue Aspekte einer digitalen Lösung oder eine völlig neue digitale Lösung ermöglichen können. Beispiele hierfür sind neue Anwendungen, die auf dem Einsatz von künstlicher Intelligenz (KI) basieren.
- *Kommunikation auf Augenhöhe mit Fachleuten für Software:* Oft werden Spezialisten benötigt, um Teile einer digitalen Lösung zu entwerfen und zu entwickeln. Die Kenntnis der jeweiligen Technologiebereiche ermöglicht die Kommunikation mit diesen Fachleuten. So hilft beispielsweise das Wissen über die Existenz, die allgemeine Verwendung und die Funktionalität von Bibliotheken zur Programmierung von User Interfaces bei dieser Kommunikation.
- *Kommunikation auf Augenhöhe mit Fachleuten für physische Produkte:* Wenn die digitale Lösung ein hauseigenes physisches Produkt enthält, ermöglicht die Kenntnis der zugrundeliegenden Technologie dem DDP, wichtige Aspekte mit den entsprechenden Fachleuten zu diskutieren und abzustimmen. Dies ist besonders wichtig, wenn die digitale Lösung spezielle Software- und Hardwareteile umfasst. Wenn z. B. die digitale Lösung, die auf einem mobilen Gerät läuft, eine sehr hohe Leistung der Prozessoren erfordert, wird eine Diskussion und Abstimmung über die maximale Stromstärke der verfügbaren Batterie notwendig.
- *Kommunikation auf Augenhöhe mit Lieferanten oder Partnern:* Ein tiefes Verständnis der eingesetzten Technologien ermöglicht es dem DDP, besser mit Lieferanten oder Partnern zusammenzuarbeiten, die Teile einer digitalen Lösung realisieren. Dies gilt sowohl für Anbieter von Hardware- als auch von Softwareteilen. So ist beispielsweise das Wissen über die Verfügbarkeit und die allgemeine Funktion von Videokompressionscodecs oder White-Label-Hardware wichtig, um solche Produktteile als Unterauftrag vergeben zu können.

Da digitale Technologien einer ständigen Entwicklung unterliegen, muss der DDP die aktuellen technologischen Entwicklungen kontinuierlich verfolgen und neue Technologien verstehen, um auf dem neuesten Stand zu bleiben.

Unterschied zwischen dem Verständnis von Technologie und der Entwicklung mit Technologie

Es ist wichtig, zwischen technologischem Wissen und der Fähigkeit, mit dieser Technologie zu gestalten und zu entwickeln zu unterscheiden. Nur wenn die Möglichkeiten und Grenzen der verfügbaren Technologie bekannt sind, kann die bestmögliche digitale Lösung entwickelt werden. Der DDP muss nicht unbedingt ein Experte in allen technologischen Bereichen sein. Zwar muss

der DDP beispielsweise alle Möglichkeiten und Grenzen von Benutzerinteraktionen unter Verwendung aktueller Technologien verstehen, aber Fachleute für Interaktionsdesign beschäftigen sich umfassend mit dieser Art von Design, um ein interaktives System zu gestalten (siehe auch 2.1.4).

Die Auswahl bestimmter Technologien wirkt sich auf die Qualität eines digitalen Systems und damit auf die gesamte digitale Lösung aus. Die Auswahl von Technologie kann sich in unterschiedlichen Qualitätsmerkmalen manifestieren, die sich auf die wahrnehmbaren und zugrundeliegenden Qualitäten auswirken (siehe Abschnitt 2.1.3).

Die Auswahl geeigneter Technologien muss daher im Einklang mit den definierten Qualitätskriterien (d. h. den Qualitätsanforderungen) für ein digitales System und die digitale Gesamtlösung stehen. Wenn Technologien ausgewählt werden, ohne vordefinierte Qualitätskriterien zu kennen oder zu berücksichtigen, kann dies zu digitalen Systemen und Lösungen führen, die nicht den erwarteten, wahrnehmbaren Qualitätskriterien der Benutzer entsprechen und deshalb möglicherweise nicht akzeptiert werden. Darüber hinaus kann sich eine ungeeignete Technologieauswahl auch auf die zugrundeliegenden Qualitätskriterien auswirken und die Wartung des Systems durch die Entwickler verhindern oder zumindest erschweren.

All diese Überlegungen gelten für Hardware und Software sowie für die Form und Funktion der wahrnehmbaren und der zugrundeliegenden Technologien. Wie in Abschnitt 2.1.3.2 diskutiert, muss ein DDP ein aktives Qualitätsmanagement betreiben. Hierzu ist es erforderlich, einen guten Überblick über die vorhandenen Technologien zu haben und deren Auswirkungen auf die wichtigsten Qualitätsmerkmale zu verstehen.

Beispiel YPRC. Die YPRC-Fallstudie liefert gute Beispiele für Diskussionen. Es sei angenommen, dass die zentrale Recheneinheit (CPU) der Smartwatch des Läufers ohne nähere Betrachtung der Qualitätsanforderungen ausgewählt wurde. Weiterhin fand diese Entscheidung ohne Verständnis der möglichen Auswirkungen statt. Als zugrundeliegende Hardwarekomponente, die die Form des digitalen Systems definiert, kann die CPU die wahrnehmbaren und zugrundeliegenden Eigenschaften des digitalen Systems und die Qualität der digitalen Lösung insgesamt stark beeinflussen.

So kann beispielsweise die Wahl einer preisgünstigen und leistungsschwachen CPU zu Leistungsproblemen führen und damit die wahrnehmbare Qualität des digitalen Systems beeinträchtigen (siehe Abschnitt 2.1.3.4). Darüber hinaus könnte sich dies auch auf die digitale Lösung auswirken, da Leistungsprobleme beispielsweise das Ziel der Benutzer, die Nutzung ihres Lauftrainers zu genießen, beeinträchtigen könnten (hedonische Qualität, siehe Abschnitt 2.1.3.5). Die Wahl einer ungeeigneten CPU kann sich auch auf andere technologische Entscheidungen auswirken, etwa auf die Wahl des Betriebssystems (zugrundeliegende Form, Software). Dies wiederum kann sich auf die Entwicklungsumgebung auswirken, die für die Entwicklung der Softwarekomponenten des digitalen Systems benötigt wird, und kann auch die Art des User Interface (wahrnehmbare Form, Software) und die Art und Weise, wie die Benutzer mit dem System interagieren können (wahrnehmbare Funktion, Software), beeinflussen.

Genau wie die bereits erwähnten Leistungsprobleme könnten sich diese Effekte negativ auf die wahrgenommene Qualität des digitalen Systems auswirken - und zwar in Bezug auf die Benutzbarkeit - und sich darüber hinaus negativ auf die digitale Lösung insgesamt auswirken, was die Freude an der Nutzung des Lauftrainers betrifft. Die Wahl einer ungeeigneten CPU wirkt sich damit nicht nur negativ auf die wahrnehmbaren Qualitäten aus, sondern zwingt die Entwickler möglicherweise auch dazu, eine ungeeignete Entwicklungsumgebung zu verwenden, was wiederum negative Auswirkungen auf die zugrundeliegenden Qualitäten wie die Wartbarkeit haben kann.

Im Idealfall ist sich der DDP dieser potenziellen negativen Auswirkungen bewusst und wird die Auswahl einer geeigneten CPU für die Smartwatch unterstützen.

3.2 Wahrnehmbare Technologie

Wahrnehmbare Technologien werden verwendet, um die Teile einer digitalen Lösung zu realisieren, die ein Benutzer direkt wahrnehmen kann. Ein DDP auf Foundation Level Niveau sollte sich mit Endgeräten, Interaktionstechnologien und Technologien für die Softwareimplementierung von User Interfaces auskennen.

3.2.1 Endgeräte

Zur Realisierung digitaler Lösungen werden häufig standardisierte Endgeräte wie Notebooks, Tablet-Computer oder Smartphones eingesetzt. Diese Geräte können der wahrnehmbaren Technologie zugeordnet werden. Andererseits bieten sie technische Fähigkeiten, die der zugrundeliegenden Technologie zugeordnet werden können (z. B. drahtlose Kommunikationstechnologien wie WLAN oder Bluetooth). Digital Design hat jedoch keinen direkten Einfluss auf die innere Struktur solcher standardisierten Endgeräte, daher ist die wahrnehmbare Form in diesem Kontext die geeignete zu betrachtende Kategorie. Wenn

standardisierte Endgeräte als Teil einer digitalen Lösung verwendet werden, müssen die angenommenen technischen Fähigkeiten - z. B. Bildschirmgröße und -auflösung, Kommunikationstechnologien, Leistung der Prozessoren, Speichergröße - klar definiert werden, um die erforderlichen Ressourcen für die digitale Lösung bereitzustellen.

Durch die folgende Unterscheidung entstehen grobe Klassen von Endgeräten:

- *Stationäre Geräte an festen Standorten*, wie beispielsweise Personal Computer, intelligente Lautsprecher (z. B. Amazon Echo oder Google Home) oder intelligente Waagen. Aufgrund ihres festen Standorts können sie eine Steckdose als Stromquelle nutzen, und der belegte Raum ist - im Prinzip - nicht begrenzt. Daher kann diese Kategorie die höchste Leistung unter diesen hier definierten Klassen in Bezug auf die Rechenleistung haben. Dies bedeutet, dass sehr schnelle Hauptprozessoren (CPUs) und Grafikprozessoren (GPUs), ein großer Arbeitsspeicher, hohe Massenspeicherkapazitäten und große Displays möglich sind.
- *Tragbare Geräte* können an verschiedenen Orten eingesetzt werden. Diese Klasse reicht von Mehrzweckgeräten wie Notebooks, Tablet-Computern oder Smartphones bis hin zu - in wachsender Zahl - Einzweckgeräten wie Kartenlesern, Fingerabdruck- oder Retina-Scannern, Geräten zur Bestellung von Konsumgütern (z. B. Amazon Dash Buttons oder Amazon Dash Smart Shelf) oder Customer-Feedback-Terminals. Aufgrund ihrer Tragbarkeit müssen diese Geräte in der Regel mit einer (wiederaufladbaren) Batterie betrieben werden und können nur eine begrenzte Größe haben, um in den vorgesehenen Raum zu passen, z. B. in eine Tasche, einen Beutel, oder um an einem bestimmten Platz aufgestellt zu werden, z. B. in der Nähe der Kasse oder der Waschmaschine (um schnell Waschmittel zu bestellen). Eine Steckdose oder eine kabelgebundene Netzwerkverbindung kann nur in bestimmten Nutzungsszenarien verwendet werden. Daher sind die verfügbare Rechenleistung und der verfügbare Speicherplatz mittelgroß im Vergleich zu den beiden anderen Klassen in dieser Liste. Die mögliche Anzeigegröße reicht von mittelgroß, für die oben erwähnten Mehrzweckgeräte, bis klein oder gar nicht vorhanden für die Einzweckgeräte.
- *Wearable Devices* werden am Körper getragen oder sogar in den Körper implantiert. Beispiele hierfür sind Aktivitätstracker, Blutzuckermessgeräte, Smartwatches oder andere kleine Geräte, die am Körper tragbar sind. Diese Geräte dürfen nur einen sehr begrenzten Raum einnehmen und müssen ausschließlich mit (wiederaufladbaren) Batterien betrieben werden. Im Vergleich zu den anderen beiden Klassen in dieser Liste haben diese Geräte die geringste Rechenleistung und das niedrigste verfügbare Speicherprofil. Außerdem ist die Anzeigegröße sehr klein. Einige Geräte haben gar kein Display und müssen andere Schnittstellentechnologien - wie z. B. die Sprachtechnologie oder drahtlose Technologien verbunden mit einem anderen Gerät - für den direkten oder indirekten Informationsaustausch mit dem Benutzer verwenden.

3.2.2 Interaktionstechnologie

Moderne Interaktionstechnologie besteht aus einer Kombination von komplexen Hard- und Softwaresystemen, die zur wahrnehmbaren Form und Funktion gehören (siehe Abschnitt 1.2). Ein funktionierender Smartphone-Touchscreen, der eine Berührungsinteraktion ermöglicht, erfordert beispielsweise eine komplizierte Kombination von Hardware- und Softwarekomponenten, um eine angemessene Reaktion des Geräts auf die Berührung des Bildschirms durch den Benutzer zu erzeugen.

Aus Sicht des DDP sind die dynamischen Aspekte dieser User Experience am wichtigsten, d.h. der Interaktionsfluss, der durch die Kombination von Hardware und Software der Interaktionstechnologie ermöglicht wird. Daher konzentriert sich der DDP hauptsächlich auf die wahrnehmbaren Funktionen der jeweiligen Interaktionstechnologie. Siehe 1.2.1 und 3.1 für eine Erklärung der wahrnehmbaren und zugrundeliegenden Ebenen.

Eine Art von Benutzerschnittstelle, die auch heute noch weit verbreitet ist, ist das grafische User Interface (GUI). Bei der Verwendung eines grafischen User Interface (GUI) navigiert der Benutzer - in der Regel innerhalb eines Fensters - durch Menüs auf einem Bildschirm und wählt (Menü-) Elemente oder Symbole mit einem Mauszeiger (oder einem anderen Navigationsgerät) oder über die Tastatur aus. Diese Art der Interaktion über Maus und Tastatur ist indirekt, da das auszuwählende Element auf dem Bildschirm nicht direkt angeklickt werden kann, z. B. mit dem Finger. Ein Vorteil eines grafischen User Interface ist, dass es alle verfügbaren Befehle der Schnittstelle durch Menüs, Symbole und andere grafische Elemente anzeigt. Der Benutzer findet Befehle, indem er diese Elemente erkundet, und muss sich keinen bestimmten Befehl merken. Dies unterstützt die psychologische Funktion des Wiedererkennens, d. h. der Benutzer muss nur die Funktion eines Befehls oder einer Option erkennen, anstatt sie sich zu merken. Spalte drei von Tabelle 16 fasst diese Charakterisierung eines GUIs zusammen.

Tabelle 16 - User Interface Paradigmen ¹²

	Kommandozeilen- Schnittstelle (Command Line Interface, CLI)	Grafisches User Interface (Graphical User Interface, GUI)	Natürliches User Interface (Natural User Interface, NUI)
Interaktion	abstrakt	indirekt	direkt
Befehle	gerichtet	explorativ	kontextabhängig
Psychologische Funktion	Erinnern	Erkennen	Intuition

Bevor es grafische User Interfaces gab, waren Kommandozeilen-Schnittstellen (CLI) eine sehr verbreitete Art der Interaktion mit Computern. Bei einer Kommandozeilen-Schnittstelle zeigt der Computer eine Eingabeaufforderung auf dem Bildschirm an und wartet darauf, dass der Benutzer ein Kommando oder einen Befehl eingibt, das den Computer zur Ausführung einer Funktion veranlasst. Der Benutzer muss dieses Kommando manuell in die Kommandozeile eingeben. Der Benutzer muss sich in der Regel die möglichen Kommandos und ihre Optionen merken. Normalerweise gibt es nur eine begrenzte Möglichkeit, diese Kommandos über zusätzliche Hilfe-Funktionen oder Optionen zu erkunden, falls diese überhaupt verfügbar sind. Daher wird diese Art der Interaktion als abstrakt bezeichnet (siehe Spalte zwei von Tabelle 16). Ein CLI kann weiterhin in vielen modernen Betriebssystemen wie Android, iOS, Linux oder Windows aktiviert werden, um spezielle Aufgaben zu lösen. Heute spielen diese Schnittstellen eine wichtige Rolle bei der Batch-Verarbeitung, bei der IT-Administration oder allgemein bei der Nutzung durch Programmierexperten.

¹² Angepasst von [Hinm2012], das auf einer Präsentation von Dennis Wixon basiert.

Für moderne Smartphones oder Tablets, die hochwertige Touchscreens verwenden, wurden in den letzten Jahren die GUIs erweitert, um eine direktere Manipulation von Objekten zu ermöglichen. Das User Interface solcher Geräte ist ein Beispiel für ein natürliches User Interface (NUI). Diese direkte Darstellung funktioniert durch die Betrachtung von Objekten auf dem Bildschirm und die Auswahl von Befehlen und Optionen mit dem Finger oder speziellen Stiften, die als Zeigeinstrument fungieren. Dies ermöglicht eine direktere Benutzerinteraktion als bei reinen GUIs. Befehle und Optionen sind aus dem Kontext heraus verfügbar und können intuitiver verwendet werden als beispielsweise in GUIs. Diese Beschreibung einer Schnittstelle ist nur ein Beispiel für ein natürliches User Interface (NUI). Eine noch natürlichere Interaktionsmethode ist beispielsweise die Spracheingabe und -ausgabe, die auch von modernen Smartphones unterstützt wird. Das Ziel von NUIs ist es, die Interaktion mit Computern auf die gleiche Weise zu ermöglichen wie die Interaktion mit der physischen Welt (siehe [ShRP2019]) oder mit anderen Menschen. Diese Aspekte der NUIs sind in der vierten Spalte der Tabelle 16 zusammengefasst.

Die in Tabelle 16 zusammengefassten Typen von User Interfaces und ihre Eigenschaften spielten in der Vergangenheit eine wichtige Rolle und sind auch heute noch wichtig für die Interaktion mit Computern. Hinman ([Hinm2012]) bezeichnet sie als User Interface Paradigmen.

Zusätzlich zu diesen User Interface Paradigmen kann zwischen verschiedenen Benutzer-Schnittstellentypen unterschieden werden. Viele dieser Typen stehen im Zusammenhang zu den erwähnten User Interface Paradigmen. Im Folgenden sind die Benutzer-Schnittstellentypen aufgeführt, die derzeit von Bedeutung sind (siehe [ShRP2019]):

Audio-Interface

Der Benutzer erhält über einen Lautsprecher oder Kopfhörer Audiosignale, manchmal zusätzlich zu den visuellen Rückmeldungen auf dem Bildschirm. Solche Signale spielen bei vielen alltäglichen Schnittstellen eine wichtige Rolle. In Haushaltsgeräten wie dem Kühlschrank, der Spülmaschine oder der Mikrowelle signalisieren sie einen bestimmten Zustand des Geräts. Ein einfacher Einparkassistent in einem modernen Auto ist ein Beispiel für ein komplexeres Audio-Interface, das Informationen über die Entfernung zu anderen Objekten vor oder hinter dem Auto z. B. durch die Frequenz, mit der ein Ton abgespielt wird, transportiert. Es ist wichtig, die richtige Komplexität des Audio-Interface zu finden, um eine Informationsüberlastung des Benutzers zu vermeiden.

Sprach-Interface

Ein komplexeres Audio-Interface ist ein Sprach-Interface oder Voice User Interface (VUI), das Sprachausgabe und Spracheingabe ermöglicht. Die Sprachausgabe erfolgt in der Regel über ein Text-to-Speech-System (TTS), mit dem geschriebener (digitaler) Text in eine Sprachausgabe umgewandelt wird. Eine Anwendung eines solchen Systems ist ein Bildschirmlesegerät, das beispielsweise für sehbehinderte Menschen nützlich ist. Moderne Systeme basieren auf vollständig aufgezeichneten Sätzen, die von professionellen Sprechern eingesprochen wurden. Wenn die Datenbank groß genug ist und das System die richtigen Satzfragmente kombinieren kann, klingt eine solche Sprachausgabe sehr natürlich. Diese Ausgabe ist in der Regel natürlicher als der künstliche Klang älterer Systeme, bei denen die Ausgabe auf einer Verkettung einzelner Phoneme („Wort-Schnipsel“) basiert, die von einem Sprecher aufgenommen oder künstlich erzeugt wurden und die gesamte gesprochene Sprache abdecken.

Der Eingabeteil eines Voice User Interface ist in der Regel ein Sprache-zu-Text-System, das mit Hilfe von Spracherkennung den Inhalt der gesprochenen Worte erkennt. Solche Eingabesysteme

sind in Callcentern schon lange im Einsatz, um den Kunden an zuständige Mitarbeiter weiterzuleiten. Moderne Anwendungen sind die Sprachassistenten (z. B. Google Assistant, Apple Siri oder Microsoft Cortana) in Smartphones oder Smart Speakern. In diesen Anwendungsbeispielen wurde die Spracheingabe und -ausgabe mit einem System der künstlichen Intelligenz kombiniert, um intelligente Unterstützung zu bieten. Solche Anwendungen für einzelne Benutzer - zum Beispiel für die Nutzung von Smartphones - funktionieren bereits recht gut. Allerdings ist die Qualität eines zwischenmenschlichen Gesprächs noch nicht erreicht. Beispielsweise sinkt die Erkennungsrate eines intelligenten Lautsprechers in einem Familienumfeld, in dem mehrere Personen mit dem Gerät kommunizieren. Auch die Erkennung der Sprache von sehr jungen Kindern (3-4 Jahre) scheitert oft, da sich ihr Sprachgebrauch deutlich von dem der Erwachsenen unterscheidet (vgl. [ShRP2019]).

Touch-Interface

Touch-Interfaces verwenden die Berührung des Benutzers als Eingabe. Touchscreens werden bereits seit langem eingesetzt, vor allem in Fahrkartenautomaten oder Geldautomaten (ATM). Sie wurden durch ihre Verwendung in Smartphones weit verbreitet. Frühe Touchscreens verwendeten eine einzelne Berührungseingabe als einzige verfügbare Eingabegeste, bei der der Benutzer seinen Finger zum Auslösen eines einzelnen Ereignisses verwendet. Die Fähigkeit, mehrere Fingerberührungen gleichzeitig (Multitouch) oder den Druck der Berührung zu erkennen, ermöglicht die Erkennung komplexerer Gesten. Zu diesen Gesten gehören Streichen (engl. swiping), Blättern (engl. flicking), mit zwei Fingern Auf- und Zuziehen (engl. pinching), Drücken oder Tippen. Eine in modernen Touchscreens häufig verwendete Technologie basiert auf der gemessenen Veränderung eines elektrischen Feldes auf dem Bildschirm, wenn sich die Finger des Benutzers dem Bildschirm nähern (engl. capacitive touch). Der größte Nachteil der meisten Touchscreens ist das fehlende taktile Feedback.

Gestenbasiertes Interface

Touchscreens können nur Fingergesten erkennen, wenn die Finger das Gerät berühren. Sollen Gesten einer ganzen Hand oder eines Arms erkannt werden, müssen zusätzliche Technologien wie Tiefensensoren oder Kameras mit einer entsprechenden Software eingesetzt werden (z. B. Nintendo Wii, Microsoft Kinect oder Leap Motion Controller). Eine einfache gestenbasierte Interaktion kann durch die Verwendung der in modernen Smartphones eingebauten Sensoren (z. B. Gyroskope) realisiert werden. In der Vergangenheit fand viel Forschung zu Gesten und ihre Verwendung in technischen Systemen statt. Eine wichtige Erkenntnis ist die Notwendigkeit einer bestimmten Syntax wie in einem gesprochenen oder geschriebenen Satz zu folgen, um eine Geste zu verstehen. Um z. B. die Lautstärke des Fernsehers zu erhöhen, muss die linke Hand auf den Fernseher zeigen, dann wird die rechte Hand erhoben, um die Geste für mehr Lautstärke mitzuteilen. Interessante Anwendungen für gestenbasierte Systeme sind Anwendungen für Chirurgen, wenn sie sich in einem Operationssaal befinden und keine unsterilen Gegenstände, wie z. B. eine Maus oder einen Touchscreen, berühren dürfen. Für diese Art von Szenario wurde eine Anwendung entwickelt, die eine Navigation durch eine Reihe von Computertomographie- (CT) oder Magnetresonanztomographie-Aufnahmen (MRT) auf der Grundlage von Gesten ermöglicht. Ein Problem, das bei gestenbasierten Systemen immer noch nicht gelöst ist, ist die zuverlässige Erkennung des Beginns und des Endes einer Geste und die Unterscheidung einer bedeutungsvollen Geste von einer Geste, die lediglich die gesprochenen Worte innerhalb eines Gesprächs unterstützt. (vgl. [ShRP2019]).

Haptische Interface

Haptische Schnittstellen geben eine taktile Rückmeldung, z. B. durch Vibration oder eine andere Form der entgegenwirkenden Kraft. Eine Braillezeile, die für sehbehinderte Benutzer entwickelt wurde, kann ein detailliertes Relief auf einer Oberfläche erzeugen, das die Darstellung von Blindenschriftzeichen ermöglicht, die sich mit den Fingerspitzen ertasten lassen. Die Blindenschriftzeichen werden in der Regel mit runden Stiften erzeugt, die durch Löcher in einer ebenen Fläche geführt werden. Vibrationsaktoren können in Smartwatches oder Wearables integriert werden. Werden sie in die Kleidung integriert, kann zum Beispiel eine Umarmung durch eine andere Person simuliert werden. Dies wurde in einer anderen Anwendung zur Unterstützung beim Erlernen eines Instruments demonstriert. Ultrahaptics ist eine Technologie, die Ultraschall verwendet, um virtuelle Elemente in der Luft zu erzeugen. Damit lässt sich eine virtuelle Taste erstellen, die bei Bedarf gedrückt wird, und die der Benutzer fühlen kann. Wenn haptische Aktoren in ein Exoskelett eingebaut werden, können sie (behinderten) Personen beim Bewegen, Gehen oder Trainieren helfen (vgl. [ShRP2019]).

Tangible Interface

Bei einem Tangible User Interface (anfassbares User Interface) werden Sensoren in physische Objekte integriert, die z. B. die Position, Geschwindigkeit oder Beschleunigung des jeweiligen Objekts erfassen. Technologien für diesen Zweck sind Bewegungs- oder Beschleunigungssensoren oder RFID-Einheiten (zur Identifikation von Objekten durch Funktechnologie). Die Manipulation eines solchen physischen Objekts bewirkt einen (digitalen) Effekt in der Anwendung. Ein bekanntes Beispiel sind die Bewegungssensoren in Spiele-Controllern.

Eine Tangible User Interface hat nicht notwendigerweise einen einzigen Kontrollpunkt für die Interaktion, wie es bei einem mausgesteuerten grafischen User Interface der Fall ist. Mehrere Objekte können gleichzeitig manipuliert werden, z. B. von mehr als einer Hand und möglicherweise von mehreren Benutzern. Im Prinzip erzwingt ein Tangible User Interface keine strikte Befehlsfolge oder modale Interaktion. Diese Freiheit trägt dazu bei, den kognitiven Prozess des Benutzers für das Verständnis und zur Steuerung des User Interface zu verbessern. Ein Beispiel für ein komplexes Tangible User Interface ist eine Tischplatte mit physischen Objekten und digitalen Informationen, die auf den Tisch projiziert werden. Werden die Objekte manipuliert (z. B. verschoben), so werden die digitalen Informationen verändert. Solche Schnittstellen werden in der Stadtplanung verwendet, wo maßstabsgetreue Gebäudemodelle auf dem Tisch bewegt werden und die künstlich erzeugten Effekte, wie Wind und Schatten, durch eine digitale Simulation beobachtet werden können. Eine Variante interaktiver Tische sind Tische, bei denen die Tischplatte ein großes Touch-Display ist (z. B. Microsoft PixleSense, früher Microsoft Surface). Es ist wichtig, dass solche Displays in der Lage sind, mehrere Berührungen gleichzeitig an unterschiedlichen Positionen zu erkennen. Weitere Anwendungen für ein Tangible User Interface sind Baukästen für Lernen der Programmierung oder der Entwicklung elektronischer Schaltungen für MINT-Fächer (Mathematik, Informatik, Naturwissenschaft und Technik). Beispiele sind kleine (preiswerte) Ein-Chip-Computer, die das entdeckende Lernen, die Erforschung und die Zusammenarbeit mit anderen bei der Erstellung kleiner Anwendungen für solche Geräte ermöglichen. Es sind auch spezielle Baukästen für sehbehinderte Menschen verfügbar, in denen die Tangible User Interfaces verwendet werden (vgl. [ShRP2019]).

Brain-Computer Interface

Bei einer Gehirn-Computer-Schnittstelle steuern die Benutzer den Computer mit ihren Gedanken. Die Neuronen im menschlichen Gehirn tauschen untereinander elektrische Signale aus, die sich verändern, wenn die Person denkt, sich bewegt oder fühlt. Diese Signale werden mit speziellen Headsets, Haarnetzen oder Mützen aufgezeichnet. Um eine Gehirn-Computer-Schnittstelle zu nutzen, werden die Benutzer darauf trainiert, sich auf eine Aufgabe zu konzentrieren, um so den Computer zu steuern. In diesem Bereich wurde bereits viel geforscht, aber Produkte für den Massenmarkt sind noch nicht verbreitet. Die Anwendungen umfassen die Steuerung von Spielen oder elektronischen Geräten in Echtzeit. In einer medizinischen Forschungsanwendung helfen solche Schnittstellen gelähmten Patienten, autonom zu werden (vgl. [ShRP2019]).

Mixed Reality

Milgram und Kishino [MiKi1994] ordnen visuelle Darstellungen in das sogenannte Virtuality Continuum ein (Kontinuum der Virtualität ein (siehe Abbildung 49)).

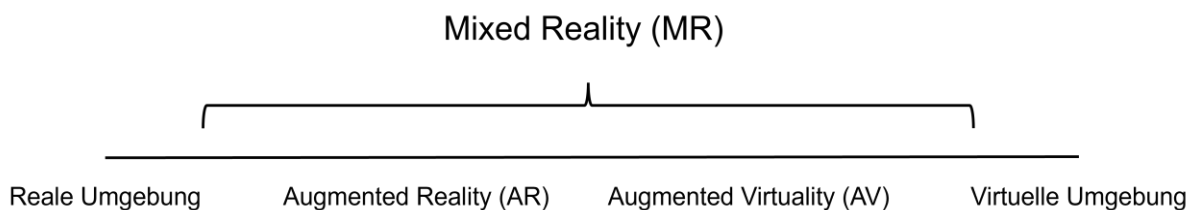


Abbildung 49 - Das Virtuality Continuum nach Milgram und Kishino [MiKi1994]

An einem Ende dieses Kontinuums befindet sich die reale Umgebung (Real Environment), d. h. die Realität, und am anderen Ende eine vollständig virtuelle Umgebung (Virtual Environment). Dazwischen wird den Benutzern eine Kombination aus virtuellen Elementen und Realität präsentiert, die als *Mixed Reality (MR)* bezeichnet wird. Ein Beispiel für MR ist das Konzept der Augmented Reality (AR), bei dem die reale Welt bzw. Bilder der realen Welt virtuell um künstliche Elemente erweitert werden. Bei einer Augmented Virtuality (AV) wird eine virtuelle Umgebung um Elemente der realen Welt erweitert.

Derzeit gibt es zahlreiche User Interface Technologien, die sich in die Kategorien Virtual Reality (VR) und Augmented Reality (AR) einteilen lassen. Weitere Einzelheiten zu VR und AR werden im Folgenden erläutert.

Virtual Reality (VR)

Virtual-Reality-Technologien zielen darauf ab, eine virtuelle Umgebung zu schaffen, in die sich ein Benutzer begeben kann. In vielen Fällen wird der Sehsinn durch ein stereoskopisches Display stimuliert, das jedem Auge zwei separate Bilder präsentiert und einen dreidimensionalen Seheindruck ermöglicht. Solche Displays werden in sogenannten VR Headsets verwendet. Bekannte Beispiele hierfür sind die HTC Vive oder die Oculus Quest. Andere Bildschirme, wie z. B. LCD-Panels oder Projektionen, können zusammen mit speziellen Brillen verwendet werden, um ebenfalls eine stereoskopische Präsentation zu ermöglichen. Neben dem visuellen Sinn können VR-Technologien auch den akustischen Sinn durch dreidimensionale Audioinhalte stimulieren. Im Gegensatz zu herkömmlichen Stereodarstellungen können die Audioinhalte an beliebigen Positionen im dreidimensionalen Raum platziert werden. Dies ist auch für mehrere Zuhörer möglich. Es ist ebenfalls möglich, andere menschliche Sinne zu stimulieren, zum Beispiel durch haptische Interfaces (siehe oben) oder olfaktorische Interfaces, die Gerüche simulieren.

Um ein gutes virtuelles Erlebnis zu erzeugen, ist es wichtig, eine Interaktion mit der virtuellen Umgebung zu ermöglichen, d.h. der Kopf und der Körper des Benutzers muss so verfolgt (tracked) werden, dass durch jede Kopfbewegung oder Änderung der Körperposition die dargestellte Szene entsprechend angepasst wird. Der Benutzer muss in der Lage sein, (einige) der virtuell dargestellten Objekte der Szene zu manipulieren, z. B. mit Hilfe von Standardeingabegeräten bzw. tangible Interfaces oder gestenbasierte Interfaces. In diesem Zusammenhang beschreibt der Begriff *Immersion* das Ausmaß, in dem die menschlichen Sinne durch die Technologie der virtuellen Realität stimuliert werden. Wenn der Grad der Immersion hoch genug ist und der Benutzer sich wirklich auf die virtuelle Umgebung einlassen will, kann der Benutzer *Präsenz* erreichen, d. h. den psychologischen Zustand, dass er sich wirklich in der virtuellen Umgebung *befindet*. In der Regel ist das Erreichen von Präsenz das Ziel bei der Erzeugung einer virtuellen Umgebung.

Häufig wird davon angenommen, dass virtuelle Umgebungen von einem höheren Grad an Realismus profitieren. Es ist jedoch bekannt, dass Realismus nicht immer zu einer besseren Präsenz führt. Selbst einfache polygonartige oder cartoonartige virtuelle Umgebungen können eine starke Immersion und Gegenwart erzeugen. Beispiele sind die Arcade-Spiele aus den 1980er Jahren oder Disney-Zeichentrickfilme.

Es gibt viele Anwendungsbereiche für Virtual-Reality-Technologien. Sie eignet sich gut für Lern- und Trainingsanwendungen, z. B. für die Bedienung von Flugzeugen oder die Simulation von Auto- oder Zugfahrten. Ein Museum könnte mit Hilfe von VR Umgebungen vergangener Zeiten auf eine viel immersivere Weise zeigen. Architekten können komplette Gebäude virtuell entwerfen und sie gemeinsam mit ihren Auftraggebern begehen, bevor sie gebaut werden. Auch von komplexen technischen Systemen wie einem Raumfahrzeug, einem Kraftfahrzeug oder einer verfahrenstechnischen Anlage lassen sich virtuelle Prototypen entwickeln. Damit können die Funktionen des Produkts simuliert und optimiert werden, bevor das Produkt entsteht.

Chirurgen können an virtuellen Patienten geschult werden, bevor die reale Operation stattfindet. Es gibt Therapieanwendungen, die helfen, Phobien - zum Beispiel die Angst vor Spinnen - durch eine Konfrontationstherapie zu mildern oder zu heilen. VR-Anwendungen können dazu beitragen, Stress bei Gesprächen in der Öffentlichkeit abzubauen oder posttraumatische Belastungsstörungen zu reduzieren. Wenn sich der Benutzer durch die virtuelle Realität in die Situation einer anderen Person hineinversetzt, kann Empathie für die Person entstehen, die sich tatsächlich in dieser Situation befindet, z. B. für einen Geflüchteten, der in ein fremdes Land und eine fremde Gesellschaft fliehen musste.

Viele Computerspiele wurden für die Verwendung mit VR-Technologien angepasst; einige davon sind Multi-User-Spiele. Komplexe Simulationen finden sich in Vergnügungsparks - zum Beispiel eine Simulation eines Fluges zum Mars. Einige Fluggesellschaften nutzen VR für die Unterhaltung während des Fluges oder zur Werbung für bestimmte Flugziele der Fluggesellschaft.

Ein wesentliches Problem der virtuellen Realität ist die VR-Krankheit (Virtual Reality Sickness), insbesondere die Bewegungskrankheit (Motion Sickness), die bei Benutzern auftritt. Ein Grund für diesen Effekt ist zum Teil die unvollkommene Technologie, die vor allem in komplexen Umgebungen zu zeitlichen Verzögerungen (Latenzen) führt. So wird beispielsweise der Kopf bewegt, und es dauert eine gewisse Zeit, bis der Benutzer die Reaktion auf diese Bewegung in der virtuellen Szene sieht. Es gibt jedoch auch grundsätzliche Probleme, die zu Unterschieden in der Wahrnehmung virtueller und realer Szenen führen, zum Beispiel, wenn die virtuelle Szene

eine stark beschleunigte Bewegung des Benutzers mithilfe eines optischen Flusses zeigt, wie etwa fliegende Sterne, der Benutzer aber in Wirklichkeit stillsteht. In dieser Situation nimmt der visuelle Sinn Bewegung wahr, während der vestibuläre Sinn keine Bewegung wahrnimmt, was eine Wahrnehmungsdiskrepanz darstellt, die wiederum zur Bewegungskrankheit führen kann. Solange sich der Benutzer während der Simulation in der Realität nicht genauso bewegen kann, wie es in der VR simuliert wird, kann diese Situation nur durch eine sorgfältige Gestaltung der virtuellen Umgebung gelöst werden.

Es sind weiterhin Forschungsfragen offen: Zum Beispiel, ob es notwendig und wichtig ist, eine Repräsentation des Selbst, d.h. virtueller Körperteile des Benutzers, in der Umgebung zu haben oder nicht. Derzeit wird an der effektivsten Bewegungsnavigation in der VR geforscht, die das Risiko der Bewegungskrankheit verringert. Auch die besten Paradigmen und die besten Technologien für Interaktionen und Bewegungen - z. B. die Verwendung von Kopf- und Körperbewegungen, die Verwendung von Tastaturen, Zeigegeräten oder Joystick-Tasten - sind noch nicht vollständig bekannt. Außerdem wird noch immer erforscht, wie in virtuellen Umgebungen am besten zusammengearbeitet und mit anderen kommuniziert werden kann.

Aus der Spieleindustrie stehen professionelle 3D-Werkzeuge zur Verfügung, mit deren Hilfe sich immersive virtuelle Umgebungen entwickeln lassen, und diese werden ständig weiterentwickelt. Beispiele hierfür sind die Unity Engine, die Unreal Engine oder die CryEngine (vgl. [ShRP2019] und [Jera2016]).

Augmented Reality (AR)

Augmented Reality ist ein weiteres Beispiel für Mixed-Reality (siehe Abbildung 49). In der AR wird die Realität durch dreidimensionale künstliche Elemente erweitert, und der Benutzer kann sowohl mit der Realität als auch mit den virtuellen Elementen in Echtzeit interagieren. Grundsätzlich sind drei Arten der Erweiterung (Augmentierung) möglich. Die häufigste Form ist *Video See Through AR*. In diesem Fall nimmt eine Kamera Bilder der realen Welt auf, und die künstlichen Elemente werden diesen Bildern rechnerisch überlagert. Das Ergebnis wird dem Benutzer über ein Display angezeigt. Dazu kann ein Smartphone oder ein spezielles Video See Through Headset verwendet werden, das im Wesentlichen ein mit einer Kamera ausgestattetes VR Headset ist. Wenn ein Headset verwendet wird, kann die Anzeige stereoskopisch sein. *Optical See Through AR* präsentiert dem Benutzer die Realität durch ein halbtransparentes Display oder einen Spiegel. Die künstlichen Elemente werden durch das halbtransparente Display oder die Projektion auf den Spiegel hinzugefügt. Ein Beispiel für ein Optical See Through Headset ist die Microsoft HoloLens. Durch die *Spatial Augmented Reality* werden künstliche Informationen per Video auf Objekte der realen Welt projiziert. Dies kann z. B. genutzt werden, um einen Kletterer mittels einer Projektion auf eine Kletterwand durch eine Tour zu führen (siehe [SchHo2016]).

Die vielleicht beliebteste AR-Anwendung ist das AR-Spiel Pokémon Go aus dem Jahr 2016, bei dem die Spieler nach virtuellen Figuren suchen müssen, die auf ihren Smartphones angezeigt werden, wenn sich der Spieler und sein Smartphone an einem bestimmten Ort befinden. Ein großer Anwendungsbereich von AR ist die Medizintechnik. Medizinische Bilder wie Röntgenbilder, Computertomographie (CT) oder Magnetresonanztomographie (MRT) können auf den Körper von Patienten projiziert werden, um eine Operation zu planen oder sogar um Chirurgen während einer Operation zu unterstützen. In diesem Bereich gibt es mehrere Therapie- und Trainingsanwendungen. Überlagerte Diagramme können Technikern helfen, Fehler in komplexen Geräten oder Installationen in Gebäuden schnell zu finden und zu reparieren. AR kann Fluglotsen und Betreibern bei der schnellen Entscheidungsfindung helfen. So verfügen

beispielsweise Fluglotsen über Bildschirme mit eingeblendeten Informationen über Flugbewegungen. Head-up-Displays (HUD), Optical See Through Displays in militärischen und zivilen Flugzeugen, können wichtige Informationen, z. B. über die Wetterbedingungen, anzeigen. In modernen Autos sind HUDs in die Windschutzscheibe integriert, durch die zum Beispiel Navigationsinformationen angezeigt werden. Solche Navigationsinformationen lassen sich auch auf einem Smartphone zusammen mit realen Bildern der Umgebung anzeigen, um (touristisch) interessante Orte in einer (fremden) Stadt zu finden.

Eine beliebte Unterhaltungsanwendung sind die Filter der Social-Media-Anwendung Snapchat. Die Benutzer können ihr eigenes Bild verändern oder verformen oder dem Originalbild Elemente hinzufügen, wie z. B. große Ohren oder Halsketten. Diese Technologie eines AR-Spiegels kann zum Beispiel für die virtuelle Anprobe von Sonnenbrillen, Schmuck oder für Make-up verwendet werden. Der Nachteil ist, dass die virtuellen Elemente nur gesehen und nicht physisch erlebt werden können.

Zu den Herausforderungen der AR gehört, an welchen Stellen Informationen eingeblendet werden sollen und wie viele Informationen angezeigt werden können weit begrenzt werden können, so dass sie für den Benutzer noch nützlich sind und ihn nicht ablenken oder kognitiv überfordern. Zu den technologischen Herausforderungen gehören die genaue Erfassung der realen Umgebung und die perfekte Integration (Registrierung) der virtuellen Elemente in die Realität. Die geometrische Anpassung (geometrische Registrierung) der virtuellen Objekte an die Realität - die mit Hilfe von Tracking-Technologien erfolgt - funktioniert bereits recht gut, aber die Anpassung an die Lichtverhältnisse (photometrische Registrierung) ist für die meisten AR-Anwendungen noch eine Herausforderung. Eine geringere Qualität der überlagerten Elemente mag für Unterhaltungszwecke akzeptabel sein, für militärische und medizinische Anwendungen muss die Qualität jedoch hoch sein. Insbesondere wenn sich der Benutzer oder die Szene bewegt, wird möglicherweise viel Rechenleistung benötigt, um den Überblick in Echtzeit zu behalten (siehe [ShRP2019] und [ScHo2016]).

Ambient Interface

Durch die Verwendung der oben beschriebenen Schnittstellentypen interagiert ein Benutzer explizit und absichtlich mit einem System. Heutzutage gibt es immer mehr Situationen, in denen ein Benutzer nur implizit oder ungewollt mit einem digitalen System interagiert. Diese Interaktionen mögen den Anschein erwecken, dass das digitale System automatisch und ohne Eingaben des Benutzers agiert, aber das ist nicht der Fall; es reagiert auf eine vordefinierte Weise auf die Aktionen des Benutzers. Wenn ein Benutzer beispielsweise seine Wohnung betritt, kann das Hausautomatisierungssystem (Smart Home) mehrere Lampen und andere Geräte einschalten und sogar anspruchsvollere Aktionen durchführen. Um solche Funktionen zu ermöglichen, müssen die digitalen Systeme die aktuelle Situation mit Hilfe von (hochentwickelten) Sensoren – des Ambient Interface – erkennen. Daher sollte der DDP zugrundeliegende Sensortechnologien verstehen, um sie in das Design einer digitalen Lösung einbeziehen zu können. Beispiele für solche Technologien sind Audiosensoren (z. B. Mikrofone), Video- und Lichtsensoren (z. B. Kameras, Infrarotdetektoren), Positionssensoren (z. B. GPS, GNSS), Beschleunigungsmesser, Gyroskope und Druck- oder Temperatursensoren.

Kombination von User Interface Typen

Alle oben beschriebenen Paradigmen und Typen von User Interfaces können kombiniert werden, um das entsprechende User Interface auf die Zielanwendung zuzuschneiden. So beinhalten

Augmented- oder Virtual-Reality-Anwendungen häufig Sprach-, Bewegungs- oder Gesteninteraktionsmöglichkeiten.

Einige dieser Schnittstellentypen werden bereits in standardisierten Endgeräten verwendet. Die Auswahl eines User Interface Typs für ein kundenspezifisches Endgerät ist eine wichtige Entscheidung für die Gestaltung einer digitalen Lösung und kann im weiteren Verlauf des Bauprozesses nur mit hohem Aufwand revidiert werden. Wenn verschiedene Interaktionsformen und Schnittstellentypen in Betracht gezogen werden, sollte die Eignung dieser Alternativen für eine digitale Lösung durch den Einsatz von Prototypen (siehe 2.3) untersucht werden, um das Risiko einer Fehlentscheidung frühzeitig zu reduzieren.

3.2.3 Software-User-Interface-Technologie

Die eingesetzte Technologie für ein User Interface (UI) fällt in die Kategorie der Software für die wahrnehmbare Form und Funktion. So wird durch ein User Interface sowohl der visuelle Aufbau (Form) als auch das dynamische Verhalten des User Interface (Funktion) bestimmt. Beispiele hierfür sind folgende:

- Funktionen der Fenster, Scrollen, Zoomen
- Sprachsynthese
- Sprach- und Gestenerkennung
- Software-gestützte Metaphern, wie z. B. Pick, Drag-and-Drop
- Software-gestützte virtuelle Geräte, wie z. B. virtuelle Tastaturen, virtuelle Schieberegler oder virtuelle analoge Instrumente

Die UI-Technologie für andere sensorische Modalitäten fällt ebenfalls in diese Kategorie. Derzeit sind neben visuellen User Interfaces auch Technologien für die Audio- und Haptik-Eingabe und -Ausgabe wichtig.

Um Anwendungen auf den in Abschnitt 3.2.1 genannten Gerätekategorien auszuführen, existieren Betriebssysteme, wie Android, iOS oder Windows. Eine Kombination, die hauptsächlich aus einem Gerät und einem Betriebssystem besteht, wird als Computing-Plattform oder Plattform bezeichnet.

Technologien zur Entwicklung von User Interfaces

Für die am häufigsten verwendeten Plattformen gibt es Software-Entwicklungsumgebungen, die die Erstellung von grafischen, akustischen oder haptischen User Interfaces unterstützen. Sie bieten Bibliotheken an, die über eine Programmierschnittstelle (API) einen einfachen Zugriff auf die Funktionen des Geräts ermöglichen. Zu diesen Funktionen gehören die Anzeige von Inhalten auf dem Display, die Erkennung von Berührungseignissen, die Wiedergabe und Aufnahme von Audiodaten über den Lautsprecher und das Mikrofon, die Aktivierung der Vibrationsfunktion oder die Verwendung des Sprachassistenten.

Bei den meisten Geräten ist die grafische Displayausgabe der komplizierteste und größte Teil der Schnittstelle. Moderne Entwicklungsumgebungen bieten grafische Editoren an, durch die sich Layouts für die Anzeigeinhalte einfach per Mausinteraktion zusammenstellen lassen. Vordefinierte visuelle Elemente, wie z. B. Schaltflächen (engl. buttons), Texteingabefelder (engl. text input fields), Listen- oder Bildansichten (engl. list or image views), stehen zur Verfügung, um schnell und ohne zusätzlichen Programmieraufwand ein Layout für den Ausgabebildschirm zu erstellen. Dies trägt dazu bei, ein ähnliches Erscheinungsbild (engl. look and feel) innerhalb einer

Anwendung und zwischen verschiedenen Anwendungen auf einem Gerät zu schaffen. Solche Editoren erlauben in der Regel ein flexibles Layout. Dieses flexible oder automatische Layout verschiebt und skaliert die Bildelemente (zur Laufzeit) entsprechend, so dass die Anwendung unabhängig von dem Gerät, auf dem sie ausgeführt wird, ein ansprechendes Erscheinungsbild hat. Die Plattformen unterstützen in der Regel Geräte mit unterschiedlichen Displaygrößen und -auflösungen, was eine solche automatische Anpassung notwendig macht, um den Entwicklungsaufwand zu verringern.

Wenn ein Bildschirmlayout in einem solchen grafischen Editor erstellt wird, wird die visuelle Komposition automatisch in eine programmierbare Beschreibung übersetzt, die von einem Programmierer verändert und in die Funktionalität der Anwendung integriert werden kann, d. h. es steht eine Schnittstelle zur Verfügung, die sich aus dem Quellcode der Anwendung nutzen lässt. Die Extensible Markup Language (XML) ist für eine solche Beschreibung beliebt, aber auch andere Auszeichnungssprachen oder native Programmiersprachen wie C#, C++ oder Java können verwendet werden.

Diese direkte Übersetzung der visuellen Beschreibung in Programmcode ermöglicht eine Arbeitsteilung in Entwicklungsprojekten zwischen einem User Interface- oder Interaktionsdesigner und einem Programmierer. Darüber hinaus trägt sie dazu bei, dass sich weder ein User Interface Designer noch ein Programmierer um die jeweils andere Disziplin kümmern oder gar in ihr arbeiten muss.

Moderne Entwicklungsumgebungen für die Anwendungsentwicklung für Massenmarktgeräte wie Smartphones, Notebooks und Smartwatches sind sehr gut dokumentiert, um die Hürde für den Einstieg in die Entwicklung für eine Plattform zu senken. Dies ermöglicht einen einfachen Einstieg für Entwickler, die wenig oder keine Erfahrung mit der jeweiligen Plattform haben, oder sogar für nicht professionelle Entwickler.

Arten von Anwendungen

Für die am häufigsten genutzten Plattformen Android, iOS, macOS und Windows können wir zwischen vier Arten von Anwendungen unterscheiden: native, Web-, Hybrid- und plattformübergreifende Anwendungen.

Bei *nativen Anwendungen* handelt es sich um Anwendungen, die für eine einzige Zielplattform unter Verwendung der für diese Plattform erstellten Entwicklungsumgebung entwickelt werden. So wird beispielsweise das Google Android Studio für Android-Anwendungen, Apple Xcode für iOS-Anwendungen und Microsoft Visual Studio für Windows-Anwendungen verwendet, wobei die jeweilig spezielle(n) Programmiersprache(n) zum Einsatz kommen. Die ausführbaren Dateien solcher Anwendungen funktionieren nur auf der jeweiligen Plattform und sind sehr gut in die Ausführungsumgebung integriert. Die Anwendung kann sehr gut in das User Interface des Betriebssystems integriert werden, und es besteht ein hervorragender Zugriff auf die Ressourcen des Geräts, wie z. B. das Mikrofon oder den GPS-Sensor. Die Leistung lässt sich für die jeweilige Plattform und sogar für bestimmte Geräte optimieren. Die Entwicklungsumgebungen für moderne Plattformen bieten verschiedene Möglichkeiten, die Anpassung des User Interface an die individuellen Eigenschaften (z. B. die Displaygröße und -auflösung) des Gerätes innerhalb der jeweiligen Plattform zu unterstützen. Das Ergebnis dieses Mechanismus ist ähnlich zu dem responsiven Webdesign für Webanwendungen.

Webanwendungen sind im Wesentlichen Websites, die für die Ausgabe auf einem bestimmten Gerät zugeschnitten sind. Wenn die Anwendung im Browser des Geräts ausgeführt wird, stehen

Mechanismen zur Verfügung, um die Eigenschaften des Geräts zu erkennen, z. B. das Betriebssystem, die Größe und Auflösung des Bildschirms und Geräteeigenschaften. Mit diesen Informationen kann sich die Webanwendung an das Gerät anpassen und z. B. die Anzeigeelemente entsprechend skalieren. So kann beispielsweise dieselbe Webanwendung auf einem Notebook mit einem 17-Zoll-Display und auf einem Smartphone mit einem 5-Zoll-Display genutzt werden. Wenn die Webanwendung diesen Mechanismus unterstützt, wird sie als responsive Webanwendung bezeichnet. Das Design für solche responsiven Anwendungen wird als responsives (Web-)Design bezeichnet.

Die am häufigsten verwendeten Technologien zur Realisierung von Webanwendungen sind die Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) und JavaScript (JS). HTML ist eine Sprache – ähnlich wie eine Programmiersprache –, durch die der Inhalt semantisch strukturiert werden kann. Es lässt sich festlegen, wie solche Inhalte auf dem Display eines Ausgabegeräts formatiert werden sollen. So werden beispielsweise spezielle Tags verwendet, um festzulegen, wo ein Absatz beginnt und endet, welcher Textteil eine Überschrift ist und welche Textteile hervorgehoben werden sollen (z. B. in Fettdruck). CSS ist eine Sprache, mit der sich die Darstellung von Dokumenten beschreiben lässt. Sie kann zum Beispiel die Farben und die Schriftarten festlegen. CSS hilft dabei, die Darstellung des Inhalts vom eigentlichen Inhalt zu trennen. JavaScript¹³ ist eine Programmiersprache, die zur Definition interaktiver Elemente auf Websites verwendet wird. Neben diesen drei grundlegenden und weit verbreiteten Technologien gibt es viele weitere Webtechnologien mit zusätzlichen Funktionen. Es gibt zum Beispiel viele JavaScript-Frameworks (z. B. als Plug-Ins), die verschiedene spezielle Zwecke für Webanwendungen erfüllen, wie z. B. Diagramm-/Grafikmodellierungs-Frameworks (z. B. GoJS), Fragebogen-Frameworks (z. B. formr) oder Formular-Frameworks (z. B. formanizr).

Da Webanwendungen in einem Browser laufen, der für eine Zielplattform implementiert und optimiert wurde, um HTML, CSS und JavaScript effizient zu verarbeiten, sind sie sehr flexibel und sind auf verschiedenen Geräten und Betriebssystemen lauffähig. Andererseits benötigen reine Webanwendungen meist eine ständige Internetverbindung zu einem Webserver. Es ist möglich, einige Daten lokal auf dem Gerät zu speichern, aber im Vergleich zu nativen Anwendungen ist dieser Speicher (Webpace) recht begrenzt. Da diese Anwendungen und ihre Daten auf einem Server gehostet werden, ist es einfach, diese Anwendungen auf dem neuesten Stand zu halten, indem nur der Quellcode auf dem Server geändert wird. Die aktualisierte Anwendung wird automatisch heruntergeladen, wenn die Webanwendung das nächste Mal gestartet wird. Im Vergleich zu nativen Anwendungen ist der Zugang zu bestimmten Gerätefunktionen eingeschränkt. Webanwendungen sind im Grunde ein Link zu einer Website und daher nicht in App-Stores erhältlich.

Hybride Anwendungen sind eine Kombination aus nativen Anwendungen und Webanwendungen. Eine hybride Anwendung lässt sich durch die Verwendung von Webtechnologien zur Programmierung der Anwendung und die Ausführung dieses Codes in einem Browser Control realisieren. Dieses Browser Control ist ein Element des User Interface der Zielplattform und kann in eine native Anwendung (Application Base) integriert werden. Für die Erstellung solcher Anwendungen gibt es Werkzeuge, die automatisch native ausführbare Dateien für (mehrere) ausgewählte Plattformen erstellen. Das bedeutet, dass aus einer einzigen Codebasis mehrere native Anwendungen für verschiedene Plattformen erstellt werden können. Hybride

¹³ Die Programmiersprache JavaScript unterscheidet sich von der populären Programmiersprache Java.

Anwendungen können genauso flexibel sein wie Webanwendungen, verfügen über mehr lokalen Speicherplatz und bieten mehr Zugriff auf Gerätefunktionen als Webanwendungen. Andererseits sind sie weniger für eine bestimmte Plattform optimiert und ihre Leistung kann daher im Vergleich zu einer nativen Anwendung geringer sein. Heute befinden sich viele Anwendungen in App Stores, die solche Hybridtechnologien verwenden.

Die Idee der *plattformübergreifenden Anwendungen* (Cross-Plattform Applications) besteht darin, eine einzige Codebasis für die Implementierung der Anwendung zu verwenden. Die Programmiersprachen C#, C++ und JavaScript werden häufig für die Entwicklung solcher Anwendungen verwendet. Ein spezielles Werkzeug (z. B. Xamarin) erstellt automatisch native Anwendungen auf der Grundlage eines solchen Quellcodes. Im Gegensatz zu hybriden Anwendungen wird dieser Quellcode nicht in einem Browser ausgeführt. Plattformübergreifende Anwendungen verwenden in der Regel native Elemente des User Interface der Zielplattformen. Mit dieser Technologie können etwa 75 % des Quellcodes auf verschiedenen Plattformen gemeinsam genutzt werden. Plattformübergreifende Anwendungen haben alle Vorteile hybrider Anwendungen und bieten potenziell eine bessere Integration des User Interface in die Zielplattform und eine bessere Leistung als hybride Anwendungen. Andererseits ist die Codebasis (heute) nicht auf allen Plattformen identisch. Sie kann plattformabhängige, betriebssystemabhängige oder sogar geräteabhängige Quellcode-Teile erfordern.

Tabelle 17 fasst die Merkmale der verschiedenen oben erläuterten Ansätze zur Anwendungsentwicklung zusammen.

Tabelle 17 - Merkmale der verschiedenen Ansätze zur Anwendungsentwicklung

	Nativ	Web	Hybrid	Plattform- übergreifend
Zugang zu Geräteeigenschaften	hoch	niedrig	mittel	mittel
Performance	hoch	niedrig	niedrig	hoch
User Interface- Integration	hoch	niedrig	niedrig	hoch
Einfachheit der Entwicklung	niedrig	hoch	hoch	mittel
Bereitstellung in App- Stores	Ja	Nein	Ja	Ja
Wartbarkeit	niedrig	hoch	mittel	mittel

Spiele-Engines können als eine besondere Form eines plattformübergreifenden Werkzeugs zur Erstellung von Anwendungen betrachtet werden. Sie bieten Modellierungswerkzeuge zur Erstellung dreidimensionaler Umgebungen mit einem Minimum an Programmieraufwand, in der Regel mit Hilfe eines grafischen Editors. Falls erforderlich, sind *Skripte* (die programmierten Quellcode enthalten) eine leistungsstarke Möglichkeit, um den Anwendungen reichhaltige Interaktivität oder spezielle Funktionen hinzuzufügen. Eine einzige Codebasis kann automatisch in native Anwendungen umgewandelt werden, die auf mehreren verschiedenen Plattformen

lauffähig sind. Mit dieser Technologie wurden in der Vergangenheit bereits komplexe und erfolgreiche Spiele entwickelt. Spiele-Engines werden für die Entwicklung von Videospielen konzipiert und entwickelt, eignen sich aber auch für andere Anwendungen außerhalb der Spieleindustrie, beispielsweise für die Entwicklung von Augmented- oder Virtual-Reality-Anwendungen (siehe Abschnitt 3.2.2).

Praxistipp

Die Auswahl einer Software-User-Interface-Technologie ist eine wichtige Entscheidung. Sie definiert, welche Möglichkeiten und Anwendungseigenschaften zur Verfügung stehen, wie die Anwendung in Zukunft gewartet werden muss und welche Abhängigkeiten zu Drittanbieter-Werkzeugen bestehen. Der Austausch einer solchen User Interface Technologie zu einem späten Zeitpunkt im Bauprozess oder in einer bestehenden digitalen Lösung kann sehr teuer werden, da diese Technologie in der Regel sehr tief in einer digitalen Lösung verankert ist. Daher sollte die Technologie für das User Interface mit Sorgfalt ausgewählt und der DDP in den Auswahlprozess einbezogen werden, um die bestmögliche Technologie für die Gestaltung der Lösung zu finden.

Wenn die Erstellung von Prototypen des User Interface einer digitalen Lösung (Idee) über einfache horizontale Prototypen hinausgeht (siehe Abschnitt 2.3), sollte die zu diesem Zeitpunkt möglicherweise bereits ausgewählte User Interface Technologie so eingebunden werden, dass die Möglichkeiten der Technologie bei der Gestaltung des User Interface berücksichtigt werden. Dadurch wird verhindert, dass Entwürfe entwickelt werden, die nur mit unverhältnismäßig großem Aufwand mit der ausgewählten Technologie zu realisieren sind.

3.3 Zugrundeliegende Technologie

Das Verständnis der zugrundeliegenden Technologie ist für das Design einer digitalen Lösung ebenso wichtig wie das Verständnis der wahrnehmbaren Technologie. Der Hauptgrund dafür ist, dass die zugrundeliegende Technologie verwendet wird, um die Teile zu erstellen, die die wahrnehmbare Form, Funktion und Qualität einer digitalen Lösung ermöglichen (siehe Kapitel 1 und Abschnitt 3.1).

Die zugrundeliegende Technologie geht über die Teile einer digitalen Lösung hinaus, mit denen ein Benutzer direkt in Kontakt kommt. Dennoch kann die zugrundeliegende Technologie einen erheblichen Einfluss auf die Lösung oder das Systemdesign haben und somit die User Experience erheblich, wenn auch indirekt, beeinflussen. Ein DDP muss zumindest ein grundlegendes Verständnis der zugrundeliegenden Technologie haben, um die Teile des Designs zu identifizieren, die möglicherweise schwer zu realisieren sind. Darüber hinaus kann eine Entscheidung über einen bestimmten Architekturstil oder ein zu verwendendes Framework bereits von anderen Stakeholdern getroffen worden sein, z. B. von Systemarchitekten oder sogar von einem anderen DDP aus einem früheren Projekt, auf dem das aktuelle Projekt basiert. Der DDP muss diese Entscheidungen und ihre Folgen bei der Entwicklung der Lösung oder des Systems berücksichtigen. Die Kenntnis der zugrundeliegenden Technologie ermöglicht es dem DDP, mit Konstruktions- und Realisierungsexperten wie Softwarearchitekten oder Entwicklern zu sprechen und deren Argumente zu verstehen.

Um die Bedeutung der zugrundeliegenden Technologie zu verdeutlichen, lassen Sie uns noch einmal einen Blick auf unser YPRC-Beispiel werfen.

Beispiel YPRC. Ein wichtiges Merkmal der digitalen YPRC-Lösung ist die Remote-Coaching-Funktion. Obwohl die unmittelbare Funktion wahrnehmbar ist (der Trainer des Läufers gibt dem Läufer Ratschläge), werden mehrere wichtige Aspekte dieser Funktion durch die zugrundeliegende Technologie realisiert:

- Die Sprachkommunikation erfordert eine Datenverbindung zwischen der Läufer-App und dem Coaching-Portal (zugrundeliegende Form).
- Die Trainingsdaten des Läufers (Geschwindigkeit, Position, Gesundheitsdaten) müssen so erfasst (zugrundeliegende Funktion), gespeichert (zugrundeliegende Form), übertragen (zugrundeliegende Funktion) und aufbereitet (zugrundeliegende Form) werden, dass der Trainer sie für Trainingszwecke nutzen kann (wahrnehmbare Funktion).
- Der Kauf einer Coaching-Sitzung bedingt eine Zahlung. Die Zahlung selbst wird von einem Zahlungsanbieter durchgeführt. Die Verbindung mit einem Zahlungsanbietersystem ist ebenfalls Teil der zugrundeliegenden Form.

Selbst dieses scheinbar einfache Beispiel zeigt, wie wichtig es ist, die zugrundeliegende Technologie zu verstehen, um digitale Lösungen zu entwickeln. Auf Foundation Level Niveau sollte ein DDP die folgenden Aspekte der zugrundeliegenden Technologie kennen: Technologien zur Programmierung, Technologien für den Betrieb von Software und unterstützende Hardware. Auf jeden dieser Aspekte wird in den folgenden Unterabschnitten eingegangen.

3.3.1 Technologien zur Programmierung

Wir haben bereits Technologien zur Programmierung als Material für die Umsetzung der wahrnehmbaren Form und Funktion einer digitalen Lösung vorgestellt (siehe Abschnitt 3.2). Technologien zur Programmierung bestimmen aber auch einen wesentlichen Teil der zugrundeliegenden Form, Funktion und Qualität einer digitalen Lösung.

Einsteiger assoziieren Technologien zur Programmierung oft intuitiv mit Programmiersprachen. Dies führt häufig zu dem Missverständnis, dass das Schreiben von Programmcode die zentrale Aktivität bei der Entwicklung von Software ist. Dies ist vor allem darauf zurückzuführen, dass die Eingabe von Programmcode in Entwicklungsumgebungen die wichtigste sichtbare Tätigkeit von Softwareentwicklern ist. Die zentrale Tätigkeit bei der Entwicklung von Software ist jedoch vielmehr die Schaffung einer geeigneten Struktur in einer gewählten Technologie zur Programmierung, die die gewünschte Form, Funktion und Qualität tatsächlich realisiert. Diese Kompetenz gehört zu den Tätigkeitsbereichen Konstruktion und Realisierung und ist nicht Teil des Kompetenzprofils eines DDP auf Foundation Level Niveau.

Für einen DDP auf Foundation Level Niveau reicht das bereits eingeführte Verständnis der wahrnehmbaren und zugrundeliegenden Form, Funktion und Qualität als Grundlage aus, um die gewünschte digitale Lösung an Fachleute aus Konstruktion und Realisierung zu kommunizieren.

Technologien zur Programmierung gehen jedoch über Programmiersprachen hinaus. Sie bieten eine Vielzahl von Features, mit denen ein DDP auf Foundation Level Niveau vertraut sein sollte. Wir halten folgende Aspekte für wichtig: Technologien zur Datenspeicherung, Technologien für Software-Frameworks und API Technologien.

Technologien zur Datenspeicherung

Die Speicherung und Verarbeitung von Daten ist der Kern digitaler Lösungen und die Aufgabe von Technologien zur Datenspeicherung. Digitale Technologie hat verschiedene Ansätze und

Werkzeuge für diesen Zweck entwickelt. Wikipedia gibt einen guten Überblick über die Geschichte von Datenbanken [Wik2020a] und die verschiedenen verfügbaren Technologien [Wik2020b].

Praxistipp

Jede Technologie zur Datenspeicherung hat bestimmte Vor- und Nachteile in Bezug auf das zu verarbeitende Datenvolumen, die Geschwindigkeit der Operationen und die Flexibilität in Bezug auf Änderungen der Datenstrukturen. Wenn der DDP mit Technologieexperten über Technologien zur Datenspeicherung spricht, sollte er ein klares Bild von den Daten haben, die die digitale Lösung speichern und verarbeiten soll. Dieses Verständnis kann der DDP mit der Definition von Entitäten und Funktionen aus den Elementdesignkonzepten erhalten (siehe Abschnitt 2.2). Ein DDP sollte sich darüber im Klaren sein, dass die Speicherung von Daten kostspielig ist (z. B. aufgrund des Realisierungsaufwands oder der Einspeisung in die Cloud). Daher sollte ein DDP bei der Definition von Daten immer abwägen, ob die Daten wirklich benötigt werden und gespeichert werden müssen.

Technologien für Software-Frameworks

Software-Frameworks sind erfunden worden, um wiederverwendbare Komponenten bereitzustellen. Die Verwendung von Software-Frameworks verbessert die Qualität digitaler Lösungen, da gute Software-Frameworks bereits einen gründlichen Qualitätssicherungsprozess durchlaufen haben. Sie beschleunigen auch den Entwicklungsprozess, da die von den Software-Frameworks bereitgestellten Funktionen sofort einsatzbereit sind und nicht (erneut) manuell implementiert werden müssen.

Praxistipp

Die Auswahl des richtigen Software-Frameworks ist eine Aufgabe für Konstruktions- und Realisierungsexperten. Nichtsdestotrotz sollte sich ein DDP auf Foundation Level Niveau der Tatsache bewusst sein, dass Software-Frameworks wichtige zugrundeliegende Funktionen bereitstellen können, die für die Realisierung einer digitalen Lösung wichtig sein können. Im Folgenden stellen wir beispielhaft einige Software-Frameworks vor. Das Ziel der folgenden Liste ist es, einige interessante Frameworks für Einsteiger in Digital Design vorzustellen und die große Bandbreite der verfügbaren Software-Frameworks zu veranschaulichen:

- *Apache Hadoop* (<https://hadoop.apache.org>) ist ein Framework für die Verarbeitung großer Datensätze.
- *Camunda* (<http://www.camunda.org>) ist ein Framework zur Erstellung und Automatisierung von Arbeitsabläufen und Prozessen.
- *ElasticSearch* (<https://www.elastic.co/>) ist ein Framework zum Durchsuchen und Analysieren großer Datensätze.
- *Spring* (<https://spring.io>) ist ein Framework, das verschiedene Funktionen für die Entwicklung von Anwendungen (z. B. Kommunikation mit Datenbanken) bietet.
- *TensorFlow* (<https://www.tensorflow.org>) ist ein Framework für maschinelles Lernen und Teil der Technologie für künstliche Intelligenz.
- *Unity* (<https://unity.com>) ist ein Framework für die Entwicklung zwei- und dreidimensionaler Spiele und Anwendungen, einschließlich virtueller und erweiterter Realität (siehe Abschnitt 3.2).

Die Hauptbotschaft dieser Übersicht für den Leser lautet: Studieren Sie regelmäßig Frameworks, um zu wissen, was möglich ist. Ein DDP sollte sich bewusst sein, dass trotz der Vorteile der Verwendung von Software-Frameworks zur Einsparung der Realisierung von Standardfunktionen auch einige Nachteile der Wiederverwendung bestehen, da die wiederverwendbaren Funktionen möglicherweise nicht perfekt zu der aktuell in der Entwicklung befindlichen digitalen Lösung passen.

Praxistipp

Wenn der DDP zumindest ein grundlegendes Verständnis der bestehenden Software-Frameworks hat, kann er zusammen mit den Konstruktions- und Realisierungsexperten entscheiden, ob er bestehende Software-Frameworks verwendet oder stattdessen die Zeit und Mühe aufwendet, ein Software-Framework genau nach seinen Vorstellungen zu bauen. Außerdem binden Sie sich durch die Verwendung eines Software-Frameworks eines Drittanbieters an ein Unternehmen, das Sie nicht kontrollieren können. Das Unternehmen kann sein Preis- oder Lizenzmodell im Laufe der Zeit zu Konditionen ändern, die Sie möglicherweise nicht mehr akzeptieren können. Sie verlassen sich auch darauf, dass dieses Unternehmen sein Software-Framework gut pflegt und langfristig wettbewerbsfähig bleibt. Änderungen am Drittanbieter Software-Framework können Sie dazu zwingen, Ihr System entsprechend anzupassen, um sicherzustellen, dass es weiterhin ordnungsgemäß funktioniert oder überhaupt noch läuft. Dies kann zeitaufwendig sein und mit der Zeit sehr teuer werden.

API Technologien

Der Ansatz der API Technologie zielt darauf ab, Funktionen anzubieten, die über eine technische Schnittstelle in Ihre eigene digitale Lösung integriert werden können. Es gibt viele verschiedene Arten der Integration, aber wenn wir heute von der Nutzung oder dem Angebot einer API sprechen, beziehen wir uns in der Regel auf eine API, die über das World Wide Web genutzt wird. Ein technischer Name dafür ist *Web-Service*. In der YPRC-Fallstudie werden zwei Web-Services genutzt: der Kartenserver und der Zahlungsanbieter.

Ähnlich wie bei den Software-Frameworks gibt es auch hier eine große Anzahl verschiedener Web-Services. Nachfolgend finden Sie eine Liste einiger beispielhafter Web-Services:

- *GraphHopper* bietet Web-Services für die Routenplanung und -optimierung.
- *ApplePay*, *GooglePay* und *PayPal* bieten Web-Services für Zahlungstransaktionen an.
- *Melissa* ist ein Webservice Framework für die Überprüfung von Adressen.

Praxistipp

Der Hauptvorteil der API-Technologie besteht darin, dass die bereitgestellte Funktionalität sofort für eine digitale Lösung genutzt werden kann. Das Geschäftsmodell der digitalen Lösung muss jedoch die Kosten für die Nutzung von Web-Services berücksichtigen. Wie bei Software-Frameworks sollte sich ein DDP der gleichen Nachteile bewusst sein, die mit der Verwendung externer APIs einhergehen. Außerdem ist es viel wahrscheinlicher, dass man Software-Frameworks an die gewünschte digitale Lösung anpasst, als dass man die APIs anpasst, da diese vollständig unter der Kontrolle einer externen Partei stehen.

3.3.2 Technologien zum Betreiben von Software

Die Computertechnologie ist ein wichtiger Bestandteil der Infrastruktur für den Aufbau einer digitalen Lösung. Sie besteht aus der Computer-Hardware, mit Prozessoren, Arbeitsspeicher und

Datenspeicher als typische Komponenten, sowie dem Betriebssystem. Computer-Hardware wird meist in großen Stückzahlen als standardisiertes Massenprodukt gebaut.

Spezialisierte Hardware, z. B. zur Datenverschlüsselung, wird genutzt, wenn bestimmte Qualitätsanforderungen (insbesondere Geschwindigkeit und Sicherheit) mit Standard-Hardware nicht erreicht werden können.

Betriebssysteme werden benötigt, um die Computer-Hardware zu verwalten, grundlegende Software-Dienste wie die Organisation des Datenspeichers mit einem Dateisystem bereitzustellen und auch um eine Umgebung für die Ausführung von Anwendungssoftware zu bieten.

Hardware und ihre Betriebssysteme können auf verschiedene Weise bereitgestellt werden:

- Teil eines Standardgeräts (z. B. ein Smartphone oder White-Label-Komponenten)
- Teil eines kundenspezifischen Geräts (z. B. eines Do-it-yourself Smart-Home-Controllers)
- Lokaler Server (z. B. ein Desktop-Computer)
- Remote Server (z. B. in einem Rechenzentrum)
- Service auf Abruf über das Internet (Cloud Computing)

Aus der Perspektive des Digital Design ist es wichtig, das breite Spektrum der Technologien zu verstehen, die für den Betrieb einer digitalen Lösung zur Verfügung stehen, sowie die verschiedenen Ebenen der Kommunikationstechnologie, die für die digitale Lösung erforderlich sein können (siehe Abschnitt 3.3.3).

Um dies zu verdeutlichen, wollen wir einige Beispiele anführen. Wir gehen von einem mittelgroßen Unternehmen aus, das Pasta Sauce herstellt und über einen einzigen Standort verfügt. Das Unternehmen möchte eine digitale Verwaltungslösung für seinen Produktionsprozess.

Es ist möglich, die erforderliche Infrastruktur als Vor-Ort-Infrastruktur mit lokalen Servern im Keller der Fabrik und einem geschlossenen Netzwerk zwischen den Clients und dem Server zu schaffen. Eine solche Lösung ist von der Außenwelt abgeschottet und erfordert keine Verbindung zum Internet. Es besteht die Gefahr des Datenverlusts, wenn die Server zerstört werden (z. B. durch einen Brand). Außerdem muss die Hardware von einem Administrator gewartet werden.

Dasselbe Unternehmen kann für die erforderliche Infrastruktur bezahlen und ein Rechenzentrum oder einen Cloud-Dienst für den Betrieb der Server nutzen (Infrastructure as a Service – IaaS). Diese Betriebsart erfordert eine Internetverbindung und ist auf eine funktionierende Internetverbindung angewiesen, damit sie funktioniert. Bei der Nutzung von IaaS kann sich das Unternehmen entweder dafür entscheiden, die Managementlösung genau wie bei einer lokalen (On-Premise) Infrastruktur zu realisieren, oder es kann seine Managementlösung auf einer vom Cloud-Anbieter bereitgestellten Entwicklungsplattform aufbauen (Platform as a Service – PaaS). Dabei nutzt das Unternehmen Software-Frameworks und APIs, um gemeinsame Funktionen zu realisieren und zu nutzen. Als weitere Option könnte sich das Unternehmen entscheiden, eine bestehende Managementlösung als Dienstleistung zu nutzen, die von einem anderen Unternehmen angeboten wird (Software as a Service – SaaS), unabhängig davon, ob diese Lösung vor Ort oder in der Cloud realisiert wird.

Praxistipp

Ein DDP sollte sich der grundlegenden Vor- und Nachteile der Nutzung von On-Premise-Infrastruktur, IaaS, PaaS oder SaaS für die Gestaltung, die Konstruktion und die Realisierung der digitalen Lösung sowie für sein Geschäftsmodell bewusst sein. Wenn beispielsweise die Internetverbindung vorübergehend unterbrochen wird, kann es sein, dass das Werk stillsteht. Es wäre auch möglich, das System so zu gestalten, dass es mit eingeschränkter Funktionalität auch ohne Serververbindung betrieben werden kann. Ein alternativer Ansatz könnte darin bestehen, eine Backup-Internetverbindung (z. B. über Mobilfunk) zu nutzen, um die Lösung lauffähig zu halten. Eine lokale Installation kann im Vergleich zu Remote Servern robuster gegen Internetausfälle sein. Die Kosten für den lokalen Betrieb der Server und das Risiko eines Datenausfalls können jedoch höher sein als die Kosten für einen möglichen Internetausfall. Es kann sogar sein, dass die Entwicklung einer Art *Offline-Modus* für die Remote-Lösung teurer ist als die tatsächlichen Kosten für Internetausfälle.

3.3.3 Technologien für die Digitale Kommunikation

Ein zentrales Merkmal des digitalen Zeitalters ist die Konnektivität auf allen Ebenen zwischen Benutzern und Geräten. Die meisten digitalen Geschäftsmodelle beruhen auf der Fähigkeit, Dienstleistungen auf der ganzen Welt anzubieten (siehe Abschnitt 4.2). Das Rückgrat dieser Fähigkeit ist die zugrundeliegende Technologie, die diese digitale Kommunikation ermöglicht.

Die Kommunikationstechnologie besteht aus Kommunikationshardware, wie Kabel, Antennen, Funkgeräten, Empfängern usw., die durch einen Stapel von Protokollschichten, die mit Computer-Hardware realisiert werden, und Kommunikationssoftware betrieben wird. Gemeinsam bieten sie Kommunikationsdienste auf verschiedenen Ebenen an, zum Beispiel:

- Basisdienste, wie Ethernet, WLAN, Bluetooth und Mobilfunk, einschließlich 5G, Radiofrequenz-Identifikation (RFID), Nahfeldkommunikation (NFC) und Infrarot (z. B. zur Gesichtserkennung)
- Netzwerkdienste, wie z. B. das Internet oder das Netzwerk, das Telefone verbindet, wenn eine Nummer gewählt wird
- Anwendungsdienste, wie WWW oder E-Mail

Ein DDP muss die Grundlagen der Form, Funktion und Qualität der zugrundeliegenden Computer- und Kommunikationstechnologie kennen, z. B.:

- Form: Welche Technologien sind auf welchen Geräten verfügbar?
- Funktion: Welche Dienste können durch diese Technologien bereitgestellt werden?
- Qualität: Wie ist die Qualität dieser Dienste in Bezug auf Geschwindigkeit, Speicherkapazität, Kommunikationsbandbreite, Verfügbarkeit, Zuverlässigkeit usw.?

Aus Sicht des Digital Designs ist Kommunikationstechnologie eine „gebrauchsfertige“ Technologie. In Bezug auf Form, Funktion und Qualität hat die Auswahl der richtigen Kommunikationstechnologie jedoch einen großen Einfluss auf eine digitale Lösung.

Kommunikationstechnologie in Bezug auf Form und Funktion

In Bezug auf die Form ermöglicht die Kommunikationstechnologie die Form einer digitalen Lösung, die aus mehr als einem Element besteht. Wie bereits erwähnt, besteht fast jede digitale Lösung aus mehr als einem Element. Verschiedene Kommunikationstechnologien ermöglichen unterschiedliche Ausprägungen der Form. Kommunikationstechnologien mit geringer Reichweite,

wie Bluetooth oder WLAN ermöglichen lokale Netze und erlauben es auch, Elemente innerhalb einer kurzen Reichweite ohne zusätzliche Kosten für den Benutzer zu verbinden. Ein gutes Beispiel ist die Bluetooth-Verbindung zwischen der Smartwatch des Läufers und der Läufer-App.

Wenn es um die Kommunikation über große Entfernungen über das Internet geht, verursacht eine digitale Lösung zusätzliche Kosten. Der Benutzer der Lösung muss über eine Internetverbindung verfügen. Darüber hinaus muss der Anbieter der digitalen Lösung auch über einen Internetanschluss verfügen. Obwohl die Internetverbindung heute ein Massenprodukt ist, sollten die zusätzlichen Kosten im Geschäftsmodell berücksichtigt werden, insbesondere wenn die digitale Lösung zu einem hohen Datenvolumen führen kann. Ein gutes Beispiel ist eine Videostreaming-Lösung. Die Kosten für die Datenverbindung für den Auftraggeber und den Anbieter können erheblich sein, da das Videostreaming eine datenintensive Funktion ist. Von der Funktion her ist die Kommunikationstechnik für den Benutzer normalerweise unsichtbar, da sie Daten zwischen den Elementen transportiert. Wenn es jedoch zu einem Verbindungsausfall oder einer schwachen Verbindung kommt, kann sich eine gut konzipierte digitale Lösung an diese Situation anpassen. Die konkrete Methode zur Bewältigung von Kommunikationsproblemen hängt von der Art der Lösung ab. Im Allgemeinen ist es möglich, Funktionen so zu definieren, dass sie mit unterbrochenen Verbindungen umgehen können und ihre Arbeit wieder aufnehmen, wenn die Verbindung wieder verfügbar ist. Ein kritischer Punkt in Bezug auf die Kommunikation in der YPRC-Fallstudie ist das Remote-Coaching, das auf eine ständige Internetverbindung angewiesen ist.

Kommunikationstechnologie in Bezug auf Qualität

Die Qualität der Kommunikationstechnologie hängt von zwei wichtigen Faktoren ab: Bandbreite und Reaktionszeit. Die Bandbreite gibt die Datenmenge an, die in einer bestimmten Zeit zwischen zwei Elementen transportiert werden kann. Die Bandbreite wird besonders wichtig, wenn größere Datenmengen von einem Element zum anderen übertragen werden müssen. Eine geringe Bandbreite verringert die Geschwindigkeit einer Funktion in einer solchen Situation erheblich. Auch dies wird bei einer gut konzipierten digitalen Lösung berücksichtigt. Die Reaktionszeit der Kommunikationstechnologie wird wichtig, wenn die Funktionen auf verschiedene Elemente verteilt sind. Auch hier ist die YPRC-Fallstudie ein gutes Beispiel. Das Remote-Coaching setzt voraus, dass die Gesundheitsdaten des Läufers nahezu in Echtzeit an den Trainer des Läufers übertragen werden, damit dieser sich in Echtzeit ein Bild vom Gesundheitszustand des Läufers machen kann. Würde dieser Datentransfer eine Minute dauern, wäre die gesamte Trainingserfahrung für den Trainer und den Läufer schlecht, denn eine Minute ist beim Laufen eine lange Zeit.

3.4 Technologie-orientierte Wissensgebiete

Wir sind uns bewusst, dass der DDP nicht in allen Bereichen der System- und Softwareentwicklung ein Experte sein kann. Im folgenden Abschnitt stellen wir jedoch drei Wissensbereiche vor, die wir für einen DDP auf Foundation Level Niveau als sehr wichtig erachten.

3.4.1 Software-Architektur

Software-Architektur befasst sich mit der Definition der grundlegenden Organisation (d. h. der zugrundeliegenden Form und Funktion) eines Softwaresystems und ist ein wichtiger Aspekt bei der Konstruktion und Realisierung einer digitalen Lösung.

Die Architektur eines Softwaresystems kann als eine Metapher verstanden werden, analog zur Architektur eines Gebäudes [PeWo1992]. Wie bei der Architektur eines Gebäudes geht es bei der Definition einer Software-Architektur darum, *grundlegende* Entscheidungen hinsichtlich der strukturellen Auswahl zu treffen. Es gibt viele Architekturstile und -muster, die für den Aufbau von Systemen verwendet werden können. Es ist nicht das Ziel, diese alle zu behandeln, sondern vielmehr einen Überblick über ausgewählte zu geben und dem Leser verständlich zu machen, welche Aspekte der Softwarearchitektur auch für einen DDP relevant sein können.

Zu diesen Architekturstilen gehören *monolithische* Systeme, die eine einschichtige Softwareanwendung darstellen, die User Interface und Datenzugriffscodes in einer einzigen Ebene vereint. Heutzutage sind jedoch Anwendungen mit mehreren Schichten häufiger anzutreffen. Solche *mehrschichtigen Architekturen* können beispielsweise aus der Präsentationsschicht (d. h. des grafischen User Interface), der Anwendungsschicht oder auch logischen Ebene genannt, die die eigentliche Funktionalität der Anwendung bereitstellt, und der Datenschicht (z. B. der zugrundeliegenden Datenbank) bestehen. Diese Trennung ermöglicht auch eine Konzentration auf eine bestimmte Abstraktionsebene im Bauprozess. Eine *ereignisgesteuerte Architektur* ist der vorherrschende Stil, wenn es um grafische User Interfaces geht. Hier finden wir in der Regel Ereignisauslöser (Event Emitter) und Ereignisempfänger (Event Consumer). Ereignisempfänger werden benachrichtigt, wenn Ereignisse von Interesse eintreten und sie reagieren müssen. In einem grafischen User Interface kann beispielsweise das Drücken einer Schaltfläche (z. B. Bestellung senden) ein Ereignis auslösen und dazu führen, dass Ereignisempfänger (z. B. das Lagersystem) darauf reagieren (z. B. indem sie die Bestellung versenden und die Bestandsliste aktualisieren). Nicht zuletzt gibt es *serviceorientierte Architekturen* (SOA), wie z. B. Microservices, die auf lose gekoppelten Diensten basieren, um eine Anwendung [JPMLT2018] zu gestalten.

Die Wahl einer geeigneten Softwarearchitektur ist von großer Bedeutung für das Erreichen der wahrnehmbaren und zugrundeliegenden Qualität (siehe Abschnitt 2.1.3). Das bedeutet, dass die Softwarearchitektur bis zu einem gewissen Grad wahrnehmbare Qualitätsattribute wie Benutzbarkeit, Sicherheit, Zuverlässigkeit und Verfügbarkeit sowie die zugrundeliegenden Qualitäten wie Wartbarkeit und Erweiterbarkeit definiert.

Ist die Entscheidung für eine geeignete Software-Architektur einmal gefallen und das System implementiert, sind Änderungen an der zugrundeliegenden Architektur kostspielig. Die Definition einer zugrundeliegenden Architektur ermöglicht jedoch eine detaillierte Analyse des Verhaltens des Softwaresystems, bevor das System tatsächlich gebaut wird [PeWo1992]. Diese Analyse ermöglicht es auch zu verstehen, ob das zukünftige Softwaresystem die gewünschten wahrnehmbaren und zugrundeliegenden Qualitäten und damit die Anforderungen der

verschiedenen Stakeholder erfüllen wird. Die Möglichkeit, die Folgen einer Architekturentscheidung in einem frühen Stadium zu verstehen, kann dazu beitragen, Risiken zu verringern und Kosten zu sparen [OKKP2015].

Dieses Verständnis ist besonders wichtig, um eine effiziente Zusammenarbeit zwischen Design und Konstruktion während des Bauprozesses zu fördern.

3.4.2 Berechnungskomplexität

Die Komplexitätstheorie befasst sich mit der Frage, welche Menge an Rechenressourcen zur Lösung eines gegebenen Problems benötigt wird [Wegn2005]. Beim Gestalten einer digitalen Lösung muss man sich ihrer Komplexität bewusst sein. Ein DDP kann sich von Fachleuten beraten lassen, wenn sich eine digitale Lösung mit komplexen Problemen befasst, aber idealerweise hat der DDP bereits einen grundlegenden Überblick über relevante Themen der Komplexitätstheorie. Im Folgenden geben wir einen allgemeinen Überblick über dieses Thema. Die Ausführungen orientieren sich stark an [Wegn2005].

Das Hauptaugenmerk der Komplexitätstheorie liegt auf der Lösung von Berechnungsproblemen. Diese Rechenprobleme können von einem echten Computer gelöst werden, aber es gibt auch andere Rechenmodelle wie Turing-Maschinen oder zelluläre Automaten, die die gleiche Rechenleistung haben.

Um gelöst werden zu können, muss ein Rechenproblem die maschinelle Anwendung von mathematischen Schritten ermöglichen. Diese endliche Folge von computerlesbaren Anweisungen wird als Algorithmus bezeichnet. Algorithmen sind ein integraler Bestandteil von digitalen Systemen. Sie sind abstrakte Beschreibungen zur Lösung von wohldefinierten funktionalen Problemen. Alle Algorithmen haben inhärente Zeit- und Speichergrenzen, die bei der Entscheidung, welcher Algorithmus für die Lösung eines bestimmten Problems verwendet werden soll, berücksichtigt werden müssen. Dies unterstreicht auch einen weiteren wichtigen Aspekt der Komplexitätstheorie: die Klassifizierung von Berechnungsproblemen. Dies geschieht durch die Untersuchung des Zeit- oder Platzbedarfs (d. h. des Speichers) für die Lösung eines Berechnungsproblems.

Die *Groß-O-Notation* wird üblicherweise zur Klassifizierung der Komplexität von Algorithmen verwendet und beschreibt die benötigte Ausführungszeit bzw. den Speicherbedarf eines Algorithmus. Die Groß-O-Notation definiert eine Obergrenze für einen Algorithmus, die im Grunde das Worst-Case-Berechnungsszenario beschreibt. $O(1)$ bezieht sich auf einen Algorithmus, der unabhängig von der Größe der Eingabe immer in der gleichen Zeit (oder im gleichen Raum) ausgeführt wird. Dies ist der Idealfall. $O(\log n)$ bedeutet, dass bei Verdoppelung der Eingabe der Zeit- und Platzbedarf nur um $\log(2)$ zunimmt. Ein Beispiel für einen solchen Algorithmus ist der binäre Suchalgorithmus. Ein Algorithmus mit einer Komplexität von $O(n)$ bedeutet ein lineares Wachstum, d. h. die Zeit- und Raumkomplexität wächst proportional zur Eingabe. Wenn Sie zum Beispiel die Eingabe verdoppeln, benötigt der Algorithmus doppelt so viel Zeit für die Berechnung. Ein Beispiel für $O(n)$ wäre die Suche nach einem Element in einer unsortierten Liste. $O(n^2)$ bezieht sich auf einen Algorithmus mit quadratischer Zeitkomplexität, was bedeutet, dass er schlecht skaliert. Erhöht man beispielsweise die Eingabegröße eines solchen Algorithmus um den Faktor 10, steigt die für die Berechnung benötigte Zeit um den Faktor 100. Noch schlimmer ist $O(2^N)$, was sich auf einen Algorithmus mit einer exponentiellen Wachstumsrate bezieht, die mit einer eher kleinen Wachstumszahl beginnt und dann dramatisch ansteigt.

Für mehrere Probleme gibt es unterschiedliche Lösungen (z. B. Sortieralgorithmen). Die Kenntnis der Worst-Case-Szenarien hinsichtlich der Zeit- und Raumkomplexität eines Algorithmus verhindert die versehentliche oder unbeabsichtigte Verwendung eines ineffizienten Algorithmus. Außerdem unterstützt es die Entwickelnden bei der Auswahl von Algorithmen, die den Bedürfnissen der Stakeholder entsprechen und die geforderten wahrnehmbaren Qualitätsmerkmale wie die Leistung erfüllen. Es gibt jedoch auch Probleme, die mit den verfügbaren Computerressourcen nicht angemessen gelöst werden können oder deren Lösung zu kostspielig und zeitaufwendig wäre. Für solche Probleme gibt es manchmal die Möglichkeit, mit Näherungen zu arbeiten und optimale Lösungen für solche Probleme anzunähern, was weniger Aufwand und Kosten erfordert [WiSh2011].

3.4.3 Mensch-Computer-Interaktion

Die Mensch-Computer-Interaktion (HCI) ist ein Wissensgebiet, das sich „mit dem Design, der Evaluierung und der Implementierung interaktiver Computersysteme für den menschlichen Gebrauch und mit der Untersuchung wichtiger Phänomene, die sie umgeben, beschäftigt“ [Hewe1992]. Es umfasst Wissen aus verschiedenen Disziplinen wie Psychologie, Informatik und Design. Das bedeutet, dass es bei HCI darum geht, Menschen und Maschinen zu verbinden, indem menschliche Bedürfnisse (siehe 4.1) berücksichtigt und adäquate Technologien zur Interaktion (siehe 3.2.2) bereitgestellt werden. Dazu gehören auch das Verstehen und Verbessern bestehender Schnittstellen, z. B. durch Beobachtung, wie Menschen mit Computern unter Verwendung einer bestimmten Schnittstelle interagieren. Darüber hinaus ist es das Ziel der HCI-Forschung, neue Technologien und innovative Interaktionsmöglichkeiten (z. B. Augmented Reality) zu entwickeln.

Insgesamt haben sich die Mensch-Computer-Schnittstellen in den letzten zwei Jahrzehnten stark verändert. Dazu gehört das Aufkommen mobiler und tragbarer Geräte und allgegenwärtiger Technologien. Außerdem haben die sozialen Medien die Art und Weise verändert, wie Menschen mit Computern umgehen. Dennoch sind grafische User Interfaces (GUIs) nach wie vor das vorherrschende Mittel der Interaktion [Hewe1992], obwohl es viele weitere Arten von Interaktionstechnologien gibt, wie in Abschnitt 3.2.2 beschrieben.

Aus der Perspektive des Digital Design beeinflussen HCI und das User Interface hauptsächlich die wahrnehmbare Form und Funktion einer digitalen Lösung, die sich mit der unmittelbaren Interaktion zwischen der digitalen Lösung und ihren Benutzern befasst. Für den DDP ist es wichtig, einen Überblick über die vorhandenen Schnittstellentypen zu haben und die spezifischen Eigenschaften dieser Schnittstellen aus der Benutzerperspektive zu verstehen, um die für einen bestimmten Benutzer und Kontext am besten geeignete Schnittstelle für die digitale Lösung auszuwählen und die digitale Lösung für diese ausgewählte Schnittstelle zu optimieren. Mit anderen Worten, es geht darum, die Benutzbarkeit der digitalen Lösung zu verbessern und dabei auch die Eigenschaften der Schnittstelle [Grud1992] zu berücksichtigen. Dazu gehört auch, die persönlichen Bedürfnisse bestimmter Benutzergruppen zu berücksichtigen und digitale Lösungen zugänglich zu machen und sie zu personalisieren. Digitale Lösungen können durch die Berücksichtigung verschiedener Prinzipien des Designs verbessert werden [LWLB2017].

Zu den Mitteln, um eine digitale Lösung aus der Sicht der Benutzer zu bewerten, gehört die Durchführung von Usability-Tests, die eine gängige Methode sind, um eine digitale Lösung an Benutzern zu testen und zu verstehen, wie reale Benutzer das System tatsächlich nutzen [Niel1994]. In der Regel werden Usability-Tests mit einer kleinen Stichprobe von Teilnehmern

durchgeführt und moderiert. Um Usability-Tests in großem Maßstab und zu geringen Kosten durchzuführen, gibt es verschiedene Methoden wie beispielsweise Remote-Usability-Tests ohne Moderator. Darüber hinaus ermöglichen integrierte Feedback-Mechanismen den Benutzern, ihre Bedürfnisse direkt an die Entwickler zu übermitteln. [Orio2018].

3.5 Die Digital Design Perspektive auf Technologie

Zum Abschluss der Vorstellung des digitalen Materials wollen wir noch einmal auf die Sichtweise des Digital Design zurückkommen. Aus der Sicht des Digital Designs sind die folgenden Faktoren wichtig:

1. *Eignung für den vorgesehenen Kontext:* Technologie erfordert immer einen bestimmten technischen Kontext. Einige Technologien sind nur auf bestimmten Betriebssystemen verfügbar. Eine solche Entscheidung kann den potenziellen Markt für eine digitale Lösung einschränken.
2. *Rechtliche Randbedingungen:* Einige Technologien (z. B. Open Source und Web-Services) können sich rechtlich auf die digitalen Lösungen auswirken. Die Nutzung von Web-Services, die in einem anderen Land gehostet werden, wirft beispielsweise fast immer Fragen im Zusammenhang mit dem Datenschutz und den Rechtsvorschriften zum Schutz der Privatsphäre auf, da die Benutzerdaten teilweise in ein anderes Land übertragen werden.
3. *Eignung zur Umsetzung der geplanten Funktionen:* Die ausgewählte Technologie muss in der Lage sein, die vorgesehenen Funktionen zu realisieren. Dies mag einleuchtend klingen, aber manche detaillierten Funktionen (z. B. im Zusammenhang mit künstlicher Intelligenz) können sehr spezielle Software-Frameworks erfordern.
4. *Inspiration für zusätzliche Funktionen:* Bei der Auswahl einer bestimmten Technologie sollte ein DDP immer die Dokumentation einer Technologie studieren oder die Möglichkeiten einer Technologie mit Fachleuten diskutieren. Dies kann zu neuen Erkenntnissen und neuen Ideen hinsichtlich zusätzlicher Funktionen der digitalen Lösung führen.
5. *Wiederverwendung vs. Implementierungsrisiko:* Die Verwendung bereits vorhandener Technologie, insbesondere von Software-Frameworks und Webservices, bietet den Vorteil der Wiederverwendung (siehe oben). Der Preis ist, dass die Technologie so genutzt werden muss, wie sie vom Anbieter konzipiert wurde. Die manuelle Umsetzung kann im Hinblick auf die Flexibilität von Vorteil sein, ist aber mit zusätzlichen Kosten verbunden.
6. *Verfügbarkeit von Fachpersonal:* Die beste Technologie ist nutzlos, wenn kein qualifiziertes Personal zur Verfügung steht, das in der Lage ist, die Technologie anzuwenden. Die Entscheidung für eine bestimmte Technologie muss daher mit der Suche nach kompetentem Personal einhergehen. Alternativ müssen bei der Planung des Bauprozesses die Zeit und die Kosten für die Schulung des vorhandenen Personals in der neuen Technologie sowie ein langsames Entwicklungstempo aufgrund des neu geschulten Personals berücksichtigt werden.
7. *Lizenzkosten:* Technologieanbieter verlangen oft Lizenzgebühren für die Nutzung. Diese Lizenzkosten müssen im Geschäftsmodell berücksichtigt werden.

Die Faktoren 1-5 betreffen die technische Machbarkeit einer digitalen Lösung. Aus Sicht des Digital Design sind sie für die Auswahl einer passenden Technologie für die vorgesehenen Funktionen wichtig und können auch eine Quelle für Innovationen sein. Die Faktoren 6-7 befassen sich mit dem Geschäftsmodell und dem gesamten Bauprozess für eine digitale Lösung. Die Kosten für die manuelle Implementierung, für qualifiziertes (oder ungelertes) Personal und für die Lizenzierung der Technologie müssen im Geschäftsmodell und in der Gesamtplanung berücksichtigt werden.

Schließlich zeigt die Liste der Faktoren, dass technische Entscheidungen eng mit Design-Entscheidungen in Bezug auf eine digitale Lösung verwoben sind. Aus der Sichtweise des Digital Designs ist es sehr wichtig, diese Abhängigkeiten zu verstehen. Dieses Verständnis ist nicht so wichtig, wenn es darum geht, technische Entscheidungen für eine digitale Lösung zu treffen; es ist vielmehr wichtig, um mit technischen Fachleuten zu kommunizieren und einen guten Zeitpunkt zu erkennen, an dem zusätzliches technisches Fachwissen für Designentscheidungen benötigt wird.

4 Querschnittskompetenzen

Neben der Design-Kompetenz (siehe Kapitel 2) und der Kompetenz im Umgang mit digitalem Material (siehe Kapitel 3) erfordert Digital Design auch ein Verständnis mehrerer querschnittlicher Kompetenzen. In diesem Kapitel stellen wir drei Querschnittskompetenzen vor, die wir für einen DDP auf Foundation Level Niveau für wichtig halten:

- Human Factors (Abschnitt 4.1)
- Digitale Geschäftsmodelle (Abschnitt 4.2)
- People Management (Abschnitt 4.3)

4.1 Human Factors

Menschliche Benutzer sind ein zentraler Aspekt fast jeder digitalen Lösung. Die Forschung in der Psychologie und verwandten Bereichen zeigt, dass menschliches Verhalten und Erleben komplex sind. Darüber hinaus haben die Fähigkeiten des Menschen gewisse Grenzen, z. B. in Bezug auf Empfindung, Wahrnehmung und Aufmerksamkeit oder Auswahl und Ausführung von Handlungen. Die Aufmerksamkeit der Benutzer, ihre Handlungen und ihre emotionalen Reaktionen spielen jedoch eine Schlüsselrolle bei der Interaktion zwischen ihnen und einer digitalen Lösung. Ein DDP sollte sich dieser Tatsachen bewusst sein, um Fachleute zu konsultieren und Prototypen zur Bewertung und Verbesserung digitaler Lösungen im Hinblick auf Human Factors einzusetzen.

4.1.1 Grundlagen der menschlichen Wahrnehmung und Aufmerksamkeit

Die menschlichen Sinne verbinden den Menschen mit der Außenwelt. Auf diese Weise können Menschen interpretieren, was um sie herum geschieht, und entsprechend reagieren. Sinnesrezeptoren empfangen Informationen aus der Umwelt, unter anderem die von digitalen Lösungen, in Form von Reizen. So empfangen beispielsweise die Stäbchen und Zäpfchen im menschlichen Auge Lichtwellen. Die Rezeptoren reagieren auf diese Reize, indem sie diese in Nervenimpulse umwandeln, die von den Sinnesnerven an das Gehirn weitergeleitet werden. Diese Kette des Empfangs und der Weiterleitung von Informationen wird als Empfindung bezeichnet. Im Gehirn werden die empfangenen Reize z. B. als Bild eines Baumes interpretiert. Dieser Schritt wird als Wahrnehmung bezeichnet. Die Wahrnehmung folgt auf die Empfindung, aber Empfindung und Wahrnehmung sind miteinander verwoben: Ohne Empfindung wäre die Wahrnehmung nicht möglich, da das Gehirn keine Informationen erhält, die es interpretieren kann. Und ohne Wahrnehmung wäre die Empfindung nur eine Ansammlung bedeutungsloser Informationen in Bezug auf physikalische Reize.

Praxistipp

Der DDP sollte sich darüber im Klaren sein, dass das Vorhandensein eines von der digitalen Lösung erzeugten Reize, wie z. B. eines Symbols auf dem Display einer Smartwatch oder eines akustischen Alarms auf dem Smartphone, nicht unbedingt bedeutet, dass die Benutzer diesen Reiz tatsächlich empfinden und wahrnehmen.

Ein Grund dafür ist, dass die menschlichen Sinne begrenzt sind. Die menschlichen Sinne haben physiologische Grenzen, die für alle Menschen gelten (*Sensorische Fähigkeiten* in Abbildung 50). Ohne Unterstützung durch Hilfsmittel kann der Mensch zum Beispiel kein Infrarotlicht sehen. Außerdem gibt es inter-individuelle Unterschiede, wie z. B. angeborene Farbenblindheit oder

angeborene Schwerhörigkeit. Darüber hinaus können kurz- und langfristige intra-individuelle Veränderungen (d. h. Abweichungen zwischen zwei oder mehreren Eigenschaften, Verhaltensweisen oder Merkmalen einer einzelnen Person) auftreten - so nimmt beispielsweise die Funktionsfähigkeit der Sinne mit zunehmendem Alter ab, oft schrittweise. Obwohl der DDP die Funktionalität der Sinne nicht beeinflussen kann, sollte er sich der sensorischen Einschränkungen der (verschiedenen) Benutzer bewusst sein und diese berücksichtigen (vgl. Barrierefreiheit). Der DDP kann beispielsweise einen Farbenblindheits-Simulator (z. B. <https://colororacle.org/>) verwenden, um zu simulieren, wie farbenblinde Nutzer die von einer digitalen Lösung erzeugten visuellen Reize sehen, um die digitale Lösung gegebenenfalls zu verbessern.

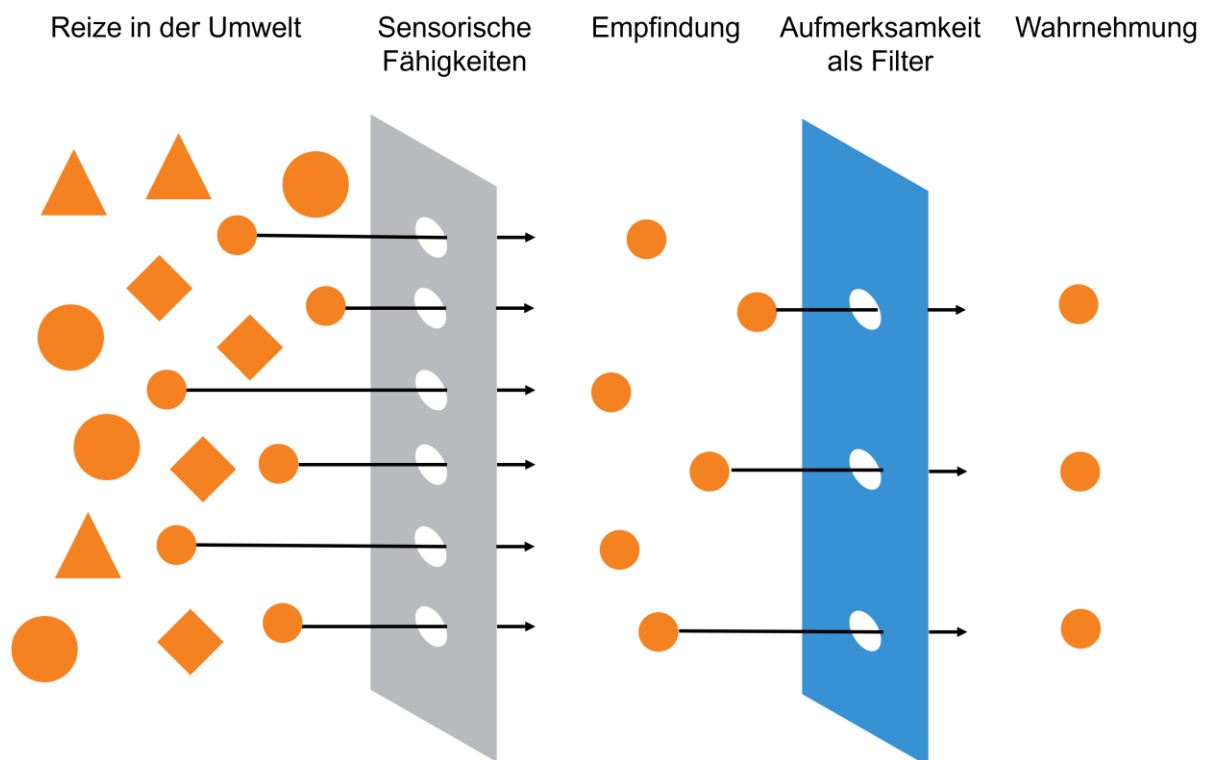


Abbildung 50 - Vereinfachtes Modell von Empfindung, Aufmerksamkeit und Wahrnehmung

Zudem wird nur ein kleiner Teil der durch die menschlichen Sinne erfassten Reize auch bewusst wahrgenommen. Aufmerksamkeit steuert, wie viele und welche Reize die menschliche Wahrnehmung erreichen. Daher wird die Aufmerksamkeit auch als *Filter* [WHBP2016] bezeichnet (*Aufmerksamkeit als Filter* in Abbildung 50), wie im nächsten Abschnitt erläutert wird.

4.1.1.1 Visuelle Aufmerksamkeit

Die meisten Informationen, die das menschliche Gehirn erhält, stammen von den Augen. Um zu sehen, muss der Mensch seine Augen bewegen und fokussieren, denn der Mensch kann nur sehen, wenn das in das Auge eintretende Licht auf der Netzhaut fokussiert wird. Dies wird als *visuell-selektive Aufmerksamkeit* bezeichnet. Laut [WHBP2016] beeinflussen mehrere Faktoren, wohin die Augen blicken und somit, welche Details einer Szene vom Menschen wahrgenommen werden. Zu diesen Einflussfaktoren gehören das Ausmaß, in dem sich ein Reiz vom Hintergrund abhebt (d. h. die Salienz), oder die Wahrscheinlichkeit, wo relevante Reize auftreten könnten. Obwohl die visuell-selektive Aufmerksamkeit für das Sehen erforderlich ist, kann sie negative Folgen haben. Bei der *Veränderungsblindheit* zum Beispiel werden Veränderungen in der Umwelt nicht wahrgenommen [WHBP2016]. Dies wird in mehreren unterhaltsamen Videoclips

veranschaulicht: Während ein Passant in ein Gespräch mit einem Interviewer verwickelt ist, wird ein großes Möbelstück über die Straße getragen. In diesem Moment und unbemerkt vom Passanten wird der vorherige Gesprächspartner durch eine andere Person ersetzt. Die meisten der Befragten bemerken nicht, dass eine andere Person das Gespräch mit Ihnen fortsetzt.

Praxistipp

Beispiele für die Veränderungsblindheit in einer digitalen Lösung sind übersehene Systemrückmeldungen: Das blinkende Symbol für einen niedrigen Batteriestand in der Menüleiste, während sich der Benutzer ein eingebettetes Produktvideo auf einer Website ansieht, oder die Fehlermeldung, die neben dem nicht angekreuzten Kästchen am oberen Rand des Buchungsformulars erscheint, während der Benutzer wiederholt versucht, am Ende des Buchungsformulars auf die Schaltfläche „Absenden“ zu klicken. Die Veränderungen im User Interface wie die Benachrichtigung oder die Fehlermeldung werden nicht gesehen, weil es z. B. zu viele Veränderungen gleichzeitig gibt (z. B. Video) oder das geänderte Element (z. B. die Fehlermeldung neben dem Kontrollkästchen) zu weit vom aktuellen Fokus des Benutzers entfernt ist (z. B. die Schaltfläche „Absenden“ am Ende des Formulars). Laut [Budi2018] gibt es mehrere Designaspekte, die dazu beitragen, Veränderungsblindheit zu verhindern. So könnte beispielsweise die Website mit dem Video abgedunkelt werden, um die Aufmerksamkeit auf das Symbol für den geänderten Batteriestand zu lenken, oder es könnte eine zweite Fehlermeldung neben der Schaltfläche „Absenden“ erscheinen, die auf das nicht angekreuzte Kästchen oben im Formular hinweist.

4.1.1.2 Auditive Aufmerksamkeit

Die auditive Wahrnehmung wird oft vernachlässigt, wenn es um Aufmerksamkeit geht. Im Gegensatz zu visuellen Informationen kann der Mensch jedoch akustische Reize aus jeder Richtung und fast jederzeit empfangen. Die meisten auditiven Informationen sind nicht dauerhaft, was bedeutet, dass der Mensch diesen Input nur für einen (sehr) kurzen Moment wahrnehmen kann [WHBP2016]. Um diese Situation zu bewältigen, teilt der Mensch seine Aufmerksamkeit auf und empfängt Reize aus verschiedenen Hörströmen, indem er unbewusst schnell zwischen diesen Strömen umschaltet.

Praxistipp

Der Mensch kann seine auditive Aufmerksamkeit bewusst auf eine Quelle von Hörinformationen oder ein bestimmtes Hörereignis richten [WHBP2016]. Zum Beispiel können Menschen mit zwei gesunden Ohren andere Gespräche herausfiltern und sich auf einen Sprecher konzentrieren; dies wird als *Cocktailparty-Effekt* [WHBP2016] bezeichnet. Obwohl der Mensch glaubt, nur diesem Sprecher zuzuhören, hört sein Unterbewusstsein immer noch die übrigen Hörströme, z. B. die Hintergrundgeräusche in der Restaurantküche und das Gespräch am Nebentisch. Allerdings hören Menschen die Ströme nicht parallel (d. h. zur gleichen Zeit). Es findet ein sehr schnelles und wiederholtes Umschalten zwischen den Strömen statt, und der Mensch bemerkt dieses Umschalten nicht einmal. Wenn das Gehirn in einem dieser Ströme etwas Bedeutsames wahrnimmt (z. B. ein plötzliches lautes Geräusch in der Küche, der eigene Name wird am Nebentisch genannt), wird dieser Hörstrom fokussiert, unabhängig davon, ob der Mensch dies will oder nicht. Wenn das Gehirn jedoch einen Hörreiz in den Hörströmen als nicht bedeutungsvoll einstuft, wird der Mensch diese Information nicht mitbekommen: Er wird den Reiz einfach nicht hören. Dies zeigt, dass das Gehirn auditive Informationen bereits unbewusst vorverarbeitet und gefiltert hat, bevor der Mensch sie wahrnimmt.

4.1.1.3 Lenkung der Aufmerksamkeit

Die Aufmerksamkeit der Benutzer wird Top-Down oder Bottom-Up gesteuert [WHBP2016]. Im Falle der Top-Down-Kontrolle lenken die aktuellen aktiven Ziele und Aufgaben des Menschen seine Aufmerksamkeit (z. B. das Ziel, das aktuelle Lauftraining zu beenden), während sich die Bottom-Up-Kontrolle auf physische Merkmale eines Reizes bezieht, die die Aufmerksamkeit lenken, wie z. B. die Farbe und der Kontrast der Schaltfläche *Training beenden* im Beispiel der YPRC-Lauf-App.

Praxistipp

Der DDP kann versuchen, die Eigenschaften der digitalen Lösung so zu gestalten, dass die digitale Lösung die Aufmerksamkeit der Benutzer in die gewünschte Richtung lenkt; oder anders ausgedrückt, dass die Benutzer den Reiz höchstwahrscheinlich so wahrnehmen, wie vom DDP beabsichtigt (Bottom-up-Steuerung der Aufmerksamkeit). Um die visuelle Aufmerksamkeit zu erregen, könnte der DDP beispielsweise die Auffälligkeit des zu erkennenden Reizes (z. B. die Farbe einer visuellen Warnmeldung) erhöhen, um die Wahrscheinlichkeit einer Veränderungsblindheit zu verringern. Die Auffälligkeit kann z. B. mit einer Saliency-Karte (z. B. <http://www.saliencytoolbox.net/>) berechnet werden. Als Beispiel für die Erregung der auditiven Aufmerksamkeit könnte der DDP einen sehr lauten Reiz oder einen Reiz mit einer einzigartigen Melodie wählen, so dass sich der zu erkennende Reiz, z. B. eine akustische Benachrichtigung, von der Umgebung abhebt und hervorsteicht. [Wein2011] bietet einen kleinen, aber gelungenen Überblick darüber, wie man mit Geräuschen Aufmerksamkeit erregen kann.

Doch selbst bei klugem Design gibt es immer noch Top-Down Prozesse der Aufmerksamkeitssteuerung, die der DDP kaum beeinflussen kann. Zum Beispiel kann der DDP kaum kontrollieren, wo und wann der Benutzer einen Reiz erwartet oder voraussieht.

4.1.2 Grundlagen der menschlichen Leistungsfähigkeit

Hat ein Mensch Informationen wahrgenommen, entscheidet er sich für eine Aktion und führt diese aus. Hierbei können Menschen sehr schnell und genau sein, aber ihnen können auch Fehler unterlaufen. Fehler werden unterteilt in Fehlentscheidung, Flüchtigkeitsfehler und Versäumnisse [WHBP2016] (siehe Tabelle 18). Diese drei Fehlertypen werden im Folgenden anhand von Beispielen aus der YPRC-Fallstudie beschrieben.

Fehlentscheidung

Bei einer Fehlentscheidung ist die beabsichtigte Handlung zur Bewältigung einer Situation falsch, verursacht durch eine Fehldiagnose der Situation (Interpretationsfehler) und/oder durch eine falsche Auswahl einer Handlung (Planungsfehler). Eine Fehlentscheidung ist also das Begehen einer falschen Handlung: Der Mensch macht einfach das Falsche.

Beispiel YPRC. Die Läuferin Maria könnte annehmen, dass die App automatisch aufhört, Distanz und Dauer zu zählen, wenn sie ihre vordefinierte Laufstrecke beendet hat. Sie geht davon aus, dass der Button „Training beenden“ der YPRC-Läufer-App dazu dient, die Messung zu stoppen, wenn die vordefinierte Laufstrecke vom Läufer abgebrochen wird (Interpretationsfehler). Aufgrund dieser Fehldiagnose der App-Funktionalität sowie des Missverständnisses der Bedeutung der Schaltfläche beschließt sie, nichts zu unternehmen. Nachdem sie ihren regulären Lauf im Wald beendet hat, schließt sie die App einfach (Handlungsfehler) und bemerkt nicht, dass die App auf der Heimfahrt noch immer Entfernung und Dauer misst.

Flüchtigkeitsfehler

Bei einem Flüchtigkeitsfehler interpretiert der Mensch die Situation richtig und entscheidet sich für die richtige Handlung, führt aber letztlich eine andere, falsche Handlung aus. Dieser Fehlertyp steht also für das Begehen einer falschen Handlung, die von der beabsichtigten richtigen Handlung abweicht (das Falsche tun, obwohl das Richtige beabsichtigt war).

Beispiel YPRC. Die Läuferin Maria weiß zum Beispiel, dass sie das Tracking beenden muss, indem sie auf die Schaltfläche „Training beenden“ drückt, und beabsichtigt auch, dies nach Beendigung ihres Laufs zu tun. Anstatt jedoch auf „Training beenden“ zu klicken, drückt sie die Home-Taste ihres Smartphones: Ihre „Standardaktion“ zum Beenden einer App hat sich durchgesetzt. Glücklicherweise erkennt Maria sofort, dass sie die falsche Handlung ausgeführt hat. Sie öffnet die App erneut und klickt auf die Schaltfläche „Training beenden“, wie sie es einige Sekunden zuvor beabsichtigt hatte.

Versäumnis

Beim Versäumnis wird überhaupt keine Handlung ausgeführt, obwohl etwas hätte getan werden müssen (*Vergesslichkeit*).

Beispiel YPRC. Die Läuferin Maria weiß, dass sie auf „Training beenden“ klicken soll, wenn sie ihren Lauf beendet hat. Doch dann erhält sie plötzlich eine lustige Nachricht von ihrem Freund. Nachdem sie eine kurze Antwort geschrieben hat, steckt sie ihr Smartphone wieder in ihre Jacke und beginnt mit ihren Dehnübungen. Sie vergisst schlicht, auf die Schaltfläche „Training beenden“ zu klicken.

Versäumnisse treten häufig bei Abläufen mit mehreren Schritten auf und wären die letzte Handlung in der Abfolge.

Tabelle 18 - Die drei Fehlerarten

Fehlerart	Interpretation der Situation	Geplante Aktion	Ausführung der Aktion
Fehlentscheidung (Mistake)	Korrekt		
	Nicht korrekt	Nicht korrekt	Korrekt
Flüchtigkeitsfehler (Slip)	Korrekt	Korrekt	Nicht korrekt
Versäumnis (Lapse)	Korrekt	Korrekt	Nicht (vollständig) ausgeführt

Die digitale Lösung hat die Aufgabe, die Benutzer vor Fehlern zu bewahren. Bei der Gestaltung der digitalen Lösung bewegt sich die DDP jedoch in einem Spannungsfeld. Einerseits sind viele Fehler, die Menschen in einem laufenden System machen, das Ergebnis eines schlechten Designs des Systems oder organisatorischer Probleme und nicht das Ergebnis unverantwortlicher Handlungen dieser Person, siehe auch [WHBP2016]. Andererseits ist es nahezu unmöglich, ein System zu bauen, das völlig immun gegen menschliche Fehler ist [Wein2011]. Darüber hinaus verwenden die Benutzer unterschiedliche Fehlerstrategien

[Wein2011], wie z. B. das systematische Erkunden von Verfahren zur Fehlerbehebung oder das zufällige Ausprobieren verschiedener Aktionen.

Der Mensch kann aber auch mit Hilfe der digitalen Lösung seine Fehler erkennen und korrigieren, z. B. durch eine geeignete Systemrückmeldung. Unmittelbares und deutliches Systemfeedback von der digitalen Lösung sowie leicht verständliche Anweisungen und Abläufe zur Fehlerbehebung sind unerlässlich. Daher sollte der DDP auch das Systemfeedback sowie die Fehler- und Hinweismeldungen durchdacht gestalten. Der DDP sollte sowohl die verschiedenen Fehlerstrategien als auch die oben erwähnten verschiedenen Fehlerarten berücksichtigen. Beim Verfassen von Anleitungen für Fehlerkorrekturen sollte der DDP beispielsweise einige allgemeine Richtlinien befolgen (siehe [WHBP2016]), wie z. B. das Zielpublikum im Auge behalten, die genannten Schritte zur Fehlerbehebung logisch gliedern und Passiv-Konstruktionen für bessere Verständlichkeit vermeiden. Darüber hinaus sollte der DDP berücksichtigen, dass das Systemfeedback, genau wie jede Ausgabe der Digital Design-Lösung, von den Benutzern wahrgenommen werden muss. [John2010] präsentiert und diskutiert positive und negative Beispiele für visuelle Fehlermeldungen und fasst verschiedene Aspekte zusammen, um sicherzustellen, dass visuelle Fehlermeldungen gesehen und verstanden werden.

Es gibt keine Patentlösung: Viele Faktoren beeinflussen das Maß, in dem Benutzer Fehler machen, Fehler erkennen und Fehler korrigieren können. Und es bleibt verwickelt: Sowohl die Erkennung von Fehlern als auch die Korrektur von Fehlern durch die Benutzer sind Tätigkeiten, welche wiederum fehleranfällig sind.

4.1.3 Emotionen in der Mensch-System-Interaktion

Der Mensch nimmt nicht nur Reize wahr und entscheidet sich dann für (einfache) Handlungen, wie z. B. das Erkennen und Anklicken von Schaltflächen, die von der digitalen Lösung bereitgestellt werden; die digitale Lösung kann bei den Benutzern subjektive Reaktionen auslösen, wie z. B. Freude oder Ärger. Diese Reaktionen können sich wiederum darauf auswirken, ob und wie die Benutzer jetzt oder in Zukunft mit der digitalen Lösung interagieren werden.

Experten und Praktiker aus verschiedenen Disziplinen verwenden unterschiedliche, aber auch sich überschneidende Begriffe, um diese Reaktion zu beschreiben, vgl. [LRHV2009].

Es ist auch umstritten, ob diese Reaktionen vor, während oder nach der Benutzung der digitalen Lösung [ISO2018], nur während der Benutzung [Hass2008] oder während und nach der Benutzung [ScKr2010] auftreten. Es besteht jedoch Einigkeit darüber, dass die Interaktion mit einer digitalen Lösung - unabhängig davon, ob es sich um eine erwartete, tatsächliche oder vergangene Interaktion handelt - Verhaltens- und emotionale Reaktionen beim Menschen auslöst, und dass eine positive Reaktion etwas Wünschenswertes ist [LRHV2009] (siehe Kapitel 6).

Es gibt verschiedene Modelle und Frameworks, die Faktoren beschreiben, die eine solche Reaktion verursachen und beeinflussen. Ein solches Modell ist das Komponentenmodell des Nutzungserlebens (*Component model of User Experience, CUE*) [ThMa2007], siehe Abbildung 51. Wenn ein Benutzer mit einer Digital Design-Lösung interagiert, wird diese Interaktion von den Eigenschaften der digitalen Lösung wie Schnittstellendesign und Funktionalität (*Systemeigenschaften*), den Eigenschaften des Benutzers wie Fähigkeiten und Wissen (*Merkmale der Benutzer*) und den Eigenschaften der jeweiligen Aufgabe und des aktuellen Kontexts des Benutzers (*Aufgabe und Kontext*) beeinflusst. All diese Eigenschaften

prägen die Merkmale der Interaktion, die vom Benutzer als Qualitäten der digitalen Lösung wahrgenommen werden. Diese Qualitäten werden in zwei Arten von Systemmerkmalen unterteilt: Instrumentelle Qualitäten wie z. B. die Effektivität und nicht-instrumentelle Qualitäten wie z. B. die Ästhetik. Die emotionale Reaktion des Benutzers auf eine digitale Lösung wird durch die Wahrnehmung dieser beiden Arten von Systemmerkmalen geprägt, und Emotionen vermitteln zwischen beiden Arten von Wahrnehmungen. Schließlich beeinflussen die emotionale Reaktion des Benutzers und das Empfinden der beiden Systemeigenschaften die Auswirkungen der Nutzung, z. B. das Gesamturteil und die Nutzungsabsicht.

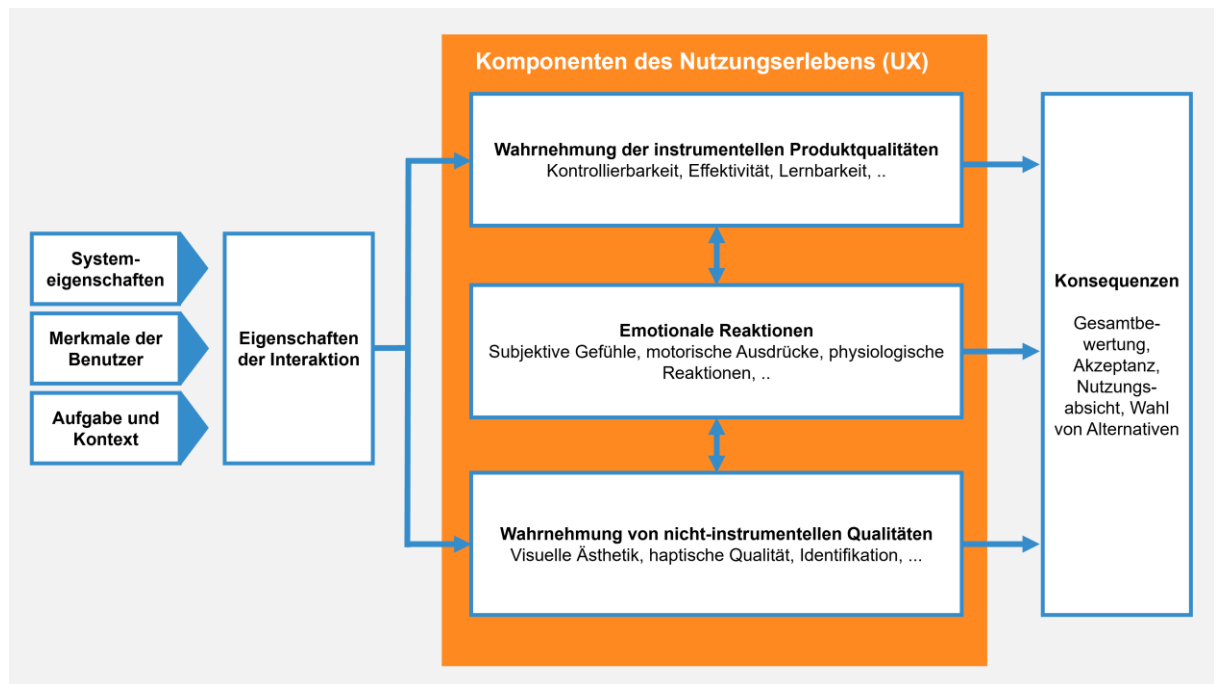


Abbildung 51 - Das Komponentenmodell der User Experience (CUE) (angepasst von [ThMa2007][Ming2020])

Die Auswirkungen der Merkmale einer digitalen Lösung auf die Erfahrung und das Verhalten ihrer Benutzer können gemessen werden. Auf der Grundlage ihres CUE-Modells hat die Forschungsgruppe den wurde der meCUE-Fragebogen [Ming2020] entwickelt. Dieser Fragebogen ermöglicht es dem DDP, zentrale Aspekte des CUE-Modells in Bezug auf die untersuchte digitale Lösung zu bewerten. Der modulare Fragebogen *meCUE* ist in deutscher und englischer Sprache inklusive praktischer Tipps zur Auswertung und Interpretation der Ergebnisse verfügbar [Ming2020].

Es gibt auch andere Modelle und Fragebögen, die untersuchen, wie eine digitale Lösung das Erleben und Verhalten des Benutzers einer digitalen Lösung prägt, z. B. das Modell von Marc Hassenzahl und seinem Team und der dazugehörige AttrakDiff-Fragebogen [HaTr2006] [User2020], oder die Typologie der 20 Stimmungszustände von Xue und Kollegen [XDF2020].

4.1.4 Die Rolle von Prototypen

Um zu vermeiden, dass eine realisierte digitale Lösung Schwachstellen in Bezug auf Human Factors aufweist oder negative Reaktionen hervorruft, sollten frühzeitig Prototypen erstellt werden (siehe 2.3). Sie helfen dem DDP, das Verhalten (z. B. visuelle Aufmerksamkeit, Leistungsfähigkeit, Nutzungsabsicht) und die Erfahrung (z. B. positive Emotionen) des Benutzers zu verstehen und in einer beabsichtigten Weise zu beeinflussen. Siehe Abschnitt 1.3 für eine ausführliche Erörterung des Evaluationskonzepts.

Eine Evaluierung des Prototyps kann helfen herauszufinden, ob und warum Probleme im Zusammenhang mit Human Factors im Prototyp auftreten. Evaluierungen werden in expertenbasierte Ansätze (auch als Inspektionen bezeichnet) und benutzerbasierte Ansätze (auch als empirische Ansätze bezeichnet) unterteilt. Sie können alternativ oder ergänzend zueinander durchgeführt werden. Ein Beispiel für einen expertenbasierten Ansatz ist, dass ein Experte für Human Factors den Prototyp durcharbeitet und nach Problemen sucht. Dabei kann der Experte Richtlinien wie die Interaktionsprinzipien [ISO2020], repräsentative Aufgabenszenarien und Heuristiken verwenden, die ihn bei der Analyse des Prototyps leiten. Benutzerbasierte Evaluierungen umfassen ein breites Spektrum an Methoden, die die Benutzer einbeziehen. Bei einem Benutzertest mit einem ersten Prototyp beobachtet der DDP beispielsweise die Interaktionsschritte des Benutzers mit dem Prototyp, zeichnet ausgesprochen Gedanken des Benutzers auf und führt nach dem Test ein Interview mit dem Benutzer durch.

Aus den Ergebnissen einer solchen Evaluierung eines ersten Prototyps könnte der DDP so den Schluss ziehen, dass die Aufmerksamkeit des Benutzers auf bestimmte Schritte in der zukünftigen digitalen Lösung noch stärker gezogen werden muss. Um das zu erreichen, erstellt der DDP Varianten von optischen, akustischen und kombinierten Alarmen. Anschließend prüft und bewertet der DDP kontinuierlich, welche Alarmvariante die meiste Aufmerksamkeit auf sich zieht und verbessert dann iterativ diese Variante. Bevor die digitale Lösung realisiert wird, wird in einer abschließenden Evaluierung überprüft, ob der entworfene Alarm die Aufmerksamkeit des Benutzers zum richtigen Zeitpunkt erhält.

Jeder Prototyp kann getestet werden, aber die Kategorie des Prototyps (siehe 2.3) kann - neben vielen anderen Aspekten - beeinflussen, ob und in welchem Ausmaß Probleme im Zusammenhang mit Human Factors überhaupt durch den Prototyp identifiziert werden können. So eignet sich z. B. ein Prototyp einer mobilen Anwendung mit mittlerer visueller Präzision weniger gut, um zu testen, ob Probleme bezüglich der Lenkung der visuellen Aufmerksamkeit vorliegen oder nicht.

Es gibt viele Evaluationsstrategien und -methoden, und die meisten von ihnen können für jeden Prototyp und jedes Testziel angepasst werden. Einige Bücher bieten gute Übersichten über Methoden und Entscheidungskriterien, zum Beispiel [Barn2011], [SSRW2013], und [TuAI2013]. Grob gesagt gibt es für jeden Zweck und jedes Budget die richtige Methode. Die Entscheidung über die richtige Test-Methode sowie die Planung, Durchführung und Analyse eines Tests ist jedoch eine Herausforderung. Je nach Hintergrund und Fachwissen des DDP ist es sehr empfehlenswert, Fachleute zu konsultieren und einzubeziehen. Dennoch liegt es in der Verantwortung des Designs, das Prototyping und die iterative Erprobung von Prototypen zu fördern und zu planen. Dabei sollte der DDP klug entscheiden, wann die Testaktivitäten durchgeführt werden sollen - zum Beispiel, sobald eine ausreichende Reife der Lösung erzielt ist oder bevor von einem Schritt im Bauprozess zum nächsten übergegangen wird (siehe Abschnitt 1.3).

4.2 Geschäftsmodelle für digitale Lösungen

Dieser Abschnitt gibt dem DDP einen kurzen Einblick in die Welt der Geschäftsmodelle. Er beschreibt traditionelle Modelle, neue Modelle und ihre Dynamik, Innovationstechniken und die Positionierung des Modells in einem wettbewerbsintensiven Umfeld.

Die Bandbreite und Geschwindigkeit, mit der innovative Geschäftsmodelle die heutige Unternehmenslandschaft verändern, ist beispiellos. In seiner führenden Rolle ist es für den DDP sehr wichtig, die Auswirkungen dieser außergewöhnlichen Entwicklung zu verstehen und die damit verbundenen Herausforderungen systematisch anzugehen. Ein Zitat von Peter Gorb fasst diese Herausforderung zusammen (Zitat aus [StSc2011], übersetzt ins Deutsche):

„Und was die Designer lernen müssen, und das ist das Wichtigste, ist die Sprache der Geschäftswelt. Nur wenn man diese Sprache lernt, kann man die Argumente für Design wirkungsvoll vorbringen.“

Ein Geschäftsmodell beschreibt, wie eine Organisation plant, Werte zu schaffen [OPBS2014]:

Business model: The rationale of how an organization creates, delivers, and captures value in economic, social, cultural, or other contexts.

Der Prozess der Erstellung und Änderung von Geschäftsmodellen wird auch als Geschäftsmodellinnovation bezeichnet und ist Teil der Unternehmensstrategie [GeSE2017].

Sowohl in der Theorie als auch in der Praxis wird der Begriff Geschäftsmodell für ein breites Spektrum informeller und formeller Beschreibungen verwendet, um Kernaspekte eines Unternehmens darzustellen, darunter Zweck, Geschäftsfähigkeiten und -prozesse, Zielkunden, Angebote, Strategien, Infrastruktur, Organisationsstrukturen, Beschaffung, Handelspraktiken sowie betriebliche Prozesse und Richtlinien einschließlich Kultur.

Wann immer eine digitale Lösung eine wichtige Rolle in der Wertschöpfungskette eines Unternehmens spielt, gilt es wichtige Beziehungen zwischen dem Digital Design und dem Geschäftsmodell eines Unternehmens zu berücksichtigen, einschließlich des Wertversprechens an interne oder externe Kunden:

- Das Wertversprechen des Geschäftsmodells (siehe Abschnitt 2.2.3) muss durch das digitale System als Teil einer digitalen Lösung realisiert werden.
- Die Möglichkeiten und Grenzen des digitalen Materials definieren die Möglichkeiten und Grenzen für die Wertschöpfung.
- Die Kunden des Unternehmens können Benutzer (User) oder Stakeholder der digitalen Lösung sein.
- Die Benutzer des digitalen Systems können ebenfalls Teil des Wertversprechens des Unternehmens sein.
- Die Kosten für Bau und Betrieb eines digitalen Systems innerhalb einer digitalen Lösung machen oft einen wesentlichen Teil der Kostenstruktur eines Unternehmens aus.
- Die Schaffung von Einnahmequellen ist oft Teil des digitalen Systems (z. B. das Sammeln von Zahlungsdaten, die Interaktion mit einem Zahlungsanbieter).

Beim Bau einer digitalen Lösung geht es daher nicht nur um die Form, Funktion und Qualität des digitalen Systems, das die digitale Lösung ermöglicht (siehe Kapitel 1), sondern auch um die Definition eines Geschäftsmodells zusammen mit dem digitalen System (siehe Abschnitt 2.1) und die damit verbundene Wertschöpfung für Unternehmen, Kunden und Gesellschaft.

Bei der digitalen Transformation (siehe Kapitel 1) geht es sogar darum, bestehende Geschäftsmodelle zu ersetzen. Mit dem iPod und dem iTunes Store hat Apple ein innovatives neues Geschäftsmodell geschaffen, welches das Unternehmen zur dominierenden Kraft im Bereich der Musik-Downloads gemacht hat. Skype hat uns mit einem innovativen

Geschäftsmodell, das auf der Peer-to-Peer-Technologie basiert, günstige weltweite Telefentarife und kostenlose Anrufe zwischen Skype-Benutzern ermöglicht. Heute ist Skype die größte internationale Telefongesellschaft der Welt.

Praxistipp

Eine digitale Lösung zu entwickeln bedeutet, ein Wertversprechen zu entwerfen, das direkt mit einem effektiven Geschäftsmodell verbunden ist. Der Vorteil für einen DDP besteht darin, dass die Anwendung der Methoden und Technologien aus diesem Handbuch die Grenzen des Denkens erweitert, um neue Optionen zu generieren und letztendlich mit Hilfe von digitalem Material einen Mehrwert für Benutzer und Unternehmen zu schaffen.

Wann immer es möglich ist, sollte der DDP die Perspektive des Digital Designs in Managementmodelle und Managementdenken integrieren und Fälle für die integrative Gestaltung von digitalen Lösungen und Systemen schaffen. Durch die Gestaltung eines Geschäftsmodells in einem frühen Stadium des Bauprozesses hat der DDP die Möglichkeit, systematisch Wertversprechen zu entwickeln, zu gestalten und umzusetzen, um alte, überholte Angebote in Frage zu stellen und sie gemeinsam mit den Stakeholdern zu testen und in neue und innovative Geschäftsmodelle umzuwandeln.

4.2.1 Geschäftsmodellmuster

Eine gute Einführung in die Welt der Geschäftsmodelle ist die Kategorisierung von Choudary. Er unterscheidet zwischen zwei allgemeinen Mustern von Geschäftsmodellen [Chou2013]:

- *Pipelines (lineare Geschäftsmodelle)*: Unternehmen stellen Waren und Dienstleistungen her, bringen sie auf den Markt und verkaufen sie an Kunden. Wert wird vorgelagert produziert und nachgelagert konsumiert. Es gibt einen linearen Fluss, ähnlich wie Wasser, das durch ein Rohr fließt.
- *Plattformen (vernetzte Geschäftsmodelle)*: Plattformen stellen nicht nur Dinge her und vertreiben sie, sie ermöglichen es den Kunden, Werte zu schaffen und zu konsumieren. Moazed definiert eine Plattform als ein Geschäftsmodell, das Werte schafft, indem es den Austausch zwischen zwei oder mehreren voneinander abhängigen Gruppen, in der Regel Konsumenten und Produzenten eines bestimmten Wertes, erleichtert [MoJo2016]. Es ist das vorherrschende Geschäftsmodell, das die digitale Transformation vorantreibt (siehe Kapitel 1).

Neben den beiden allgemeinen Modellen sollte der DDP auf Foundation Level Niveau weitere detaillierte Typen von Geschäftsmodellmustern kennen [OPBS2014]:

- *Entflechtungs-orientiertes Geschäftsmodell (Unbundling Business Model)*: Geschäftsmodelle, die die drei Bereiche Kundenbeziehungen, Produktinnovation und Bereitstellung und Wartung von Infrastrukturen in unterschiedlichem Umfang kombinieren (Beispiele sind Telekommunikationsanbieter wie die Deutsche Telekom, Swisscom und AT&T).
- *Nischenprodukt-orientiertes Geschäftsmodell (Long Tail Business Model)*: Anstatt eine begrenzte Anzahl von Produkten anzubieten, bedeutet das Long Tail Business Model, dass eine breite Palette verschiedener Produkte angeboten wird, indem eine hervorragende Logistik genutzt wird. Dies ermöglicht es einem Unternehmen, mit ansonsten unrentablen Nischenprodukten Gewinne zu erzielen (Beispiel: Apple iTunes).

- *Multi Sided Platform Geschäftsmodell:* Eine Plattform ermöglicht die Interaktion von zwei oder mehr unabhängigen Gruppen. Der Wert für eine einzelne Gruppe ergibt sich aus dem Vorhandensein einer anderen Gruppe (Beispiel: Google.com, die Gruppen sind Werbetreibende und Suchmaschinennutzer). Je mehr Benutzer die Suchmaschine von Google verwenden, desto mehr Daten stehen Google zur Verfügung, um die Suchergebnisse zu verbessern. Je größer der Marktanteil der Google-Suchmaschine ist, desto mehr Inserenten schalten ihre Anzeigen über Google, was wiederum Googles Verhandlungsposition bei der Preisgestaltung stärkt, und in diesem Geschäftsmodell gibt es mehrere leistungsstarke Regelkreise.
- *Freemium-Geschäftsmodell:* Ein Standarddienst wird kostenlos angeboten; für erweiterte Funktionen ist ein kostenpflichtiges Abonnement erforderlich (Beispiel: Online-Community LinkedIn).
- *Verbundene Produkte Geschäftsmodell:* Ein kostengünstiges oder kostenloses Erstprodukt oder eine Dienstleistung motiviert zur Nutzung künftiger kostenpflichtiger Ersatzprodukte oder Dienstleistungen (Beispiel: Gillette-Rasierer und -Klingen, HP Farbtintenstrahldrucker). Auch bekannt als Köder-und-Haken- oder Rasierklingen-Geschäftsmodelle.
- *Offenes Geschäftsmodell:* Ein kollaboratives Geschäftsmodell, das externe Fachleute einsetzt, um Mehrwert zu schaffen und zu sichern (Beispiel: GlaxoSmithKline).

Für eine Betrachtung von Geschäftsmodellen auf Foundation Level Niveau ist die vorliegende Liste mehr als ausreichend. Leser, die mehr Interesse an Geschäftsmodellen haben, finden weitere Geschäftsmodellmuster in [Gass2013].

4.2.2 Digitale Geschäftsmodelle

Ein Digital Business zielt darauf ab, digitales Material für digitale Lösungen nutzbar zu machen, um neue Geschäftsmodelle zu ermöglichen, die einer Organisation einen Wettbewerbsvorteil verschaffen. Im Gegensatz dazu zielt E-Business primär auf die digitale Umsetzung (siehe 0) eines bestehenden Geschäftsmodells ab, typischerweise mit dem Ziel, Kosten zu sparen oder mehr Kunden zu gewinnen, ohne jedoch das Geschäftsmodell selbst grundlegend zu verändern.

Digital Business wiederum ist ein Enabler für die digitale Transformation, d.h. digitale Lösungen verändern das Verhalten und das Leben der Menschen und haben Auswirkungen auf die Gesellschaft. Soziale Medien, wie Instagram, oder Streaming-Dienste wie Netflix sind Beispiele dafür, wo digitale Lösungen die digitale Transformation vorantreiben.

Digitale Geschäftsmodelle sind daher ein Spezialfall der oben vorgestellten Geschäftsmodellmuster (siehe Abschnitt 4.2). Im Folgenden werden drei Beispiele für digitale Geschäftsmodelle erörtert, um die breite Palette der Möglichkeiten aufzuzeigen:

- Uber setzt auf die weite Verbreitung von Smartphones und nutzt ein Geschäftsmodell, das relativ wenig Kapital erfordert. Ein traditionelles Taxi-Unternehmen benötigt Fahrzeuge und hat die Kosten für die Einstellung und Verwaltung seiner Angestellten. Denkbar wäre zwar ein Geschäftsmodell, bei dem die Kunden ihr Taxi online bestellen und eventuell auch bereits durchgeführte Fahrten einsehen können, das Erlebnis der Taxifahrt selbst bliebe aber dasselbe. Uber hingegen behauptet sich als digitales Unternehmen, weil es im Wesentlichen eine Plattform ist, die Fahrgäste und Fahrer über das Internet miteinander verbindet und gleichzeitig ein besseres Kundenerlebnis schafft.

- Netflix ist ein gutes Beispiel dafür, wie ein Unternehmen sein ursprüngliches E-Business in ein Digital Business umwandeln kann. Ursprünglich nutzte Netflix Technologie, um ein Inventarsystem zu verwalten und DVDs zum Verleih an Kunden zu senden. Obwohl dies für die Kunden bequemer war, glich es letztlich einer digitalen Version des Videoverleihs. Durch das Angebot von Video-on-Demand-Streaming-Diensten hat Netflix sein eigenes Geschäftsmodell durchbrochen und die Art und Weise, wie Menschen Film- und Fernsehmedien nutzen, verändert. Der Konsum von Filmen und Fernsehsendungen zu jeder Zeit und an jedem Ort wäre heute ohne den Zugang zum Internet, der für die Verbraucher nahezu universell ist, nicht möglich.
- Disney ist ein traditionelles Unternehmen, das digitale Technologien einsetzt, um das Erlebnis in den Themenparks zu verbessern. Mit dem MagicBand, einem Armband, das über RFID und Funk mit Sensoren im Park verbunden ist, können die Besucher des Parks nun einen außergewöhnlichen Parkbesuch genießen. Fünf grundlegende Dinge wurden verbessert: Der Besuch einer Attraktion, die Unterbringung im Hotel, das Essengehen, das Fotografieren und Versenden der Bilder an Freunde oder Verwandte sowie der Kauf von Souvenirs. Die Gäste erhalten das MagicBand einige Wochen vor dem Besuch und können damit den Park betreten, Essen oder Lebensmittel kaufen, Attraktionen buchen und Wartezeiten in Echtzeit einsehen. Letztendlich haben die Gäste das Gefühl, dass sie sich fließend zwischen der digitalen und der physischen Welt bewegen.

4.2.3 Nachdenken über zukünftige Digital Business Potenziale

Das Wertversprechen und das Geschäftsmodell einer digitalen Lösung hängen von der Qualität der Dienste ab, die den Benutzern der Lösung angeboten werden. Das Three-Horizon-Model [BaCoWh1999] kann als Werkzeug verwendet werden, um systematisch über die Art und Weise des Digital Business nachzudenken, den Reifegrad einer digitalen Lösung zu ermitteln und den Grad der Unternehmenstransformation zu bestimmen. Es strukturiert mögliche Ideen für Digital Business in drei Horizonten (Abbildung 52 - Das Three-Horizon-Model).

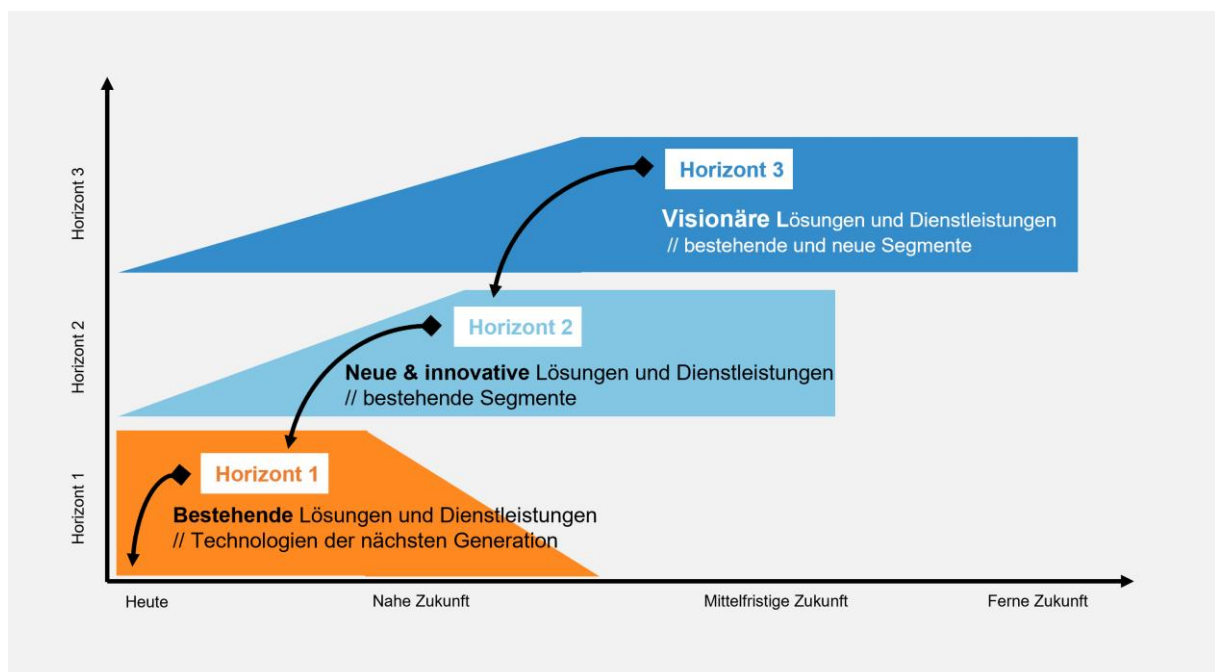


Abbildung 52 - Das Three-Horizon-Model

Horizont 1 befasst sich mit dem vorhandenen Geschäft (Business) einer Organisation. Das bestehende Geschäft wird durch bestehende Lösungen vorangetrieben, die als Teil des Geschäftsmodells einer Organisation wichtige Einnahmen liefern. Im Hinblick auf Digital Design können diese bestehenden Lösungen durch innovative digitale Technologien optimiert werden. Bei solchen Optimierungen handelt es sich in der Regel kurzfristige Projekte, die schnell am Markt realisiert werden können und jeweils nur eine kurze Bauphase erfordern. Diese Optimierungen können direkt geplant werden und resultieren oft aus einer Tame Problem Situation (siehe 2.1). Die Zeitspanne von heute bis in die ferne Zukunft hängt von der Domäne und vielen anderen Bereichen ab. Die Smartphone-Industrie ist beispielsweise ein sehr schnelllebiger Bereich.

Beispiel YPRC. Nehmen wir an, dass die Smartwatch mit dem Pulssensor ein erfolgreiches Produkt auf dem Markt ist und dass der Remote-Coaching-Service ebenfalls erfolgreich ist. Gehen Sie weiter davon aus, dass die technischen Fortschritte im Bereich der Smartwatches es ermöglichen, preiswerte Smartwatches mit eingebauter mobiler Internetverbindung zu bauen. Mit einer solchen Technologie wäre es möglich, eine reine Smartwatch-Lösung anzubieten, die das Smartphone als Verbindung zum Läuferportal nicht mehr benötigt.

In Horizont 2 geht es um innovative Lösungen/Dienstleistungen im Bereich des Unternehmens. Solche Lösungen können das Geschäftsmodell eines Unternehmens in bereits bekannten Bereichen erweitern. Die Schaffung solcher Lösungen resultiert aus einer Tame Problem oder sogar einer Wicked Problem Situation. Es handelt sich um einen mittelfristigen Bauprozess, der eine größere Investition in Konzeptarbeit erfordert, um das Risiko zu minimieren, das mit dem Aufbau einer erfolgreichen Lösung verbunden ist.

Beispiel YPRC. Zusätzlich zum Remote-Laufcoach-Service könnte ein Remote-Coaching-Service für Leichtathletik (z. B. Gewichtheben) oder Radfahren eingerichtet und angeboten werden. Mit einer solchen Erweiterung könnte das Unternehmen den YPRC seinen Kundenstamm auf andere Segmente von Sportlern ausweiten.

In Horizont 3 geht es um zukünftige Möglichkeiten, die auf visionäre Ideen in Bezug auf Lösungen und/oder Technologien abzielen. Solche Ideen können das Geschäftsmodell einer Organisation erheblich erweitern (und herausfordern). Ein Beispiel für einen solchen Horizont 3 ist der Fall Netflix (siehe Abschnitt 4.2.2). Die Annäherung an solche Ideen ist ein schwieriges Problem und ein eher langfristiges Programm, das in der Regel einen erheblichen Aufwand der Scoping- und Konzeptarbeit erfordert. Handelt es sich bei der Idee auch um eine neue Technologie, besteht ein weiteres Risiko, dass die neue Technologie nicht ausgereift genug ist, um produktiv eingesetzt werden zu können.

Beispiel YPRC. Derzeit wird das Remote-Coaching von einem Menschen durchgeführt. Unter der Annahme, dass die Technologie der künstlichen Intelligenz so weit entwickelt ist, dass eine Echtzeitanalyse des Gesundheitszustands des Läufers möglich ist, könnte der menschliche Trainer durch einen virtuellen KI-basierten Trainer ersetzt werden, der den Läufer mit Hilfe von Sprachtechnologie anleitet.

Wenn Sie sich dafür entscheiden, eine digitale Lösung zu schaffen, die als Horizont 2 oder sogar Horizont 3 eingestuft werden kann, ist eine völlig andere Arbeitsweise erforderlich. Die Planungs- und Denkrichtung von Horizont 3 oder 2 wird von der Zukunft in die Gegenwart gerichtet.

Wie in Abschnitt 4.3 hervorgehoben wird, ist in dieser Situation eine breite, abstrakte Denkweise des Umsetzungsteams erforderlich. Mit einer solchen Denkweise ist es möglich, sich mit Hilfe von

Zukunftsszenarien eine inspirierende Zukunft vorzustellen. Solche Szenarien sind notwendig, um die zukünftige Lösung zu gestalten, indem man sich einen zukünftigen Kontext und zukünftige Kundenbedürfnisse in ferner Zukunft vorstellt. Solche Zukunftsszenarien können dann als immer konkretere Storys ausgearbeitet werden, die schrittweise näher an die Gegenwart geführt werden. Wenn ein gemeinsames Verständnis des Umfangs, seiner Risiken und der Lösungsidee stabil ist, folgt die Planungsrichtung dem Planungstyp eines Bauprozesses: Den Schritten Scoping, Konzeption, Entwicklung und Betrieb.

4.3 People Management

Bisher haben wir den Bauprozess sowohl aus methodischer (Kapitel 2 und Abschnitt 4.1) und technischer (Kapitel 3) als auch aus unternehmerischer Sicht (Abschnitt 4.2) betrachtet. In diesem Abschnitt wird eine weitere wichtige Perspektive eingeführt: Die Perspektive auf Personen im Bauprozess digitaler Lösungen. Diese Perspektive ist wichtig, weil Personen beim Bau digitaler Lösungen unterschiedliche Rollen einnehmen:

- Personen können die zukünftigen Benutzer- oder Kunden-Rolle der digitalen Lösung einnehmen
- Personen können Auftraggeber sein, die zusätzliche Anforderungen an die digitale Lösung stellen
- Personen können als Teammitglied den Bauprozess durchführen

Personen als Auftraggeber, Benutzer, Kunde, Teammitglied oder als Stakeholder stehen im Mittelpunkt des Digital Designs und werden in diesem Handbuch differenziert betrachtet. Die in diesem Handbuch vorgestellten Techniken sollen Personen (z. B. Teammitglieder, Auftraggebende, manchmal auch Sponsoren genannt) unterstützen, die den Bauprozess anwenden, insbesondere den Digital Design Prozess führen.

Es gibt jedoch eine weitere Dimension für Personen innerhalb des Bauprozesses. Gutes Digital Design wird durch die interdisziplinäre Zusammenarbeit erreicht (siehe auch Kapitel 6). Auf Foundation Level Niveau muss der DDP Schlüsselindikatoren für das Verständnis von menschlichem Verhalten und Teamdynamik benennen können. Dieses Verständnis ist notwendig, um die zwischenmenschliche Dynamik in interdisziplinären Teams zu charakterisieren und die Notwendigkeit verschiedener Typen im Bauprozess für eine digitale Lösung ansprechen zu können. Wie der Abschnitt 4.1 über Human Factors darauf abzielt, ein Bewusstsein für die menschliche Dimension bei der Nutzung digitaler Lösungen zu schaffen, zielt dieser Abschnitt darauf ab, ein Bewusstsein für die menschliche Dimension während des Bauprozesses zu schaffen.

Ein weit verbreitetes Missverständnis sieht den Bauprozess als einen mechanischen Prozess; stattdessen sollte er als ein sozialer Prozess betrachtet werden, der zwischen den am Bauprozess beteiligten Menschen stattfindet (vgl. z. B. [VPGV2008]). Dazu gehören die Personen, die die digitale Lösung tatsächlich bauen und diejenigen, die an der Gestaltung beteiligt sind (d. h. zukünftige Benutzer und andere Stakeholder). Die soziale Prozessperspektive berücksichtigt die einzelnen Individuen mit ihren Beziehungen und ihrem Verständnis für die digitale Lösung sowie die Kommunikation und Zusammenarbeit zwischen den beteiligten Personen. Darüber hinaus bedeutet die soziale Perspektive, zu verstehen, wie der persönliche Hintergrund und die Profession einer Person die Kommunikation und die Teamdynamik beeinflussen können. Die große Bandbreite dieser Perspektiven unterstreicht die Wichtigkeit, den

Bauprozess als sozialen Prozess zu verstehen. Der Umgang mit diesen Perspektiven geht weit über den Foundation Level hinaus. Ein DDP auf Foundation Level Niveau sollte sich jedoch über die Bedeutung dieser Perspektiven bewusst sein.

4.3.1 Den Bauprozess als sozialen Prozess begreifen

Die folgenden Aspekte sind nützlich, um den Bauprozess als sozialen Prozess zu verstehen:

- Personen und Organisationen bevorzugen unterschiedliche Formen und Mittel zur Kommunikation.
- Personen und Organisationen haben unterschiedliche Ausgangspunkte (z. B. Bildung, persönliche Erfahrung, Herkunft und Lernstil), um einen Sachverhalt wahrzunehmen und zu verstehen.
- Personen und Organisationen bevorzugen je nach ihrer Persönlichkeit unterschiedliche Arbeitsweisen, um eine Aufgabe zu bewältigen.
- Die verschiedenen Phasen des Bauprozesses stellen unterschiedliche Herausforderungen an die Personen und die Organisation in Bezug auf Teamdynamik und Führung.
- Das Arbeitsumfeld (z. B. Zeit, Ort, Erwartungen) hat einen entscheidenden Einfluss auf das Handeln und Verhalten von Personen.

Digitale Lösungen werden immer in einem kulturellen Umfeld entwickelt

Die oben genannten Aspekte lassen sich als das kulturelle Umfeld der am Bauprozess beteiligten Personengruppe zusammenfassen. Kultur ergibt sich, wie auch die Kultur im Bauprozess, aus den Überzeugungen und Verhaltensweisen der Teammitglieder und Entscheider, die darüber entscheidet, wie sie miteinander interagieren. Oft ist Kultur implizit, nicht ausdrücklich definiert und entwickelt sich im Laufe der Zeit organisch aus den kumulativen Merkmalen der beteiligten Menschen.

Für ein Umsetzungsteam kann die Kultur durch Landeskulturen und Traditionen, wirtschaftliche Entwicklungen, Unternehmensgröße und Produkte beeinflusst werden. Kulturen, ob absichtlich geformt oder organisch gewachsen, reichen bis zum Kern der Glaubenssätze und Praxis eines Unternehmens und beeinflussen jeden Aspekt des beruflichen Alltags [TARV2019]. Daher haben die Persönlichkeit der am Bauprozess beteiligten Personen und die Unternehmenskultur der Bauorganisation einen großen Einfluss auf die Effizienz und Effektivität des Bauprozesses.

Der Bauprozess für eine digitale Lösung bietet Orientierung und ein Verständnis für die angestrebte Zukunft (siehe Abschnitt 2.1). Dieses Verständnis sollte nicht als Nebenprodukt des Prozesses betrachtet werden; es ist oft ein zentraler Wert für die Beteiligten. Das Verständnis der zukünftigen Zusammenhänge ist ein wichtiger Faktor, dass Menschen sich für den Wandel engagieren. Erst dann können Personen die angestrebte Zukunft klar kommunizieren und als Botschafter für die Idee auftreten. Darüber hinaus sorgt eine klare Kommunikation über die angestrebte Zukunft dafür, dass alle Beteiligten das gleiche Ziel verfolgen. Dies erhöht die Bereitschaft zur vertrauensvollen Zusammenarbeit als Team. In Kapitel 5 stellen wir Werkzeuge und Methoden vor, um das Verständnis für Inhalte als Teil des Bauprozesses zu verbessern.

Herausforderungen des Bauprozesses aus einer sozialen Perspektive

Die am Bauprozess beteiligten Personen (das Umsetzungsteam) müssen sich mit den folgenden Herausforderungen auseinandersetzen:

- *Aufbau des Teams:* Im Bauprozess kommen Personen aus verschiedenen Disziplinen oder Geschäftspartner zusammen. Manchmal kennen sie sich bereits, manchmal arbeiten sie zum ersten Mal zusammen. Eine neue Beziehung kann zu Verständigungsschwierigkeiten im Hinblick auf die unterschiedlichen Terminologien, Hintergründe und Erfahrungen führen.
- *Gegenseitiges Verständnis im Team:* Übersetzungslücken und Missverständnisse entstehen nicht nur, weil Personen z. B. das Projekt nicht vollumfänglich kennen oder über unzureichende Branchenkenntnisse der Auftraggeber-Organisation verfügen, sondern auch, weil Menschen unterschiedlich sind! Ein Projekt oder ein Unternehmen ist in der Tat ein lebendiges Ökosystem; es besteht aus Interaktionen verschiedener Personen, die jeweils individuell innerhalb ihrer Komfortzone agieren und reagieren. Unter Stress können Personen völlig irrational handeln, obwohl sie genau wissen, wie man sich in der Arbeitssituation verhalten sollte.
- *Fitness für jeden Schritt des Bauprozesses:* Die Teams sind für jeden Schritt des Bauprozesses unterschiedlich geeignet und mental fit. Je nach Projektphase hängt diese Eignung und Fitness von einer wirksamen Mischung der kognitiven Diversity der Teammitglieder, ihrer mentalen Gesundheit, einer klaren, vereinbarten Rollendefinition und Verantwortlichkeiten sowie dem individuellen Führungspotenzial ab.
- *Erwartungsmanagement:* Zeit- und Erwartungsdruck in einem Bauprozess bei gleichzeitig abnehmender inhaltlicher Transparenz, einem unterschiedlichen gegenseitigen Verständnis der Beteiligten und einer zunehmenden Orientierungslosigkeit über die Relevanz digitaler Trends beim Kunden sind nur einige Beispiele, die zu divergierenden Erwartungen führen.
- *Erwartungsentwicklung:* Die Benutzererfahrung mit neuen digitalen Lösungen führt zu neuen Benutzeranforderungen in der Zukunft. Diese Veränderung des Bedarfs findet im digitalen Zeitalter im Vergleich zu früher in immer kürzeren Abständen statt. Das begrenzte Verständnis attraktiver Zukunftsszenarien erfordert eine ständige Interaktion mit dem Kunden im Rahmen des Designprozesses, um herauszufinden, welche Lösung funktionieren wird.
- *Creative Tension Engine* [NIJS2019]: Ab einem bestimmten Punkt nimmt die Aufmerksamkeit in der Teamarbeit ab, die Dinge laufen nicht wie erwartet, Projekte scheitern oder es kommt zu Konflikten. Dies kann verhindert werden, indem eine attraktive Mission erstellt wird (z. B. als Teil der Pressemitteilung aus der Zukunft während des Scoping-Schrittes), um den Sinn des Projekts nach außen zu transportieren und das positive Energieniveau des Teams hochzuhalten. Damit kann Stress vermieden werden, und die Teammitgliedern können sich wieder leichter für das Projekt engagieren.

People Management als systematischer Ansatz

Diese Herausforderungen zeigen, dass die Akzeptanz der Kunden/Benutzer und die Akzeptanz der Menschen in der eigenen Organisation nicht automatisch entsteht. Sie erfordert eine systematische Gestaltung und Steuerung.

People Management bedeutet, drei Aspekte parallel zu bewältigen:

- Management von Prozessstrukturen für die Kommunikation durch Regeln und Verfahren
- Management der Gestaltung der digitalen Lösung innerhalb der Prozessstrukturen
- Management der Erfahrungen der Beteiligten in Bezug auf Erwartungen und Emotionen innerhalb der Prozessstrukturen

Prozessstrukturen (z. B. Definition von Prozessmodellen, Rollen und Vorlagen) sind notwendig, um eine klare Arbeitsweise während des Bauprozesses zu gewährleisten. Sie reichen jedoch nicht aus, um die oben genannten Herausforderungen angemessen zu bewältigen.

Das People Management muss das Verständnis für Ziele und Lösungen, die Wahrnehmung und die Emotionen der Beteiligten integrieren. Das explizite Management von Erwartungen ermöglicht es, Risiken für die Kunden-/Benutzerakzeptanz zu erkennen. Unangenehme Überraschungen auf Auftraggeber Seite können so frühzeitig vermieden werden.

Die Erwartungen der Menschen zu verstehen, ist eine Frage der Kommunikation. Zunächst ist eine gemeinsame Sprache erforderlich, die ein klares und bildhaftes Verständnis der Inhalte vermittelt, so dass Übersetzungslücken zwischen verschiedenen Personen geschlossen, Vorbehalte gegenüber bestimmten Lösungsideen entschärft und individuelle Motive gegenüber der neuen Lösung angesprochen werden können.

Diese gemeinsame Sprache zu finden, ist nicht nur eine Frage der Syntax (Wörter, Konzepte, Modelle, Prototypen) und der Semantik (Bedeutung), sondern vor allem eine Frage der beteiligten Personen (Pragmatik). In Abschnitt 4.3.2 sehen wir, dass Menschen in verschiedenen Dimensionen unterschiedlich sind. Diese verschiedenen Dimensionen haben einen erheblichen Einfluss darauf, wie Personen miteinander kommunizieren und arbeiten. Diese verschiedenen Dimensionen der am Bauprozess beteiligten Personen zu verstehen, ist ein großer Vorteil für das People Management, um eine angemessene gemeinsame Sprache zu finden.

Die Grundlage dafür ist das Verständnis des ganzen Menschen mit seiner Grundhaltung und seinem Verhalten in der Komfortzone wie auch unter Druck. Der DDP kann mit der Komplexität und Instabilität der zwischenmenschlichen Beziehungen im Bauprozess umgehen, wenn er die Wirksamkeit von Persönlichkeitsmodellen und persönlichen Indikatoren kennt, um:

- Die Zuweisung von Mitarbeitern zu Teams und deren Aufgaben zu klären.
- Das Verständnis der Beteiligten über die zukünftige Wertschöpfung einzuschätzen.
- Die Herausforderungen und Vielfalt im Stakeholder Management zu identifizieren und Kommunikationsmaßnahmen zu gestalten.

Hierzu führen wir im folgenden Abschnitt 4.3.2 Persönlichkeitsmodelle und Persönlichkeitsindikatoren ein. In Abschnitt 4.3.3 verwenden wir dieses Modell, um die Teamdimension des Bauprozesses zu diskutieren.

4.3.2 Menschen durch Persönlichkeitsmodelle verstehen

Menschen unterscheiden sich in ihrem Denken, ihren Werten, ihrer Sprache oder Terminologie, ihrem Verhalten oder ihrer Art, mit Stress umzugehen. Jeder Mensch denkt und kommuniziert anders, je nach Tätigkeitsbereich, Persönlichkeit, individuellem Lebensweg oder kulturellem Hintergrund (z. B. Nationalität, Bezugspersonen, Bildung). Auch wenn sich ein Mensch Fachwissen aneignen kann, fällt es ihm auf unterschiedliche Weise leicht oder schwer, etwas zu lernen und anzuwenden, was sich erheblich auf das Stressniveau und das Verhalten auswirkt.

Ein wichtiges Beispiel für den Bauprozess ist folgendes: Menschen sehen und verstehen die Welt aus unterschiedlichen Perspektiven. Einige ziehen es vor, mehr auf Details zu achten. Anderen fällt es leichter, das große Ganze zu sehen. Einige verlassen sich auf Wissen aus der Vergangenheit, indem sie ihr hervorragendes Erinnerungsvermögen nutzen. Dafür haben diese Menschen Schwierigkeiten, eine eigene Vorstellungskraft zu entwickeln. Anderen wiederum fällt es leicht, in die Zukunft zu schauen [Keir1998].

Persönlichkeitsindikatoren sind ein gutes Instrument, um die Verhaltensweisen von Menschen zu verstehen und um ihre Vorlieben zu ermitteln, die wenig Energieaufwand und damit wenig Stress verursachen. Das Wissen über diese Indikatoren wird in den folgenden Abschnitten genutzt, um die Kommunikation zu leiten und die Teamdynamik und Führung im Bauprozess zu verstehen. Persönlichkeitsindikatoren werden in verschiedenen Persönlichkeitsmodellen verwendet.

Persönlichkeitsmodelle sind - wie jedes andere Modell - nützlich, aber begrenzt

Bevor wir jedoch näher auf die Persönlichkeitsmodelle eingehen, müssen wir die Verwendung solcher Modelle im Bauprozess erörtern: Trotz unseres Bedürfnisses nach Vorhersagbarkeit lassen sich menschliche Persönlichkeiten und die komplexe Realität zwischenmenschlicher Beziehungen auch mit dem besten Instrument der Welt nicht erfassen. Alle Modelle bergen die Gefahr, Menschen „in Schubladen zu stecken“. Es macht keinen Unterschied, ob das Modell eine Farbe, eine Buchstabenkombination oder etwas Ähnliches zur Kategorisierung verwendet. Wir sehen den Nutzen dieser Modelle darin, einen ersten Zugang zu verschiedenen Persönlichkeitsmustern zu erhalten, um Unterschiede zwischen Menschen deutlich zu machen. Wirklich gute Teams, Abteilungen oder Unternehmen haben die Diversity der menschlichen Individualität erkannt. Sie nutzen sie für mehr Innovation und Leistung.

Dies kann jedoch nur gelingen, wenn wir uns der Unterschiede zwischen den Menschen bewusst werden. Und Bewusstwerden bedeutet in diesem Zusammenhang nicht, dass wir Teammitglieder anhand verschiedener Zahlen/Buchstaben/Farben in Kategorien einordnen können, sondern vielmehr, dass wir eine erste Vorstellung vom jeweiligen Persönlichkeitsmuster bekommen. Damit erhalten wir eine erste Orientierung, um uns selbst, andere und den eigenen Umgang mit anderen zu verstehen, die der DDP in Gesprächen und gemeinsamen Aktionen überprüfen und vertiefen kann.

Der Keirsey Temperament Sorter ist ein guter Ausgangspunkt

Der Keirsey Temperament Sorter (KTS) ist ein konkretes Persönlichkeitsmodell, das eng mit dem Myers-Briggs Type Indicator (MBTI) verbunden ist. Der MBTI ist ein introspektiver Fragebogen zur Selbsteinschätzung, der unterschiedliche psychologische Präferenzen aufzeigt, wie Menschen die Welt wahrnehmen und Entscheidungen treffen. Die vier Paare von Präferenzen oder „Dichotomien“ sind: Extraversion/Introversion, Sensing/Intuition, Thinking/Feeling, Judging/Perception. Der MBTI basiert auf der Theorie des Schweizer Psychiaters Carl Gustav Jung, der davon ausging, dass Menschen die Welt mit Hilfe von vier kognitiven Funktionen erleben – sensitive oder intuitive Wahrnehmung, fühlende oder denkende Entscheidung – und dass eine dieser vier Funktionen für eine Person die meiste Zeit über dominierend ist.

Keirsey hat das KTS als ein Modell definiert, das aus vier grundlegenden Temperamenten besteht: Artisans, Guardians, Idealists und Rationals [Keir1998]. Gunter Dueck greift in einigen seiner Werke auf Keirseys Bücher zurück und definiert in seinem auf dem KTS basierenden Charaktermodell anschaulichere Temperamentsnamen: Go West, Citizen, Blue Helmet, Star Trek [Duck2013]. Diese Namen stehen für Archetypen, die ein einfaches Verständnis der einzelnen Temperamente ermöglichen. Tabelle 19 fasst die vier Temperamente zusammen und gibt Empfehlungen aus der People-Management-Perspektive.

Tabelle 19 - Charakterisierung der vier Grundtemperamente

Temperament	Motivation	Zeitrahmen	Syntax	Hinweise zum People Management	Archetyp
<p>Artisans... konkret und <u>anpassungsfähig</u>.</p> <p>Auf der Suche nach Anregung und Virtuosität geht es ihnen darum, etwas zu bewirken. Ihre größte Stärke ist die Taktik. Sie sind hervorragend in der Fehlersuche, Beweglichkeit und im Umgang mit Werkzeugen, Instrumenten und Geräten. Sie verhalten sich prozessorientiert.</p> <p>Als konkrete Pragmatiker sprechen Artisans hauptsächlich über das, was sie direkt vor sich sehen, was sie in die Hände bekommen können, und sie werden alles tun, was immer möglich ist, um ihnen einen schnellen, effektiven Gewinn zu bringen, selbst wenn sie die Regeln brechen müssen.</p>	Handlungsfreiraum, Stimulation	Gegenwart	Beschreibend	<ul style="list-style-type: none"> • Persönliche Vorteile, Fakten und Zahlen hervorheben • Freiraum ermöglichen • Toleranz und Freiraum anbieten • Taten, nicht Ergebnisse loben • Vermeiden von Regeln und Routinen <p><u>Typische Rollen:</u> Entwickler, Teamleiter Implementierer, Kommunikator</p>	<p>(SP) Sensitives Wahrnehmen, Perceiving (Wahrnehmen)</p> <p>Die Go West Pioniere in Amerika sind ein Archetyp für dieses Temperament.</p>
<p>Guardians sind <u>konkret</u> und <u>organisiert</u> (geplant).</p> <p>Auf der Suche nach Sicherheit und Zugehörigkeit geht es ihnen um Verantwortung und Pflicht. Ihre größte Stärke ist die Logistik. Sie sind hervorragend im Organisieren, Kontrollieren und Unterstützen. Sie verhalten sich ergebnisorientiert.</p> <p>Als loyale und detailorientierte Personen sprechen Guardians oft über ihre Aufgaben und Verantwortungen, über das, was sie im Auge behalten und worauf sie gut aufpassen können. Sie achten darauf, die Gesetze einzuhalten, die Regeln zu befolgen und die Rechte anderer zu respektieren.</p>	Pflichtgefühl, soziale Stellung, Kontrolle	Vergangenheit	Komparativ	<ul style="list-style-type: none"> • Fakten und Zahlen, persönlichen Nutzen hervorheben • Verantwortung, Funktion und Erfahrung respektieren • Dankbarkeit zeigen • Bestätigen: „Es funktioniert“ • Engagement und ordentliche Arbeit herausstellen • Vermeiden von oberflächlicher Umsetzung <p><u>Typische Rollen:</u> Systemadministrator, Support Service, CFO</p>	<p>(SJ) Sensitives Wahrnehmen, Judging</p> <p>Der Citizen der aufstrebenden Städte ist ein Archetyp für dieses Temperament.</p>

Temperament	Motivation	Zeitraumen	Syntax	Hinweise zum People Management	Archetyp
<p>Idealists sind <u>abstrakt</u> und <u>mitfühlend</u>.</p> <p>Auf der Suche nach Sinn und Bedeutung geht es ihnen um persönliches Wachstum und das Finden ihrer eigenen einzigartigen Identität. Ihre größte Stärke ist die Diplomatie. Sie sind hervorragend darin, zu klären, zu individualisieren, zu integrieren und zu inspirieren. Sie verhalten sich personenorientiert.</p> <p>Idealists beschäftigen sich im Allgemeinen lieber mit Menschen und Gefühlen als mit Logik und Details. Sie reagieren möglicherweise übermäßig emotional (positiv oder negativ) auf die übermittelten Informationen und sind sehr frustriert, wenn sie diese Gefühle nicht zum Ausdruck bringen können. Wenn es notwendig ist, Idealists negative Nachrichten oder Kritik zu übermitteln, ist es wichtig, solche Dinge positiv zu formulieren.</p>	<p>Selbstverwirklichung, Identität, Bedeutung, Authentizität</p>	<p>Zukunft</p>	<p>Metaphorisch</p>	<ul style="list-style-type: none"> • Begeisterung zeigen • Besondere Merkmale hervorheben • Persönlich ansprechen und Beziehungsebene hervorheben • Verständnis zeigen • Lob, persönliche Bestätigung • Vermeiden von Unstimmigkeiten • Vermeiden von Konformität und Werte ignorieren <p><u>Typische Rollen:</u> Scrum Master, Product Owner, Coach, CMO</p>	<p>(NF) iNtuitives Wahrnehmen, Feeling</p> <p>Die Blue Helmets der Vereinten Nationen sind ein Archetyp für dieses Temperament.</p>
<p>Rationals sind <u>abstrakt</u> und <u>objektiv</u>.</p> <p>Auf der Suche nach Meisterschaft und Selbstbeherrschung beschäftigen sie sich mit dem Wissenserwerb und ihrer eigenen Kompetenz. Ihre größte Stärke ist die Strategie. Sie eignen sich hervorragend für jede Art von logischer Untersuchung wie z. B. Engineering, Konzeptualisierung, Theoretisierung und Koordination.</p> <p>Rationals führen gerne lebhaft Diskussionen, legen ihren Standpunkt dar und verwenden Logik, um ihre Argumente zu vermitteln. Sie vermitteln relevante Faktoren auf technische Art und Weise, um ihre Argumente zu verdeutlichen. Sie schätzen prägnante, auf den Punkt gebrachte Kritik und sind genervt von großspurigen Erklärungen für Informationen, die bereits klar und verstanden sind.</p>	<p>Macht über die Natur, Kompetenz, Wissen</p>	<p>Intervall: Vergangenheit, Gegenwart, Zukunft</p>	<p>Konjunktorisch, kategorisch</p>	<ul style="list-style-type: none"> • Wissen und alternative Perspektiven hervorheben • Spielraum bei der Konzeption zulassen • Professionellen Dialog anbieten • Diskutieren statt loben • Vermeiden von fehlerhaften Prinzipien und Inkompetenz <p><u>Typische Rollen:</u> Designer, Architekt, Datenwissenschaftler, CIO, CEO</p>	<p>(NT) iNtuitives Wahrnehmen, Thinking</p> <p>Die Star Trek-Figuren sind dieses Temperament</p>

Die Kommunikationsstile der vier Temperamente lassen sich verstehen, wenn man sie mit den folgenden Ringen eines Baumes vergleicht:

- *Innerer Ring*: Abstrakt versus konkret (Verstehen, Wahrnehmen und Lernen)
- *Der zweite Ring*: Kooperativ versus pragmatisch (Handlungsorientierung)
- *Der dritte Ring*: Direktiv versus informativ (Kommunikationsstil)
- *Der vierte Ring*: Expressiv versus aufmerksam (Energie, Impulsivität)

4.3.2.1 *Der innere Ring: Abstrakt versus konkret (Verstehen, Wahrnehmen und Lernen)*

Menschen denken und sprechen natürlich über das, was sie interessiert. Wenn wir Gesprächen von Personen aufmerksam zuhören, können wir ihre Art, Informationen zu verstehen und wahrzunehmen, nachvollziehen. Die verschiedenen Lernformen machen es ihnen auf unterschiedliche Weise und auf unterschiedlichem Niveau leicht, Dinge zu verstehen [BeBa2007]. Es gibt zwei wesentliche Merkmale, die Unterschiede in der Kommunikation bewirken [Wiki2020c]

- **Abstrakt**: Manche Menschen nehmen Informationen durch symbolische Darstellung oder abstrakte Begriffsbildung auf und sind introspektiv (ihre Wahrnehmung ist abstrakt bzw. intuitiv) Menschen, die im Allgemeinen introspektiv sind, wirken als Träumer „*head in the clouds*“. Sie haben eine eher abstrakte Weltanschauung und konzentrieren sich auf globale oder theoretische Fragen - all die Warum-, Wenn- und Was-sein-könnte-Aspekte des Lebens.

Wichtig für den Bauprozess: Ihnen fällt es leicht, das große Ganze, das Big Picture zu sehen und Vorstellungen für die Zukunft zu entwickeln.

Temperamente, die abstrakt wahrnehmen, sind Idealists (Blue Helmet) oder Rationals (Star Trek).

- **Konkret**: Andere Menschen nehmen Informationen durch direkte Empfindungen über ihre Sinne auf; sie beobachten (ihre Wahrnehmung ist konkret oder sensibel). Menschen, die im Allgemeinen aufmerksam sind, sind eher *bodenständig*. Sie sprechen vor allem über die äußere, konkrete Welt der alltäglichen Realität: Fakten und Zahlen, Arbeit und Spiel, Heim und Familie - all das Wer, Was, Wann, Wo und Wieviel des Lebens.

Wichtig für den Bauprozess: Sie achten mehr auf Details, sind Pragmatiker und greifen auf Erfahrungswerte aus der Vergangenheit zurück.

Temperamente, die konkret wahrnehmen, sind Artisans (Go West) oder Guardians (Citizen).

Jeder verwendet beide Arten der Wahrnehmung, aber im täglichen Leben sprechen konkret orientierte Menschen meist über die Realität, während abstrakte orientierte Menschen häufiger über Ideen und die Zukunft sprechen. Jeder dieser diametral entgegengesetzten Arten stellt eine Richtungsentscheidung dar, die der Einzelne treffen muss, und im Laufe der Zeit entscheidet er sich für einen bestimmten Stil, der ihm leichter fällt. Das Ausmaß, in dem Personen eher beobachtend (konkret) oder introspektiv (abstrakt, intuitiv) sind, wirkt sich direkt auf ihr Verhalten aus.

Praxistipp

Der Bauprozess und insbesondere der Scoping-Schritt erfordert vor allem die Wahl von Personen, denen es leicht fällt, Abstraktes zu denken, und die in der Lage sind, sich mögliche Zukunftsszenarien vorzustellen. Für diesen Schritt werden jedoch auch Personen benötigt, die eine abstrakte Vision in ein konkretes Zukunftsszenario übersetzen können (siehe Abschnitt 4.3.3.1).

4.3.2.2 Der zweite Ring: Kooperativ versus pragmatisch (Aktionsschwerpunkt)

Es gibt zwei grundverschiedene Arten von Maßnahmen, um ein Ziel zu erreichen:

- *Pragmatisch*: Manche handeln in erster Linie nützlich oder sachorientiert, d. h. sie tun das, was zu Ergebnissen führt, was ihre Ziele so effektiv und effizient wie möglich erreicht, und erst danach prüfen sie, ob sie die Regeln einhalten oder den richtigen Weg gehen. Pragmatische Temperamente sind Artisans (Go West) oder Rationals (Star Trek).
- *Kooperativ*: Andere verhalten sich in erster Linie kooperativ oder sozialverträglich, d. h. sie versuchen, das Richtige zu tun, indem sie sich an vereinbarte soziale Regeln, Konventionen und Verhaltenskodizes halten, und kümmern sich erst später um die Wirksamkeit ihres Handelns. Kooperative Temperamente sind Idealists (Blauhelm) oder Guardians (Citizen).

Diese beiden Handlungsweisen können sich natürlich überschneiden, aber in ihrem Leben tun pragmatische Menschen instinktiv und größtenteils das, was funktioniert, während kooperative Menschen tun, sie mit anderen verbindet.

Praxistipp

Prüfen Sie, ob die Situation Rationalität erfordert, wie z. B. die Entwicklung einer Lösung, oder ob persönliche Interessen im Vordergrund stehen, wie z. B. die gemeinsame Zusammenarbeit, die Akzeptanz der Lösung oder die Durchsetzung der Interessen des Auftraggebers. Achten Sie dann darauf, dass Personen über die jeweils notwendige Präferenz verfügen, die die Führung der Situation erfordert.

4.3.2.3 Der dritte Ring: Direktiv versus informativ (Kommunikationsstil)

Der dritte Ring unterscheidet zwischen Menschen, die in der Regel kommunizieren, indem sie entweder nach Informationen suchen oder Menschen, die andere anleiten. Dies wird auch als reaktiv oder proaktiv bezeichnet und beschreibt die Art und Weise, wie Menschen kommunizieren oder handeln.

Praxistipp

Jede Situation erfordert ein anderes Verhalten. Achten Sie auf diejenigen, die unabhängig von ihrer Position und Rolle einen natürlichen Führungsanspruch verfolgen - entweder um in ihrer Rolle bewusster wahrgenommen zu werden oder sich ganz bewusst auf ihre Persönlichkeit zurückziehen.

4.3.2.4 *Der vierte Ring: Ausdrucksstark versus reflektiert (Energie, Impulsivität)*

Der vierte Ring beschreibt, wie Menschen mit ihrer Umwelt interagieren und wie sie Energie erhalten.

- *Expressiv*: Menschen, die in erster Linie sprechen und handeln (aktiv, extrovertiert), bevor sie zuhören und zusehen, werden als expressiv bezeichnet. Einige assoziative Wörter für *expressiv* sind: Impulsiv, aktiv, schwatzhaft, gesprächig, überschwänglich, fließend, ausgiebig, wortreich. Extrovertierte geben dem Leben Weite.
- *Aufmerksam*: Menschen, die in erster Linie zuhören und beobachten (inaktiv, introvertiert), bevor sie sprechen und handeln, werden als aufmerksam bezeichnet. Einige assoziative Wörter für *aufmerksam* sind: Wachsam, mit offenen Augen und Ohren, vorsichtig, umsichtig, achtsam, reflektierend. Introvertierte Menschen geben dem Leben Tiefe.

Extravertierte Menschen schöpfen Energie aus dem Zusammensein mit anderen Menschen, Introvertierte aus dem Alleinsein. Die Dichotomie von expressiv und aufmerksam ist die kontextbezogenste.

Praxistipp

Versuchen Sie, die reflektierte Menschen zum Nachdenken anzuregen, wenn sie sich auf Details konzentrieren oder tiefere Einblicke in ein Thema gewinnen wollen. Versuchen Sie, die reflektierte Menschen dann mehr einzubinden, wenn sie sich auf Details konzentrieren oder tiefere Einblicke in ein Thema gewinnen wollen.

4.3.2.5 *Schlussfolgerung zur Arbeit mit Persönlichkeitsmodellen*

Wenn wir die vier Ringe kombinieren, haben wir 16 Typen, die zur Kategorisierung menschlicher Profile verwendet werden können. Bei diesen Archetypen wird davon ausgegangen, dass jede Person eine bevorzugte Eigenschaft aus jeder Kategorie hat, was 16 einzigartige Typen ergibt. Die vier Ringe stehen in Beziehung zueinander und zu den verschiedenen Temperamenten, die in den 16 zugehörigen MBTI-Typen enthalten sind.

Die wichtigste Lektion für das People Management bei einem DDP auf Foundation Level Niveau ist die Auswahl geeigneter Kommunikationsstile/-formen, um:

- Die Eigenschaften von Menschen und deren individuellen Motive mit ihren angeborenen und erlernten Verhaltensmustern in verschiedenen Situationen zu verstehen (z. B. in Komfortzonen oder unter Stress).
- Sich in die Lage der anderen Person zu versetzen und die Welt mit deren Augen zu betrachten. Beispielsweise eine Persona-Beschreibung zu erstellen oder eine Feldanalyse in der Kundenrolle durchzuführen. Dabei ist es nicht notwendig, das, was der andere denkt oder fühlt, zu billigen, sondern vielmehr ein objektives Interesse daran zu haben.
- Hören Sie Ihrem Gesprächspartner ohne Ablenkung zu und schenken Sie ihm Ihre volle Aufmerksamkeit. Stellen Sie Fragen, um herauszufinden, was in der anderen Person vorgeht, ohne zu urteilen. Fragen Sie nach Motiven und Beweggründen. Es kann hilfreich sein, das Verstandene erneut zu formulieren, um eine Bestätigung zu erhalten.

Praxistipp

Die Anwendung dieses Instruments in der täglichen Arbeit erfordert jedoch ein hohes Maß an Übung und Fortbildung. Zusätzlich zu den persönlichen Indikatoren können psychometrische Indikatoren die Identifizierung von individueller Fitness oder Teamstabilität unterstützen und ein tiefes Verständnis für die Konditionierung und die Eigenschaften Ihrer Kollegen vermitteln. Dieses Verständnis hilft Ihnen, Risiken und gesundheitliche Probleme des Personals frühzeitig zu vermeiden.

4.3.3 Der Bauprozess aus einer gruppendynamischen Perspektive

Im Folgenden verwenden wir das vorgestellte Persönlichkeitsmodell, um den Bauprozess aus einer gruppendynamischen Perspektive zu betrachten. Wir befassen uns mit der zwischenmenschlichen Interaktion, die das Verhalten und die psychologischen Prozesse der am Bauprozess beteiligten Personen prägt.

Um eine wirksame digitale Lösung zu entwickeln, muss ein Umsetzungsteam das Spannungsfeld zwischen den Dimensionen Kunden-/Benutzerbedürfnisse, wirtschaftliche Machbarkeit sowie technische Möglichkeiten während des Entwicklungsprozesses berücksichtigen (siehe Kapitel 1). Die zentralen Merkmale, die für die Bearbeitung dieses Spannungsfeldes erforderlich sind, sind (vgl. [BeBa2007]):

- *Offenheit für Mehrdeutigkeit:* Die Teammitglieder müssen mit Phasen während des Bauprozesses umgehen, in denen Ideen und Richtungen nicht zu 100 % sicher sind. Der chaotische Prozess, der durch den Design-Squiggle (siehe Abschnitt 2.1) beschrieben wird, ist eine gute Analogie für diese Mehrdeutigkeit.
- *Abstrakte Gestaltung:* Die Teammitglieder müssen im Rahmen der Konzeptarbeit (siehe Abschnitt 2.2) abstrakte Konzepte ihrer Ideen entwickeln, um sie auf einen Detaillierungsgrad zu bringen, der im Hinblick auf die drei Dimensionen Kunden-/Benutzerbedarf, wirtschaftliche Machbarkeit und technische Möglichkeiten diskutiert werden kann.
- *Entscheidungssicherheit:* Die Diskussion über Ideen kann in einem Bauprozess nicht ewig weitergehen: Irgendwann muss sich das Team für oder gegen eine Idee entscheiden, um den nächsten Schritt des Bauprozesses und schlussendlich die tatsächliche Realisierung der digitalen Lösung anzugehen.

Um die volle kollektive Intelligenz und Stärke eines Teams während des Bauprozesses zur Entfaltung zu bringen, muss die Organisation People Management im Bauprozess als Erfolgsfaktor begreifen, insbesondere bei selbstorganisierenden Umsetzungsteams. Es muss aktiv die Verantwortung übernehmen, die Gesundheit der Beteiligten (Teamfitness) im Auge zu behalten und darauf zu achten, welche Personen in der jeweiligen Situation das Potenzial des Teams besonders gut mit ihrer Persönlichkeit zur Entfaltung bringen können.

Falsche Personalbesetzungen im Team führen zu einer unzureichenden Zukunftsvision, zur vorzeitigen Auswahl unwirksamer Implementierungsvarianten, zu Zeitverlusten bei der Lösungsfindung und zu individuellem Stress. Persönlichkeitsmuster unterstützen den Entscheider dabei, das Umsetzungsteam hinsichtlich einer wirksamen Team-Diversity (z. B. Kompetenzen, Präferenzen, Disziplinen etc.) zusammenzustellen sowie die Risiken passiver, versteckter Widerstände oder versteckte wirtschaftliche Potenziale frühzeitig zu entdecken.

In den nachfolgenden Abschnitten unterscheiden und erarbeiten wir folgende Aspekte in Bezug auf das People Management im Bauprozess:

- Steuerung der Wahrnehmungen und des Lernpotenzials
- Führung der Zukunftsvorstellungen
- Management von Arbeitsstil und Rollenzuweisung
- Steuerung der Führung während des Bauprozesses
- Berücksichtigung der Arbeitsumgebung und des inhaltlichen Verständnisses

4.3.3.1 Steuerung der Wahrnehmungen und des Lernpotenzials während des Bauprozesses

Das People Management kann die Dimensionen *abstrakt* (Offenheit in der Wahrnehmung und im Lernen) und *konkret* (Hang zur Entschiedenheit oder Abgeschlossenheit in der Wahrnehmung und im Lernen) nutzen, um die Arbeitsweise und die zwischenmenschlichen Beziehungen während des Bauprozesses zu steuern (siehe Tabelle 20 und Abschnitt 4.3.3.3).

Abbildung 53 zeigt den in Abschnitt 2.1.5 vorgestellten idealisierten Bauprozess und veranschaulicht die Verteilung der Temperamente entlang der verschiedenen Schritte des Bauprozesses. Im Scoping-Schritt ist eine abstrakte und offene Denkweise erforderlich, um verschiedene Ideen zu finden und leistungsstarke Lösungsideen zu entwickeln. In der konzeptuellen Phase steigt der Bedarf an konkreten Entscheidungen und Abschlüssen, um einen guten Lösungskandidaten zu identifizieren. Der Bedarf an konkreten Entscheidungen und Abschlüssen nimmt mit der Entwicklung und dem Betrieb der Systeme weiter zu. Doch im Betrieb steigt der Bedarf an abstraktem und offenem Denken wieder an, wenn die bestehende Lösung überdacht oder angepasst werden muss.

In der Forschung über Temperamente und kognitive Stile wurden ähnliche Merkmale (Offenheit für Mehrdeutigkeit und Bedürfnis nach Abgeschlossenheit) als wichtig für die Gestaltung identifiziert. Der generische Bauprozess basiert auf Modellen, die zeigen, wie Menschen lernen und Entscheidungen treffen. Eine Person mit geringer Ambiguitätstoleranz beispielsweise empfindet unklare Situationen als bedrohlich und neigt dazu, Gewissheit zu suchen, indem sie zum Beispiel angesichts neuer Informationen an alten Erkenntnissen festhält, da diese sicherer sind. Dies gilt auch für das Bedürfnis nach einer Entscheidung, das sowohl ein Persönlichkeitsmerkmal als auch situationsbedingt sein kann. Das Bedürfnis nach einer Entscheidung bedeutet, dass eine Person nach Gewissheit strebt und oft die erste verfügbare Information aufgreift und sich daran festhält, anstatt offen für neue Informationen zu bleiben, die möglicherweise verfügbar werden [BeBa2007].

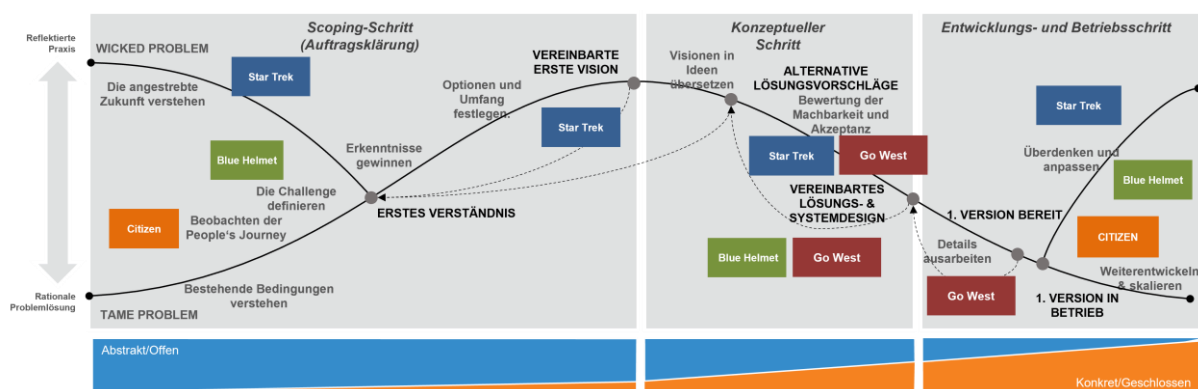


Abbildung 53 - Effektive Verteilung der Temperamente für die interdisziplinäre Zusammenarbeit bei der Erstellung einer digitalen Lösung [KEMP2017]

Praxistipp

Die Art und Weise, wie Informationen (konkret oder abstrakt) wahrgenommen werden, hängt von dem Arbeitsstil ab, der sich aus den spezifischen Disziplinen ergibt. Strategisches oder zukunftsorientiertes Denken erfordert ein Verständnis für Systeme und abstrakte Zusammenhänge (Star Trek). Umsetzungsorientierte Aufgaben erfordern oft eine detailorientierte Arbeitsweise (Go West). Personen mit hohen Empathiewerten (Blue Helmet oder supportorientierte Citizen) sind in der Lage, interdisziplinäre Teams zu führen oder zu unterstützen. Die Auswahl der richtigen Personen ist von entscheidender Bedeutung, insbesondere im Scoping-Schritt. Entscheidend sind die Personen, mit denen die Zukunft gestaltet oder validiert werden kann. Nur wenige Personen sind in der Lage, von aktuellen Bedarfen (aktuelle Nachfrage) auf zukünftige Bedarfe (attraktive Zukunft und zukünftige Nachfrage) abstrahieren zu können (Star Trek). Zu diesem Zeitpunkt ist es auch wichtig, dass permanentes Feedback verfügbar ist, da die zukünftige Nachfrage unbekannt ist. Dies erfordert kooperative Menschen mit einer zukunftsorientierten Denkweise (Blue Helmet). Blauhelme finden gemeinsam mit dem Kunden oder Kundenvertretern heraus, was in Zukunft attraktiv sein kann. Auf die Fähigkeit, sich die Zukunft vorstellen zu können, wird im folgenden Abschnitt näher eingegangen.

4.3.3.2 Zukunftsvorstellungen führen

Im Grunde genommen arbeitet jeder Bauprozess an der Schaffung einer angestrebten Zukunft (siehe Kapitel 1). In Abschnitt 4.2 haben wir das Three-Horizon-Model eingeführt, um mögliche Zeithorizonte für unterschiedliche Zukunftsvorstellungen zu beschreiben. Mit dem Verständnis über Persönlichkeitsmodelle können wir die Temperamente identifizieren, die für die Arbeit innerhalb eines bestimmten Horizonts besonders geeignet sind. Eigenschaften und Fähigkeiten des abstrakten Denkens sind beispielsweise erforderlich, wenn die Vorstellung der digitalen Lösung noch weit in der Zukunft liegt oder das Verständnis über zukünftige Zusammenhänge noch unklar ist (Horizont 2 & 3).

Abbildung 54 zeigt die drei Horizonte und die entsprechenden Temperamente:

- *Horizont 1 (Verbesserung des bestehenden Geschäfts)* erfordert detailorientierte Temperamente (Go West, Citizen), um sich auf die aktuelle Situation zu konzentrieren und den Status Quo zu verbessern.
- *Horizont 2 (innovative Lösungen/Dienstleistungen im Bereich der Organisation)* erfordert eine Kombination von Temperamenten, die mit abstrakten Ideen umgehen können (Star Trek) und sich gleichzeitig auf den Kunden/Benutzer fokussieren (Blue Helmet). Diese Kombination ermöglicht es Ihnen, neue Ideen zu entwickeln und sich gleichzeitig auf die Bedürfnisse der bestehenden Kunden/Benutzer zu konzentrieren.
- *Horizont 3 (visionäre Ideen in Bezug auf Lösungen und/oder Technologien)* erfordert Temperamente, die an Zukunftsideen arbeiten können (Star Trek). Dieses Temperament ermöglicht es Ihnen, den Lösungsraum zu öffnen und in verschiedenen Richtungen nach visionären Ideen zu suchen.

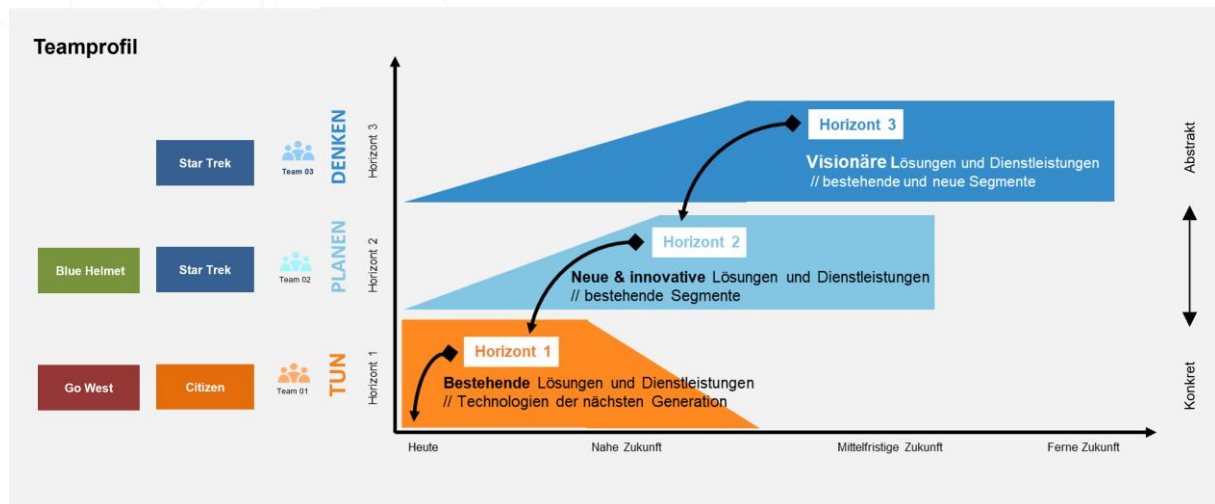


Abbildung 54 - Three-Horizon-Model und unterstützende Teamprofile

Dabei ist es wichtig zu erkennen, dass alle anderen Temperamente ebenfalls in den Horizonten eingesetzt werden können. Die Hauptaussage von Abbildung 54 ist, dass es den genannten Temperamenten leichter fällt, mit den spezifischen Herausforderungen eines jeden Horizonts umgehen zu können.

Praxistipp

Das Profil eines bestehenden Teams zu verstehen, ist ein erster Schritt, um mit der kognitiven Diversity des Teams zu arbeiten. Ein Teamprofil kann als Instrument verwendet werden, um die Temperamentsvielfalt innerhalb eines Umsetzungsteams zu visualisieren. Das Teamprofil ergibt sich aus der kumulierten Anzahl der Temperamente der Teammitglieder und kann mit einem Netzdiagramm visualisiert werden.

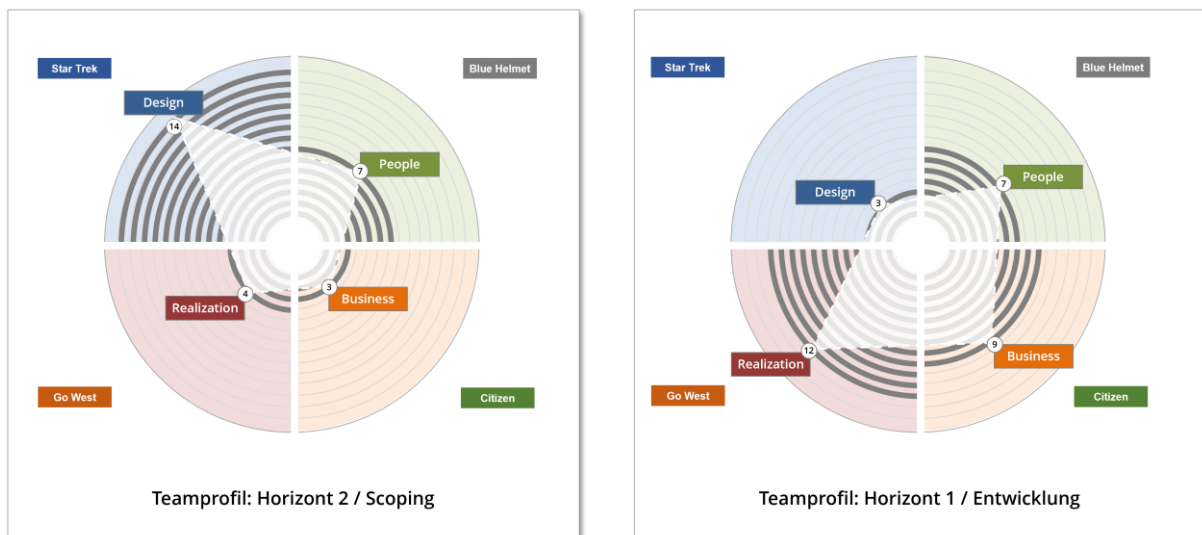


Abbildung 55 - Beispiele für Teamprofile

Abbildung 55 zeigt zwei Beispiele für Teamprofile:

- Fall Horizont 2 (linkes Diagramm): Hauptsächlich abstrakte, ideenorientierte (Star Trek) und auf den zukünftigen Kunden/Benutzer fokussierte Teammitglieder (Blue Helmet). Dieses Profil kann für den Beginn des Scoping-Schrittes geeignet sein.

- Fall Horizont 1 (rechtes Diagramm): Hauptsächlich detailorientierte (Go West, Citizen) und kooperative Teammitglieder (Citizen, Blue Helmet). Dieses Profil kann für den Beginn der Entwicklungsphase wirksam sein.

4.3.3.3 Management von Arbeitsstil, Schlüsselkompetenzen und Rollenzuweisung im Bauprozess

Aus praktischer Sicht sind drei Aspekte für das People Management während des Bauprozesses wichtig:

- Verstehen der allgemeinen Herausforderungen der einzelnen Bauprozess-Schritte in Bezug auf die Temperamente der Personen (Abschnitt 4.3.3.3.1)
- Verstehen der wichtigen Fertigkeiten für das digitale Zeitalter und ihre Beziehung zum Temperament (Abschnitt 4.3.3.3.2)
- Verstehen typischer Rollen im Bauprozess und ihrer Beziehung zu den einzelnen Schritten des Bauprozesses (Abschnitt 4.3.3.3.3)

Tabelle 20 - Wirksamkeit eines Temperaments während des Bauprozesses

Schritt	Wichtigste Herausforderung	Wirksamkeit der Temperamente	
<p>Scoping</p>	<p>Zukunft verstehen, Bedürfnis nach Orientierung, Vorstellung entwickeln, Richtung entscheiden</p>	<p><u>Rationale Problemlösung</u> Guardians oder Artisans beschreiben das konkrete Problem.</p>	<p><u>Reflektierende Praxis</u></p> <ul style="list-style-type: none"> • Durch ihre Vorstellungskraft gestalten die Rationals attraktive Zukunftsszenarien. • Rationals entdecken mit dem Sponsor das Narrativ und die Dringlichkeit zum Handeln, das von Idealists verständlich übersetzt und kommuniziert werden kann.
		<p>Input für die Definition des Digital Design Briefs liefern:</p> <ul style="list-style-type: none"> • Idealists sammeln Geschichten, nehmen kulturelle Kräfte wahr und filtern relevante Themen. • Rationals und Idealists entwerfen eine tragfähige Vision, einen Designrahmen und ermitteln zukünftige Stakeholder. Sie orientieren das Team, um ein Verständnis für das Ziel zu gewinnen. • Guardians oder Artisans beschreiben den Stakeholdern die die aktuelle Situation, die Ausgangssituation und akzeptierte Kultur auf der Grundlage ihrer Erfahrungen, als Indikator für die Veränderungsbereitschaft der beteiligten und betroffenen Menschen. 	

Schritt	Wichtigste Herausforderung	Wirksamkeit der Temperamente
Konzeption	Lösungsideen erforschen, den Kontext abgrenzen, die Vision klären, Netzwerke bilden	<ul style="list-style-type: none"> • (Branchenfremde) Rationals bieten Impulse zu neuen Ideen und hinterfragen diese gemeinsam mit unternehmensinternen Rationals, die bereit sind, bestehende auf die Probe zu stellen. Zusammen ergründen sie den Kontext und die Bedarfe und Anforderungen. • Idealists nehmen Erlebnisse und Bedarfe auf, entwickeln Anforderungen, gewinnen Fans in der Community, bauen Verständnis auf, bewerten Engagement und kümmern sich um den Teambuilding Prozess.
Entwicklung & Betrieb	Details erarbeiten, umsetzen und verbessern	<ul style="list-style-type: none"> • Idealists und Artisans definieren Storys, arbeiten Anforderungen aus und sorgen für Verständnis. • Guardians steuern den Bauprozess. • Idealists fördern den Teamprozess. • Guardians und Artisans setzen die Lösung um. • Rationals überprüfen die Qualität der Gesamtarchitektur und empfehlen Maßnahmen zur Weiterentwicklung. • Idealists erweitern die Community, prüfen die Akzeptanz bei den Entscheidungsträgern und empfehlen Maßnahmen zur Kommunikation.

4.3.3.3.1 Management des Arbeitsstils aus der Perspektive des People Managements

Wir haben gesehen, dass es einen Zusammenhang zwischen den Temperamenten der beteiligten Personen und ihrer Wirksamkeit für die verschiedenen Schritte des Bauprozesses gibt. People Management bedeutet in diesem Zusammenhang, Teammitglieder zu identifizieren, die für die Herausforderungen des jeweiligen Bauabschnitts besonders geeignet sind, und diesen Personen die Verantwortung für die Arbeit zu übertragen, um das individuelle Potenzial freizusetzen.

Praxistipp

Tabelle 20 fasst die Schritte des Bauprozesses zusammen, einschließlich der wichtigsten Herausforderungen (siehe Abschnitt 2.1.2), und erörtert die Eignung der verschiedenen Temperamente in Bezug auf die Herausforderungen. Die Tabelle kann als Übersicht und als Instrument für die Zuweisung von Personen und für die Bewertung des Profils bestehender Teams verwendet werden.

Jüngste Forschungsergebnisse zeigen, dass Teams mit einer größeren kognitiven Diversity in Bezug auf den Hang zur Entschiedenheit und die Offenheit gegenüber Mehrdeutigkeit besser abschneiden als Teams mit einer geringeren Vielfalt. Mit anderen Worten: Erfolgreiche Innovation setzt voraus, dass sowohl Personen mit hoher Ambiguitätstoleranz als auch solche mit geringer Ambiguitätstoleranz im selben Team arbeiten [BeBa2007]. Die Umsetzungsteams müssen daher zu einem bestimmten Zeitpunkt eine bewusste Entscheidung darüber treffen, welcher kognitive Stil ihre Aktivitäten und ihre Arbeitsweise dominieren soll. Zum Beispiel sind Menschen, die von Natur aus Designer sind, oft keine wirksamen Vorgesetzten.

Die Wahl der Arbeitsweise sollte sich an der Phase des Bauprozesses orientieren, da es in jeder Phase bevorzugte kognitive Stile gibt, die den besonderen Herausforderungen der jeweiligen Phase entsprechen. Die Rollenverteilung im Umsetzungsteam ist optimal, wenn

jedes Teammitglied die Rolle übernimmt, für die es am besten geeignet ist. Zum Beispiel ist in der Scoping Phase die Wahl des richtigen Temperaments wichtiger als die persönliche Erfahrung der Person, um alternative und möglicherweise radikale Lösungsideen zu finden.

4.3.3.3.2 Fähigkeiten für das digitale Zeitalter und ihre Beziehung zum Temperament

Das World Economic Forum hat eine Reihe von Fertigkeiten (Skills) definiert, die im digitalen Zeitalter besonders wichtig sind (siehe Tabelle 21, vgl. [WEF2016], [WEF2018]).

Tabelle 21 - Temperamente und Fähigkeiten für das digitale Zeitalter

Fertigkeiten für das digitale Zeitalter ([WEF2016], [WEF2018])	Temperament
<ul style="list-style-type: none"> Analytisches Denken/Innovation Aktives Lernen Technologie-Design Kreativität Kritisches Denken Komplexe Problemlösung 	Star Trek (NT, Rationals)
<ul style="list-style-type: none"> People Management Originalität Vernetzung mit anderen 	Blue Helmet (NF, Idealists)
<ul style="list-style-type: none"> Programmierung Vernetzung mit anderen 	Go West (SP, Artisans)
<ul style="list-style-type: none"> Kritisches Denken Evaluierung Vernetzung mit anderen 	Citizen (SJ, Guardians)

Diese Fertigkeiten sind besonders wichtig beim Bau und Design einer digitalen Lösung. Aus People-Management-Perspektive können sie auch mit Temperamenten in Verbindung gebracht werden.

Praxistipp

Tabelle 21 zeigt das entsprechende Temperament für jede Fertigkeit. Die Tabelle kann als Hilfsmittel verwendet werden, um die Personen für jede Fertigkeit zu identifizieren, denen das Erlernen dieser Fertigkeiten aufgrund ihres persönlichen Temperaments besonders leichtfällt.

4.3.3.3.3 Rollenzuweisung aus der People-Management-Perspektive

Rollen sind ein gängiges Instrument, um Personen in einem Bauprozess Verantwortlichkeiten zuzuweisen. Die Zuweisung von Rollen zu Personen kann durch das Verständnis ihrer Fertigkeiten und Erfahrungen, aber auch durch das Verständnis ihres individuellen Temperaments unterstützt werden. In der digitalen Wirtschaft werden viele Rollennamen und Titel verwendet. Tabelle 22 ist als People-Management-Werkzeug zur Unterstützung bei der Rollenzuweisung im Bauprozess gedacht. Sie zeigt typische Projektrollen, ihren Beitrag zur jeweiligen Phase des Bauprozesses, eine Rollen Metapher, eine Beschreibung ihrer Verantwortung und Temperamente, die dort wirksam sein können.

Praxistipp

Tabelle 22 kann zur Unterstützung der Rollenzuweisung für einen Bauprozess und zur Klärung der Verantwortungen einer Person innerhalb einer Rolle verwendet werden. Wenn beispielsweise eine Person die Rolle des Lösungsarchitekten übernimmt, besteht eine wichtige Aufgabe dieser Person darin, externe Impulse zu generieren, die das Team dazu anregen, neue Lösungsideen zu finden. Darüber hinaus kann Tabelle 22 verwendet werden, um die Eignung einer Person zu überprüfen, die eine Rolle im bestehenden Bauprozess hat.

Tabelle 22 - Rollen, Temperamente und Aufgaben

Typische Projektrolle	Bauprozess Fokus	Metapher	Verantwortung	Wirksames Temperament
Projektleitung Projektleiter Produktmanager	Alle Schritte	Orientierungshilfe	Intermediär „Ich evaluiere und manage das Geschäftspotential, Personen, Kontext, Kultur und Spielwiese!“	Star Trek, Blue Helmet
Auftraggeber	Scoping	Sponsor	Neues Business „Ich kann mir da etwas vorstellen.“	Star Trek, Blue Helmet
			Tame Problem „Ich habe ein Problem!“	Citizen
Erfinder Designer Lösungsarchitekt	Scoping, Konzeption	Erfinder	Externe Impulsgeber „Wir können es so machen!“	Star Trek
Business Designer Prozess-Experten	Scoping, Konzeption	Experte	Interne Kritiker „Ich breche die Regeln!“	Star Trek, Go West
Business-Analyst Requirements Engineer UX-Designer Interaktionsdesigner Berater Promoter	Konzeption, Entwicklung & Betrieb	Brückenbauer	Übersetzer „Abstrakt vs. Konkret“ „Ich sammle Storys, Anforderungen und übersetze Ideen & Konzepte. Ich werde euch Fans besorgen!“	Go West, Blue Helmet
Entwickler Softwarearchitekt Data Scientist Analyst	Entwicklung & Betrieb	Implementierer	Praktiker „Lass uns die Dinge jetzt tun!“	Go West, Citizen
Benutzer	Alle Schritte	Kunde	Feedback-Geber „AHA!“	

Schließlich ist es wichtig zu erkennen, dass der DDP nicht in Tabelle 22 vorkommt, weil wir den DDP nicht als eine Rolle im Bauprozess betrachten (siehe Abschnitt 1.4.3).

4.3.3.4 Führung während des Bauprozesses steuern

Gute Teams wechseln die Führung nach Bedarf, je nachdem, wo sie sich im Bauprozess befinden (vgl. [Pari2021]). Die Führung sollte nicht demjenigen übertragen werden, der als nächster an der Reihe ist, sondern demjenigen, der aufgrund seiner Persönlichkeitsstruktur für die aktuelle Phase des Bauprozesses am wirksamsten ist.

In diesem Sinne verhalten sich gute Teams wie Teams im Rennradsport, bei denen die einzelnen Personen während des Rennens aufgrund ihrer Stärken eingeteilt werden und nicht aufgrund ihres Dienstalters oder einer anderen Maßgabe. In diesen Teams ist jeder zu einem bestimmten Zeitpunkt in einer führenden Rolle und wird zu dem Zeitpunkt, an dem seine Fertigkeiten am meisten gebraucht werden, als Führungskraft respektiert.

Führen im Bauprozess bedeutet, den Bauprozess zu verstehen und sich zwischen Abstraktem und Konkretem sowie zwischen Analyse und Synthese zu bewegen, um einen effizienten und effektiven Prozess durchzuführen. Dazu gehört, die richtige Mischung von Personen im Team zusammenzustellen, um den Prozess durchzuführen, und Führungsrollen für das Team bereitzustellen. Personen, die die Führung übernehmen, verfügen nicht nur über die klassischen Führungsqualitäten, sondern verstehen auch den Prozess und sind in der Lage, die verschiedenen Denkweisen, die im Team vertreten sind, reibungslos zu nutzen und zu integrieren [BeBa2007]. Das breite Spektrum des Bauprozesses und die damit verbundenen Anforderungen an die Persönlichkeit machen deutlich, dass es fast unmöglich ist, dass eine Person allein den gesamten Bauprozess leitet. Es ist eine Herausforderung für die Teamleitung hierzu jeweils die richtigen Personen zu identifizieren. Abbildung 57 hebt die vorgeschlagenen Temperamente in der führenden Position zu diesem bestimmten Zeitpunkt des Prozesses hervor.

Schließlich ist jeder Bauprozess auch ein Prozess der Orientierung und der kontinuierlichen Erkenntnis in einem interdisziplinären Umfeld. Um dies zu erreichen, muss das Team auch in der Lage sein, Personen so zu inspirieren, dass sie sich die Zukunft vorstellen zu können - erst dann können sie eine Erfolgslogik schaffen, die für das Team und die relevanten Stakeholder akzeptabel ist

Praxistipp

Um bestimmte Aufgaben bei der Leitung eines interdisziplinären Umsetzungsteams wahrnehmen zu können, benötigt die Führungskraft ein mit ihrer eigenen Persönlichkeit fest verankertes Mindset [Hüth2009]:

- Um Personen zur Teilnahme an einem Gestaltungsprozess einladen zu können, sind Persönlichkeiten wirksam, die Menschen mögen und ihr Potential zur Entfaltung bringen wollen.
- Um Personen für die Gestaltung einer attraktiven Zukunft zu begeistern, sind Persönlichkeiten wirksam, die selbst von Neuem fasziniert sind.
- Um Personen zu ermutigen, gegen gängige Regeln zu handeln und zu hinterfragen, sind Persönlichkeiten wirksam, die sich nicht scheuen, Risiken einzugehen und ehrlich gegenüber sich selbst sind.

Die wichtigsten Eigenschaften, die für diese Aufgaben erforderlich sind, sind Empathie und ein starkes Selbstbewusstsein.

Darüber hinaus ist es wichtig, Rollen und Zuständigkeiten zu klären, Arbeitsstrukturen zu schaffen und die Bedeutung und Auswirkung der Arbeit des Teams hervorzuheben. Der Schlüssel liegt darin, ein Teamumfeld mit psychologischer Sicherheit zu schaffen, in dem die Teammitglieder Risiken eingehen und Entscheidungen, Ideen und Anweisungen in Frage stellen können, ohne sich unsicher oder peinlich berührt zu fühlen [GOOG2005].

Eine effiziente Kombination eines Führungsteam-Profiles während des gesamten Bauprozesses, vor allem in der Scoping- oder Konzeptphase, können Star Treks und Blue Helmets sein. Dieses Führungsprofil schafft ein Arbeitsumfeld für Innovation, zukunftsorientiertes Denken, Wissenserwerb und Erkenntnis sowie für ein Big Picture Design unter aktiver Einbeziehung von Teammitgliedern, Kunden und Benutzern.

Je stabiler die Erkenntnis zur Lösung wird, desto stärker wird auch der Bedarf zu einer Entscheidung zu kommen, sodass ein Wechsel im Führungsprofil in Richtung eines Profilgefüges aus extravertierten und entschiedenen Star Treks, extravertierten und kooperativen Citizens und pragmatischen Go Wests stattfinden kann.

Diese Profile schaffen ein Arbeitsumfeld für die Umsetzung, die Entscheidungssicherheit und die operative Unterstützung (siehe Abbildung 53).

4.3.3.5 Berücksichtigung des Arbeitsumfelds und Verständnis des Kontextes

Die Bauorganisation (d. h. die Organisation, die den Bauprozess durchführt) hat einen großen direkten und indirekten Einfluss auf die digitale Lösung. Auch die Personen innerhalb der Organisation müssen die digitale Lösung akzeptieren und unterstützen, damit der Bauprozess und damit die digitale Lösung ein Erfolg werden können. Andernfalls kann die Organisation den Bauprozess behindern oder ihn sogar zum Scheitern bringen [VPGV2008]. Das Arbeitsumfeld (z. B. Zeit, Arbeitsraum, psychologische Sicherheit, Führungsstil oder klare Erwartungen) hat einen entscheidenden Einfluss auf die Handlungen und das Verhalten der Menschen und damit auf ihre Bereitschaft, sich zu engagieren.

Eine weitere Motivation für das Engagement ist das Verständnis für die Zusammenhänge im Projekt und für die persönlichen Auswirkungen: Dazu unterstützt die inhaltliche Dimension (siehe Abschnitt 2.2) bei der Identifikation des Informationsbedarfs der am Bauprozess beteiligten Personen.

Praxistipp

Folgende Punkte sollten während des Bauprozesses berücksichtigt werden, um ein gemeinsames Verständnis zu erreichen:

- Der permanente Perspektivwechsel zwischen der Bauorganisation und ihrer Partner
- Die Perspektive der zukünftig betroffenen Kunden und Benutzer

Die Beteiligten der Bauorganisation haben Einfluss auf die Gestaltung und Umsetzung einer Lösung, Kunden und Benutzer wiederum haben Einfluss auf die Akzeptanz. Jeder Stakeholder dieser beiden Gruppen (Beteiligte und Kunden/Benutzer) hat ein eigenes Verständnis über die Vorstellung der Lösung, verfolgt individuelle Motive und emotionale Bedürfnisse. Überprüfen Sie das Verständnis und das Commitment anhand einer Stakeholderliste und achten Sie auf

die Bedürfnisse der einzelnen Stakeholder, um Kommunikationsstrategien zu entwickeln und die Teamzusammensetzung zu ändern.

Die Einbeziehung von Repräsentanten zukünftiger Kunden oder Benutzer ist von großer Bedeutung für die Akzeptanz, insbesondere für das Design digitaler Lösungen. Die folgenden Aspekte sind als Checkliste für die Einbeziehung künftiger Kunden oder Benutzer gedacht:

- Die Kunden, *die von der Lösung betroffen sind*, brauchen die Lösung, um in der entsprechenden Situation einen klaren Mehrwert für ihre Bedarfe zu erhalten. Aber auch während des Bauprozesses gibt es anonyme Kunden (z. B. Persona) oder Vertreter, die den Co-Creation Designprozess unterstützen. Sie erwarten frühe und anfassbare Erlebnisse und ein Umfeld, in dem ihre Kritik gehört und berücksichtigt wird.
Aufgabe für das People Management: Die Kundenperspektive berücksichtigen, kontextbezogene Kundenmotive verstehen und eine Feedback-Kultur etablieren.
- Schaffen Sie Kundenakzeptanz für das Verfahren. Dies ist wichtig für die Mitgestaltung, damit auch der Kunde ehrlich einbezogen wird.
Aufgabe für das People Management: Erklären Sie den Kunden, die am Co-Creation Prozess beteiligt sind, den gesamten Ablauf und die Methoden. Erläutern Sie die Verantwortung, die Art der Beteiligung und inwiefern sowohl das Team aber auch das Verständnis der Lösungsreife von der Mitwirkung der Kunden und Benutzer profitiert.
- *Verständnis über die Auswirkungen:* Für die betroffenen Kunden verringert das Verständnis über zukünftige Zusammenhänge – Herausforderungen, Trends oder Lösungen – das Gefühl der Besorgnis.
Aufgabe für das People Management: Je nach Dauer des Bauprozesses merkt die betroffene zukünftige Kunden- und Benutzergruppe, dass etwas im Gange ist. Organisieren Sie so früh wie möglich einen Kommunikationsplan mit Personen, die für den Wandel Verantwortung übernehmen. Bieten Sie Raum für Beteiligung und Dialog.

4.3.4 Fazit

Für einen DDP auf Foundation Level Niveau ist die wichtigste Schlussfolgerung aus diesem Abschnitt, dass das People Management eine wichtige Querschnittskompetenz darstellt und ein zentraler Erfolgsfaktor für den Aufbau einer digitalen Lösung ist: Es reicht nicht aus, die Fertigkeiten, Erfahrungen und Verfügbarkeiten der Teammitglieder zu verwalten - der DDP muss auch Persönlichkeitsindikatoren, eine effektive Diversität der Teammitglieder und die Gesundheit des Teams berücksichtigen.

Dieser Abschnitt ist als eine erste Einführung in die Welt des People Managements gedacht. Dazu haben wir das Keirsej-Modell als Grundlage verwendet. Fachleute aus der Psychologie nutzen auch andere Modelle, die auf das People Management angewendet werden können (z. B. den MBTI oder die Big Five Persönlichkeitsdimensionen). Wir haben uns für das Keirsej-Modell entschieden, da wir wissen, dass dieses Modell einen guten Ausgangspunkt für Einsteiger darstellt und einen guten Einblick in die Herausforderungen und Aufgaben des People Managements bietet. Die wichtigsten Erkenntnisse für das People Management im Bauprozess sind die verschiedenen Kommunikationsstile und die damit verbundenen Motive und Verhaltensmuster:

- Verstehen: Zukunftsorientierter Theoretiker vs. konkreter Umsetzer; abstraktes vs. konkretes Wahrnehmen und Lernen
- Handlungsorientierung: Kooperativ, personenorientiert vs. sachorientiert
- Führungsanspruch und Kommunikationsstil: Direktiv vs. informativ
- Impulsivität, Energie: Ausdrucksstark vs. aufmerksam, reflektierend

Eine weitere Schlussfolgerung ist, dass es mehrere Herausforderungen für den Bauprozess gibt, die bewältigt werden können, wenn man das People Management ernst nimmt.

- Ein gutes Umsetzungsteam braucht Rollen, die Brücken schlagen und zwischen abstrakten Denkern bzw. Designern und konkreten Umsetzern vermitteln. Es bedarf einer Person, die als Guide die Menschen verbindet, die sich sonst nur auf das Business, die Lösung oder auf das digitale System fokussieren.
- Menschen brauchen ein Arbeitsumfeld, das die Bewältigung ihrer Aufgaben unterstützt: Neben den notwendigen Rahmenbedingungen sollte auf eine wirksame Kultur der Bauorganisation sowie auf ein Regelwerk geachtet werden, das psychologische Sicherheit für jeden Einzelnen gewährleistet.
- Ein gemeinsames Verständnis der Zusammenhänge unter allen Stakeholdern ist ein unterschätzter Erfolgsfaktor, wenn es darum geht, dass Menschen sich für Erneuerung und Wandel engagieren.
- Der Zielhorizont der digitalen Lösung und der Schritt im Bauprozess erfordern ein wirksames Teamprofil, das sich im Laufe des Bauprozesses verändern kann - von einem abstrakt denkenden, offeneren Team im Scoping-Schritt hin zu einem Team, das konkret wird und entschieden ist.
- Die Führung sollte der Person übertragen werden, die am besten qualifiziert und von der Persönlichkeitsstruktur her am besten für die aktuelle Phase des Bauprozesses geeignet ist. Gute Teams wechseln die Führung nach Bedarf, je nachdem, wo sie sich im Bauprozess befinden.

Eine abschließende Schlussfolgerung für den DDP auf Foundation Level Niveau ist, dass People Management beim DDP selbst beginnt: Jede DDP zertifizierte Person sollte die eigene Persönlichkeit mit ihren individuellen Motiven und Werten kennen. Jeder sollte sich über die eigene Wirkung auf andere bewusst sein, um Individualität und Diversity im Bauprozess zu erkennen, zu differenzieren und um mit Andersartigkeit umgehen zu können.

5 Ein Bauprozess für Einsteiger

Dieses Kapitel enthält Leitlinien für einen vollständigen Bauprozess für eine digitale Lösung (siehe Abschnitt 2.1.2). Die Leitlinien beruhen auf den Erfahrungen der Autorinnen und Autoren dieses Handbuchs. Einsteiger sollten sich darüber im Klaren sein, dass es viele andere Ansätze für den Bau einer digitalen Lösung gibt.

Andere Ansätze zu studieren ist wichtig und andere Ansätze mit eigener praktischer Erfahrung zu studieren ist noch besser. Die Intention dieser Leitlinien ist es daher, Einsteigern einen Ausgangspunkt zu bieten, um eigene Erfahrungen zu sammeln und eigene Bauprozesse zu entwickeln. Erfahrene Leser werden andere Ansätze kennen und in der Lage sein, einen ganz anderen Prozess zu definieren oder sogar die vorgestellten Techniken in anderen Schritten oder Situationen anzuwenden.

Die vorgestellten Leitlinien setzen voraus, dass der Leser die Person ist, die für die digitale Lösung und das Management des Bauprozesses verantwortlich ist. In Anlehnung an die Terminologie der agilen Entwicklung bezeichnen wir diese Person als Product Owner: Der Product Owner ist die verantwortliche Person, die von der Auftraggeber-Rolle ermächtigt ist, in Absprache mit dem Umsetzungsteam, dem Auftraggeber und anderen Stakeholdern über alle Fragen im Zusammenhang mit der digitalen Lösung zu entscheiden.

Die vorgestellten Leitlinien setzen weiterhin voraus, dass der Product Owner mit einem Umsetzungsteam arbeitet, das über alle notwendigen Fähigkeiten verfügt, um die digitale Lösung zu bauen. Ein Mitglied dieses Teams wird als Umsetzungsteam-Mitglied bezeichnet. Da diese Leitlinien den gesamten Bauprozess abdecken, gehen sie über den Rahmen von Digital Design hinaus. Diese breitere Perspektive ist jedoch notwendig, um die Integration von Digital Design in den Bauprozess zu verstehen.

Wir schließen dieses Kapitel in Abschnitt 5.4 mit einer Erörterung von *Lean Startup* als einem weiteren wichtigen Ansatz für den Start einer neuen digitalen Lösung ab.

5.1 Leitlinien für den Scoping-Schritt

Das Ziel des Scoping-Schritts ist es, die Vision, den Kontext, den Umfang und die allgemeinen Bedingungen der digitalen Lösung (siehe 2.1) zu definieren.

In Abschnitt 2.1 haben wir den Begriff „Wicked Problems“ eingeführt, um zwei Kategorien von Designproblemen zu charakterisieren (Wicked vs. Tame Problems). Obwohl dieser Begriff für ein erstes Verständnis des Spektrums von Designproblemen sehr nützlich ist, bietet er nur wenig Orientierung für die praktische Arbeit.

Ein Instrument zur Messung von Verständnis und Engagement

Aufbauend auf den Kategorien Wicked Problem und Tame Problem stellt Abbildung 56 ein Instrument zur Messung des Verständnisses einer digitalen Lösung während des Bauprozesses vor. Ein Product Owner kann dieses Werkzeug einsetzen, um das Verständnis der Stakeholder und der Mitglieder des Umsetzungsteams während des gesamten Bauprozesses zu bewerten.

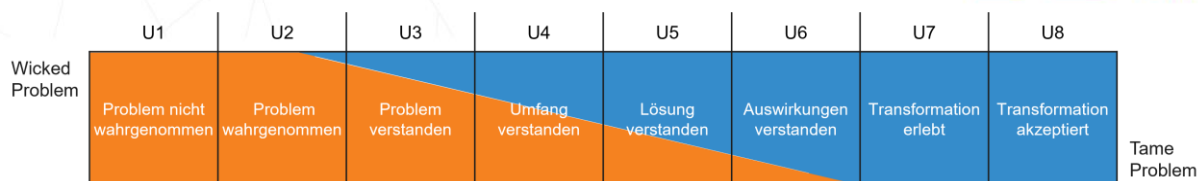


Abbildung 56 - Ein Instrument zur Messung des Verständnisses für eine digitale Lösung

Auf der linken Seite befindet sich ein Wicked Problem. Das Problem ist nicht klar definiert und wird von vielen Menschen nicht einmal wahrgenommen. Auf der rechten Seite gibt es ein Tame Problem, das jeder akzeptiert und das durch die digitale Lösung gelöst werden muss. Von links nach rechts nimmt die Unlösbarkeit des Wicked Problems ab und das Problem wird immer handhabbarer (Tame Problem).

Neben dem Verständnis einer digitalen Lösung ist es wichtig, den Grad des Engagements der Stakeholder zu kennen. Abbildung 57 stellt ein Instrument zur Messung des Engagements von Stakeholdern vor: Ein Stakeholder der Kategorie C++ setzt sich proaktiv für die neue digitale Lösung ein und will sie zum Erfolg führen. Ein Stakeholder der Kategorie C-- arbeitet proaktiv gegen die digitale Lösung.

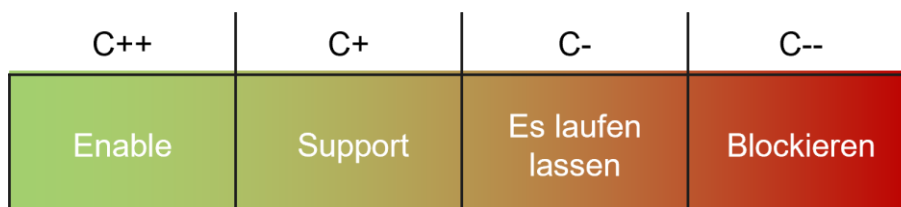


Abbildung 57 - Ein Instrument zur Messung des Grades des Engagements

Wahl eines Ansatzes für den Umgang mit dem Problem auf der Grundlage von Verständnis und Engagement

Im Folgenden werden die beiden oben beschriebenen Werkzeuge verwendet, um verschiedene Verfahren während des Bauprozesses zu erläutern.

Zu Beginn des Scoping-Schrittes für die neue digitale Lösung empfehlen wir zwei Aktivitäten:

- 1) Machen Sie sich ein Bild von den Temperamenten und der Persönlichkeit des Teams, mit dem Sie zusammenarbeiten. Verwenden Sie die in Abschnitt 4.3 vorgestellten Ideen und versuchen Sie, die Ratschläge für den Bauprozess zu nutzen.
- 2) Verschaffen Sie sich ein klares Bild über den Kenntnisstand aller relevanten Stakeholder. Wenn das Verständnis der relevanten Stakeholder zwischen den Stufen U1 und U5 liegt (siehe Abbildung 56), befolgen Sie unsere Leitlinien für ein Wicked Problem (Abschnitt 5.1.1). Andernfalls sollten die Leitlinien für ein Tame Problem befolgt werden (Abschnitt 5.1.2)

In einem letzten Schritt stellen wir Leitlinien für die Definition der allgemeinen Begriffe zur Verfügung, die Teil des Digital Design Briefs sind.

5.1.1 Scoping eines Wicked Problems

Die Aufgabe des Scoping-Schrittes für ein Wicked Problem ist es, das Verständnis des Problems zu verbessern und mögliche Lösungswege zu erkunden.

Design Thinking als Ansatz für das Scoping von Wicked Problems

Wir empfehlen für diese Situation Design Thinking [Brow2009]. Design Thinking ist ein beliebter Ansatz und besteht aus einem iterativen Prozess mit klaren Regeln, der die Bedeutung des Testens/Validierens von Ideen mit frühen Prototypen hervorhebt. Aufgrund des breiten Spektrums an Ressourcen, die bereits zur Verfügung stehen, gehen wir nicht näher auf das Design Thinking ein. Stattdessen konzentrieren wir uns nur auf die besonderen Aspekte, die für die Anwendung von Design Thinking im Scoping-Schritt wichtig sind.

Die Regeln des Design Thinking empfehlen, dass ein interdisziplinäres Team den Design-Thinking-Prozess durchführt. Der Design-Thinking-Prozesses liefert drei Ergebnisse:

- Dokumentation der verschiedenen Facetten des „Wicked Problems“
- Eine Reihe von Lösungsideen, die anhand von frühen Prototypen validiert wurden
- Ein Team von Personen mit einem detaillierten Verständnis des Problems (einschließlich der Motivation der Auftraggeber-Organisation) und der Lösungsideen

Das letzte Ergebnis (Personen mit detailliertem Verständnis) wird in der Praxis oft unterschätzt. Die Teilnehmer eines Design-Thinking-Prozesses gewinnen wichtige Erkenntnisse, die für den gesamten Bauprozess sehr nützlich sind. Wir empfehlen daher dringend, das Design-Thinking-Team während des gesamten Bauprozesses so stabil wie möglich zu halten, um von diesem Verständnis zu profitieren.

5.1.1.1 Phase 1: Zusammenstellung eines Design-Thinking-Teams

Um einen Design-Thinking-Prozess erfolgreich durchzuführen, ist es wichtig, ein interdisziplinäres Team (8-10 Personen) zusammenzustellen. Eine systematische Stakeholder-Analyse ist ein guter Ausgangspunkt für die Zusammenstellung des Teams. Das Zwiebschalenmodell [Alex2005] ist ein einfaches Instrument für diese erste Analyse. Es unterscheidet zwischen den direkten Stakeholdern des Systems (z. B. Benutzer, Betreiber), den Stakeholdern aus dem direkten Systemumfeld - also aus Digital Design Sicht der digitalen Lösung - (z. B. Eigentümer, Sponsoren), und den Stakeholdern des weiteren Umfeld (z. B. juristische Personen, Normungsgremien). Darüber hinaus sollten die Leitlinien aus Abschnitt 4.3 zu den Temperamenten berücksichtigt und das Verständnis und der Grad des Engagements (siehe Abbildung 57) ebenfalls bewertet werden (z. B. durch kurze Interviews).

Das Ergebnis dieser Stakeholder-Analyse sollte im Digital Design Brief dokumentiert werden (siehe Abschnitt 2.2.2).

5.1.1.2 Phase 2: Durchführung des Design-Thinking-Prozesses

Für die Durchführung eines Design-Thinking-Prozesses auf Foundation Level Niveau sind die folgenden Empfehlungen als Ausgangspunkt hilfreich:

- Stellen Sie ein Team von 8-10 Personen zusammen und nutzen Sie die Ergebnisse der Stakeholder-Analyse. Versuchen Sie, eine gute Mischung von Personen zu finden, die sich gut verstehen. Versuchen Sie, Personen zu vermeiden, die die digitale Lösung blockieren könnten (Verpflichtung C--). Der Product Owner ist Teil des Teams. Es wird empfohlen, einen unvoreingenommenen Moderator hinzuzuziehen, der den Prozess leitet.

- Versuchen Sie, auch Personen einzubeziehen, die Erfahrung in der Konstruktion und Realisierung haben. Zu einem optimalen Team gehört mindestens ein Teilnehmer, der für das spätere Entwicklungs- und Betriebsteam vorgesehen ist.
- Definieren Sie einen klaren Zeitplan für den Prozess. Planen Sie Zeit für die Nachbearbeitung der Ergebnisse ein. Wenn die Zeiteinteilung unklar ist, planen Sie zwei Wochen für das Design Thinking und drei Tage für die Nachbereitung ein. Wenn die Mitarbeiter nicht Vollzeit zur Verfügung stehen, sollten Sie einen klaren Zeitplan aufstellen (z. B. halbtags in einem Zeitfenster), damit die Mitarbeiter ihre anderen Aufgaben gut organisieren können.
- Bestimmen Sie den Zeithorizont, auf den sich das Problem bezieht (verwenden Sie das Horizontmodell, siehe Abschnitt 4.2). Öffnen Sie den Lösungsraum so weit wie möglich, je nach dem Zeithorizont, den Sie berücksichtigen wollen. Fördern Sie eine zukunftsorientierte Denkweise als Kernkultur des Teams. Sammeln Sie alle für den Zeithorizont erforderlichen Informationen (z. B. Fallstudien über zukünftige Trends und neue Technologien). Laden Sie Fachleute für innovative digitale Technologien (z. B. Experten für künstliche Intelligenz oder Blockchain), analoge Technologien (z. B. Materialwissenschaftler) und andere Wissenschaftler (z. B. Sozialwissenschaftler) dazu ein, Vorträge zu ihren Themen zu halten, um das Design Thinking-Team zu inspirieren.
- Schließen Sie auch Techniken zur Förderung der Empathie mit zukünftigen Kunden, Benutzern und anderen Stakeholdern ein. Erstellen Sie zum Beispiel mit dem Team verschiedene Personas oder entwickeln Sie eine Customer Journey (siehe Abschnitt 2.2.3).
- Geben Sie die Erlaubnis für offenes Denken und eine zukunftsorientierte Denkweise. Das Scoping einer digitalen Lösung erfordert ein offenes Denken und eine Vorausschau auf die Auswirkungen verschiedener digitaler Materialien. Schaffen Sie ein Umfeld, in dem sich die Menschen frei fühlen, offen und weit in die Zukunft zu denken (vgl. z. B. [GOOG2020]).
- Verwenden Sie einfache Prototypen wie Papierprototypen und Storyboards (siehe Abschnitt 2.3).
- Nutzen Sie die Pressemitteilung aus der Zukunft (siehe Abschnitt 2.2.3) als temporäres Artefakt, um verschiedene Visionen in Ihrem Team zu entwickeln.
- Suchen Sie als Inspirationsquelle nach Konkurrenten, verwandten Lösungen und Mustern, die ganze Branchen verändern. Dokumentieren Sie diese auch im Digital Design Brief.
- Um die entwickelten Ideen zu bewerten, sollten Sie ein breites Spektrum von Stakeholdern - bezüglich Verständnis und Engagement - einladen. Es kann schwierig sein, Stakeholder mit einem negativen Engagement (C-, C--) davon zu überzeugen, sich an einer Evaluierung zu beteiligen. Unserer Erfahrung nach können sie jedoch kritische Beiträge liefern, die zur Verbesserung der Ideen beitragen, und die Evaluierung der Ideen kann dazu beitragen, diese Stakeholder davon zu überzeugen, die neue Lösung zu unterstützen.

5.1.1.3 Phase 3: Dokumentieren der Ergebnisse und ggf. Iteration

Am Ende des Design-Thinking-Prozesses wird eine Menge Material entstanden sein. Der Product Owner ist für die Bewertung dieses Materials verantwortlich. Wenn die Ergebnisse nicht überzeugend sind, planen Sie einen weiteren Design-Thinking-Prozess. Aus Sicht von

Digital Design muss Design Thinking nicht nur Teambildung und die Ausbildung einer gewünschten Arbeitshaltung ermöglichen, sondern auch konsequent als ergebnisorientierter Ansatz verstanden werden.

Die Vorlage des Digital Design Brief dient als Leitfaden für die Dokumentation der Ergebnisse (siehe Abschnitt 2.2.2). Eine Pressemitteilung aus der Zukunft (siehe Abschnitt 2.2.3) ist ein gutes Instrument, um die Vision einer digitalen Lösung im Sinne einer angestrebten Zukunft zu beschreiben. Der Design-Thinking-Prozess kann unterschiedliche Lösungsideen hervorbringen. Im Digital Design Brief sollten alle Ideen dokumentiert werden.

5.1.2 Scoping eines Tame Problems

Ein Wicked Problem ist eine Herausforderung für jeden Bauprozess. Ein Problem, das auf den ersten Blick wie ein Tame Problem erscheint, sollte jedoch nicht unterschätzt werden. Unsere eigene Erfahrung hat uns gelehrt, dass viele Probleme auf den ersten Blick wie ein Tame Problem erscheinen und sie erst dann als Wicked Problem deutlich werden, wenn wir uns die Details und den kontextuellen Bereich des Konflikts ansehen (siehe Abschnitt 2.1.2). Jeder Einsteiger im Digital Design sollte dies als ernste Warnung verstehen.

Systematische Analyse als Ansatz zum Scoping eines Wicked Problems

Die Aufgabe des Scoping-Schrittes für ein Tame Problem besteht daher darin, das Problemverständnis (und die Handhabbarkeit des Problems) zu hinterfragen und eine klare Vision für die digitale Lösung zu definieren. Wir empfehlen für diese Situation einen analyseorientierten Ansatz, der aus einer Reihe von Interviews und einem kompakten Scoping-Workshop-Format besteht.

Auch wenn die potenziellen Stakeholder bekannt sind, sollte eine systematische Stakeholder-Analyse (siehe Abschnitt 5.1.1) durchgeführt werden. Auf diese Weise können wichtige Stakeholder, die bisher übersehen wurden, ermittelt und zu dem Workshop eingeladen werden. Versuchen Sie, wichtige Stakeholder mit geringem Verständnis (U1-U3) für die Lösung zu identifizieren und laden Sie sie zum Workshop ein. Diese Stakeholder machen den Schritt schwieriger, verbessern aber definitiv das Ergebnis im Hinblick auf ein detailliertes Verständnis.

5.1.2.1 Phase 1: Befragung wichtiger Stakeholder

Alle ermittelten wichtigen Stakeholder sollten vom Product Owner befragt werden, um ihr Verständnis des Problems und ihr Engagement zu erfassen. Die Kapitel 1-3 des Digital Design Briefs können als grober Interview-Leitfaden dienen. Wenn das Problem es erfordert, sollten weitere Fachleute hinzugezogen werden. Für Einsteiger sollte die Maxime immer sein, alle Interviews selbst zu führen, um das eigene Verständnis für das Problem zu vertiefen. Wir empfehlen außerdem eine zusätzliche Person, die bei der Dokumentation des Gesprächs hilft.

Beginnen Sie mit einem Interview der Person in der Auftraggeber-Rolle, um deren Motivation für das Projekt, den Grad des Problemverständnisses und die Gründe für das Handeln zu ermitteln.

Die Ergebnisse dieser Interviews können im Digital Design Brief dokumentiert werden. Wenn Sie auf diese Weise kein klares Bild des Problems erhalten, empfehlen wir Ihnen, auf den Ansatz für „Wicked Problems“ auszuweichen (siehe Abschnitt 5.1.1).

Wenn die Ergebnisse ein klares Bild des Problems ergeben, empfehlen wir einen Scoping-Workshop, um die Ergebnisse der Interviews mit den ermittelten relevanten Stakeholdern zu diskutieren. Das Ziel des Scoping-Workshops ist es, das Problemverständnis im Detail zu hinterfragen. Zu diesem Zweck ist eine gute Mischung von Stakeholdern mit positivem und negativem Engagement der Schlüssel zu einem realistischen Feedback.

5.1.2.2 Phase 2: Scoping-Workshop

Zur Vorbereitung des Scoping-Workshops erarbeitet der Product Owner eine Pressemitteilung aus der Zukunft, ein Storyboard und Papierprototypen, um die Gesamtvision der digitalen Lösung zu visualisieren. Berücksichtigen Sie bei der Ausarbeitung des Materials den Kenntnisstand und das Engagement der Stakeholder. Fügen Sie eine angemessene Beschreibung und Erläuterung des vorgestellten Materials bei, damit auch Personen mit geringem Kenntnisstand die Möglichkeit haben, den Stoff zu verstehen und nachzuvollziehen.

Dieses Material kann den am Workshop Teilnehmenden vorab zur Vorbereitung zugesandt werden, dies ist aber nicht unbedingt erforderlich. Beide Optionen sind möglich und hängen von der jeweiligen Situation ab:

- Wir empfehlen die Zusendung von Vorbereitungsmaterial, wenn das Thema recht kompliziert ist und ein gründliches Studium durch die Teilnehmenden erfordert. Die Teilnehmenden sind eingeladen, dem Product Owner vor dem Workshop Feedback zu geben. Schließlich sammelt der Product Owner das Feedback und analysiert es.
- Wenn Sie die unmittelbare Reaktion der Beteiligten sehen wollen, sollten Sie das Material nicht zur Vorbereitung verschicken, sondern es direkt im Workshop vorstellen. So können Sie unmittelbare Reaktionen und spontane Eindrücke festhalten und sofort diskutieren.

Wir empfehlen die folgenden Schritte als Agenda für einen Scoping-Workshop:

Schritt 1: Der Product Owner präsentiert die Pressemitteilung aus der Zukunft und das Feedback

Alle Teilnehmenden diskutieren das Feedback und die Arbeit und einigen sich auf eine endgültige Pressemitteilung aus der Zukunft. Die Diskussion sollte sich vor allem auf die Schattenseiten der beschriebenen Zukunft konzentrieren. Ist die beschriebene Zukunft für die Menschen, die Wirtschaft und die Gesellschaft wirklich wünschenswert? Wie könnte die beschriebene Zukunft aus einer negativen Sicht wahrgenommen werden? Hier können Stakeholder mit einem negativen Engagement eine wertvolle Quelle für Erkenntnisse sein.

Nehmen wir als Beispiel die Pressemitteilung aus der Zukunft zum YPRC. Eine negative Sichtweise auf den YPRC kann von professionellen Lauftrainern kommen. Sie könnten den YPRC als eine Bedrohung für ihr eigenes Geschäft betrachten. Wie könnten sie reagieren, wenn die Lösung auf dem Markt präsentiert wird?

Schritt 2: Der Product Owner präsentiert das Storyboard, die Papierprototypen und das Feedback

Alle Teilnehmenden diskutieren das Feedback und entscheiden über mögliche Änderungen/zusätzlichen Input. Die Diskussion sollte sich vor allem auf die Ausnahmesituationen und deren Auswirkungen auf die Lösung konzentrieren. Die Gruppe identifiziert dabei Schlüssel-Storys (z. B. Ausnahmesituationen) und relevante Themen (z. B.

Themen, die mit der Lösung in Zusammenhang stehen), die den Umfang der Lösung genauer umreißen.

Eine Ausnahmesituation könnte ein Remote-Coach sein, der einen Läufer so stark belastet, dass er während einer Trainingseinheit zusammenbricht. Wie fühlt sich das für Sie an, wenn Sie sich in die Rolle der Menschen versetzen, die den YPRC gestalten?

Schritt 3: Der Product Owner präsentiert den ausgearbeiteten Digital Design Brief einschließlich des Feedbacks

Alle Teilnehmenden diskutieren das Feedback und entscheiden über mögliche Änderungen/zusätzlichen Input. Dabei sollten vor allem die Einzelheiten der Mittelzuweisung und des Zeitrahmens erörtert werden. Die Teilnehmenden sollten sich vor allem auf die kritischen Aspekte der allgemeinen Begriffe konzentrieren. Was geschieht, wenn das Budget nicht ausreicht oder der Zeitrahmen zu kurz ist?

Der konkrete Zeitplan hängt von der Menge des Materials und dem Niveau des gemeinsamen Verständnisses ab (siehe Abschnitt 4.3). Bis zu drei Tage Workshopzeit sind angemessen. Wenn dieser Zeitrahmen zu kurz erscheint, sollten mehrere Workshops zu bestimmten Unterthemen geplant werden.

5.1.2.3 Phase 3: Dokumentieren Sie die Ergebnisse und iterieren Sie, falls erforderlich

Am Ende des Workshops aktualisiert der Product Owner den Digital Design Brief anhand des Feedbacks und erstellt eine endgültige Version. Wenn es nicht möglich ist, eine finale Version zu erstellen, sollten eine zweite Interviewrunde und ein zweiter Workshop angesetzt werden, um die offenen Fragen zu klären. Sollte auch die zweite Runde zu keinem eindeutigen Ergebnis führen, empfehlen wir, das Verfahren auf einen Wicked-Problem-Ansatz umzustellen (siehe Abschnitt 5.1.1).

5.1.3 Definition der allgemeinen Bedingungen für den Bau einer digitalen Lösung

Das Kapitel Allgemeine Bedingungen des Digital Design Briefs definiert unter anderem Zeitplan, Budget und Ressourcen (siehe Abschnitt 2.2.2). In den folgenden Leitlinien wird auf jeden dieser Teile eingegangen.

5.1.3.1 Leitlinien für die Festlegung des Zeitplans für den Aufbau einer digitalen Lösung

In der Regel ist es nicht realistisch zu erwarten, dass während des Scoping-Schrittes genaue Zeitpläne festgelegt werden können. Ohne einen anfänglichen Zeitplan ist es jedoch schwierig, den Fortschritt zu verfolgen. Auf Foundation Level Niveau sind die folgenden Leitlinien für die Festlegung eines Zeitplans hilfreich:

- Planen Sie den konzeptuellen Schritt iterativ und mit Zeitvorgaben. Wenn das Verständnis bei allen relevanten Stakeholdern gering ist, sollten kürzere Zeiträume bevorzugt werden, um regelmäßiges Feedback zu erhalten und das Verständnis bei allen relevanten Stakeholdern zu verbessern.
- Vereinbaren Sie mit Ihrem Auftraggeber feste Termine zur Überprüfung der Ergebnisse.

- Planen Sie für den konzeptuellen Schritt maximal drei Monate ein. Längere Konzeptionsphasen erhöhen das Risiko von Fehlentwicklungen, da sich viele Detailfragen erst während der Realisierung stellen. In diesem Fall ist zu überlegen, ob die Lösung in Teile zerlegt werden kann, damit die Umsetzung schneller beginnen kann.
- Wenn Sie es mit großen Teams zu tun haben, sollten Sie Zeit für die Kommunikation und das gemeinsame Verständnis einplanen.
- Wenn die Zeit für den konzeptuellen Schritt feststeht (weil der Beginn der Entwicklung feststeht), planen Sie mindestens drei Iterationen ein: verwenden Sie 25 % der Zeit für eine erste und schnelle Iteration, 50 % für die zweite und wiederum 25 % für die letzte Iteration.
- Seien Sie darauf vorbereitet, die Lösungsidee frühzeitig zu verwerfen, wenn Sie keine Fortschritte machen oder kein Engagement von den Stakeholdern für die Lösungsidee erhalten. Gehen Sie in diesem Fall zurück zum Scoping-Schritt.
- Wenn der Liefertermin der digitalen Lösung bereits feststeht, planen Sie die Lieferung einer ersten Version der digitalen Lösung nach 50 % der Zeit. Machen Sie Ihren Auftraggebern klar, dass ein festes Datum bedeutet, dass der Umfang der digitalen Lösung ggf. angepasst werden muss, um die Frist einzuhalten.
- Analysieren Sie, wie sich Indikatoren wie Temperament, Verständnis und Engagement der Personen sowie der Reifegrad der digitalen Lösung auf den Zeitrahmen auswirken können. Planen Sie zusätzliche Zeit ein, um sich um wichtige Stakeholder zu kümmern.

5.1.3.2 Leitlinien für die Festlegung der Art der Zusammenarbeit für den Aufbau einer digitalen Lösung

Das Umsetzungsteam entwickelt die digitale Lösung nicht für sich selbst, sondern für die Auftraggeber-, die Kunden- und die Benutzer-Rolle. Daher ist die Art und Weise der Zusammenarbeit zwischen dem Umsetzungsteam und allen relevanten Stakeholdern von besonderer Bedeutung für den Erfolg eines Bauprozesses und für die Klärung der Rolle aller Beteiligten. Auf Foundation Level Niveau sind die folgenden Leitlinien nützlich, um die Art der Zusammenarbeit zu definieren:

- Planen Sie regelmäßige Treffen mit der Auftraggeber-Rolle ein, um über die Fortschritte zu informieren und um wichtige Entscheidungen bezüglich der digitalen Lösung zu treffen und zu dokumentieren. Zu Beginn sollten die Intervalle dieses Austauschs eher kürzer sein, um schnell reagieren zu können. Verlängern Sie die Sitzungsintervalle, wenn die Themen weniger dringend werden, und verkürzen Sie sie, wenn die Themen wieder dringlicher werden.
- Legen Sie die Rechte und Pflichten des Auftraggeber-Rolle explizit fest. Wichtige Rechte und Pflichten des Auftraggeber-Rolle können sein: Kurzfristige Information über kritische Themen (z. B. Umfang, Budget oder Zeitplan); Einsicht in alle Konzepte; nachvollziehbare Begründung der Entscheidungen des Umsetzungsteams.
- Legen Sie Rechte und Pflichten des Umsetzungsteams ausdrücklich fest. Wichtige Rechte und Pflichten des Umsetzungsteams können sein: Schnelle Entscheidungen der Auftraggeber-Rolle und anderer Stakeholder; klar definierte Entscheidungsspielräume bei der Gestaltung der digitalen Lösung (wann muss die Auftraggeber-Rolle konsultiert werden, wann entscheidet das Team);

- Explizite Festlegung der Rechte und Pflichten von Kunden und Benutzern im Bauprozess. Wichtige Rechte und Pflichten können sein: Entscheidungsspielraum bei der Gestaltung der digitalen Lösung (wann muss das Umsetzungsteam die Anforderungen des Kunden bzw. Benutzers umsetzen, wann kann das Umsetzungsteam selbst entscheiden); Recht auf Information über die digitale Lösung (welche Informationen dürfen Kunden bzw. Benutzer bekommen, welche nicht).
- Definieren Sie eine Eskalationsinstanz für den Fall einer Krisensituation. Tritt ein unlösbarer Konflikt zwischen verschiedenen Parteien auf, so ist eine übergeordnete Instanz erforderlich, um diesen Konflikt zu lösen und eine Entscheidung herbeizuführen. Diese Stelle kann z. B. ein Vertreter der Auftraggeber-Organisation oder eine neutrale Autorität sein. Es ist wichtig, dass sich alle Parteien über diese Befugnis einig sind und die Entscheidung respektieren.

5.1.3.3 Leitlinien für die Festlegung des Budgets für den Aufbau einer digitalen Lösung

Die Aufstellung eines Budgets ist ebenso schwierig wie die Festlegung des Zeitplans. Ohne ein definiertes Budget ist es schwierig, den Überblick über den Fortschritt und die Kosten zu behalten. Auf Foundation Level Niveau sind die folgenden Leitlinien nützlich, um ein Budget für den konzeptuellen Schritt festzulegen.

- Planen Sie explizite Budgets für das Team, das an dem konzeptuellen Schritt arbeitet.
- Wenn das Team Teilzeitkräfte umfasst (eine Situation, die in größeren Organisationen recht häufig vorkommt), dann planen Sie die Teilzeitkräfte in Timeboxes auf den Tag auf
- Planen Sie gegebenenfalls ein zusätzliches Budget für zusätzliche Fachleute ein (z. B. Softwareentwickler für das Prototyping, Interaktionsdesigner, empirische Forscher, Softwarearchitekten).
- Planen Sie gegebenenfalls zusätzliche Mittel für Anreize ein (z. B. für die Einbeziehung echter Benutzer in die Tests).
- Wenn Sie davon ausgehen, dass eine beträchtliche Anzahl relevanter Stakeholder ein negatives Engagement zeigt, sollten Sie zusätzliches Budget einplanen, um Maßnahmen durchzuführen, die dazu beitragen, die relevanten Stakeholder zu überzeugen (z. B. Budget für Studien oder Umfragen).

Planen Sie immer ein Entwicklungs- und ein Betriebsbudget gemeinsam Ihrem Auftraggeber. Auch wenn dieser Schritt für Ihren Auftraggeber schwierig ist, sind diese Budgets ein wichtiger Bezugspunkt für den Prozess, um beispielsweise zu erkennen, dass das ursprüngliche Budget nicht ausreichend ist.

5.1.3.4 Leitlinien für die Definition potenzieller Einnahmequellen für den Aufbau einer digitalen Lösung

Genau wie bei Budget und Zeitplan ist es schwierig, in dieser Phase des Bauprozesses potenzielle Einnahmequellen zu definieren. Wenn es nicht möglich ist, zumindest eine grobe Vorstellung von den Einnahmequellen zu definieren, ist die Chance, ein starkes Geschäftsmodell zu definieren, eher gering. Auf Foundation Level Niveau sollten Sie die verschiedenen digitalen Geschäftsmodelle betrachten, die in Abschnitt 4.1 vorgestellt werden. Versuchen Sie, jedes digitale Geschäftsmodell auf Ihre Lösung anzuwenden.

Wir empfehlen außerdem, die maximale Anzahl der Kunden zu schätzen, die Ihre Lösung nutzen könnten (oder ein anderes Maß für Wachstum). Eine gute Möglichkeit, sich dies

vorzustellen, ist die *unsichtbare Asymptote* [Wei2018], die eine Schätzung des maximalen Wachstums darstellt. Ausgehend von einer solchen Zahl können Sie Ihr potenzielles Einkommen rückwärts berechnen. Denken Sie daran, dass die Zahl in dieser Phase des Bauprozesses noch nicht perfekt sein muss. Weitere Einzelheiten werden im konzeptuellen Schritt ausgearbeitet.

5.1.3.5 Leitlinien für die Festlegung der verfügbaren Ressourcen für den Aufbau einer digitalen Lösung

Verfügbare Ressourcen umfassen personelle und technische Ressourcen, die für den konzeptuellen Schritt oder die Entwicklung und den Betrieb notwendig sind. Eine explizite Definition dieser Ressourcen ist notwendig, damit sie zur Verfügung stehen, wenn sie benötigt werden.

Die konkrete Art der benötigten Ressourcen hängt von der digitalen Lösung ab. Gängige Ressourcen für den konzeptuellen Schritt sind:

- Zusätzliche Fachleute für die Konzeptarbeit (z. B. Requirements Engineers, Wirtschaftsanalytiker, Interaktionsdesigner, Softwarearchitekten, Industriedesigner, Usability Ingenieure)
- Verfügbarkeit von Stakeholdern für Überprüfungen, Workshops und Evaluierungen
- Verfügbarkeit von externen Moderatoren (z. B. für Workshops)
- Geeignete Räume und Büros für die Teamarbeit
- Kompatible Infrastruktur für verteiltes Arbeiten (z. B. Dateiaustausch, Videokonferenz)
- Spezielle Software und Hardware für Konzeptarbeit und Prototypen (siehe Abschnitt 2.3, z. B. Mock-up-Software, Entwicklungsumgebung, Frameworks)
- Workshop-Materialien (z. B. Karten, Post-its, Stifte, Pins usw.)

Gemeinsame Ressourcen für den Entwicklungs- und Betriebsschritt sind:

- Personal für das Software/Hardware-Entwicklungsteam
- Software und Hardware für die Entwicklung der digitalen Lösung
- Hardware für den Betrieb der Lösung während der Entwicklung
- Testumgebungen für die Qualitätssicherung
- Hardware für den Betrieb der digitalen Lösung
- Mitarbeiter zur Unterstützung des Transformationsprozesses (falls ein Transformationsprozess erforderlich ist)
- Mitarbeiter für den Betrieb der digitalen Lösung (z. B. Support)

5.2 Leitlinien für den konzeptuellen Schritt

Der konzeptuelle Schritt ist ein vorgelagerter Schritt, der stattfindet, bevor die Entwicklung der digitalen Lösung beginnt. Das Ziel des konzeptuellen Schritts ist es, ein ausreichendes Verständnis für die geplante digitale Lösung zu erlangen, bevor das Risiko eingegangen wird, mit der Entwicklung zu beginnen. Für dieses Ziel werden zwei Konzepte erstellt

- Das initiale Lösungsdesignkonzept
- Das initiale Designkonzept auf Systemebene

Diese Konzepte werden als initial bezeichnet, da sie während des Entwicklungs- und Betriebsschritts weiter verfeinert und überarbeitet werden (siehe Abschnitt 5.3).

Beim konzeptuellen Schritt geht es um ein gemeinsames Verständnis

Ein typisches Missverständnis der Konzeptarbeit ist, dass sie auf die Erstellung von Dokumenten ausgerichtet ist. Dieses Missverständnis hat seinen Ursprung in der Dokumentenstruktur von Konzepten (siehe Abschnitt 2.2.2) und der Annahme, dass die Dokumente wie eine Art Fragebogen oder Checkliste sequentiell abgearbeitet werden müssen. Ganz im Gegenteil: Die Konzeptarbeit zielt auf ein gemeinsames Verständnis ab und ist ein hochgradig iterativer Prozess, der eine analytische, kreative und heuristische Denkweise beinhaltet.

Menschenzentrierter Designprozess als Arbeitsweise in einem langfristigen Workshop

Ein allgemeines Prozessmodell für Einsteiger im konzeptuellen Schritt ist der menschenzentrierte Designprozess [ISO2019]. Er besteht aus vier gleichrangigen, iterativen Aktivitäten: Verstehen, Definieren, Gestalten und Evaluieren.

Eine gute Arbeitsweise für Einsteiger im konzeptuellen Schritt ist ein Workshop von einer oder mehreren Wochen mit dem Product Owner, dem Umsetzungsteam und anderen Fachleuten, an dem bei Bedarf auch die Auftraggeber-Rolle, potenzielle Kunden-, potenzielle Benutzer-Rollen und andere relevante Stakeholder teilnehmen.

Ein ständiger Workshopraum als Zentrale für das Team

Wir empfehlen die Einrichtung eines festen Workshop-Raums, in dem konzeptuell gearbeitet werden kann (vgl. z. B. [GBGSV2013]). Die Wände des Raumes können zur Visualisierung der verschiedenen Ergebnisse der Konzeptarbeit genutzt werden. Eine Wand kann für das Lösungsdesignkonzept, eine Wand für das Systemdesignkonzept und eine Wand für die Konstruktionsperspektive verwendet werden. Die verbleibende vierte Wand kann zum Sammeln von Material, zur Verwaltung der Arbeit oder für alternative Perspektiven (z. B. Change Management) genutzt werden. Ein alternativer Ansatz besteht darin, drei Wände des Raums alternativen Lösungsideen und eine Wand dem Arbeitsmanagement zu widmen.

Bei einem solchen Ansatz wird der Raum selbst zu einem Medium, das für die kreative Arbeit und auch für die Präsentation der entwickelten Ideen gegenüber der Auftraggeber-Rolle und anderen relevanten Stakeholdern genutzt werden kann. Im Folgenden beschreiben wir vier Phasen als Arbeitsstruktur für den konzeptuellen Schritt.

5.2.1 Phase 1: Erkundung des Lösungsraums für die digitale Lösung aus der Kundenperspektive

Einsteiger in das Digitale Design sollten immer mit einer Erkundung des Lösungsraums als Phase 1 beginnen. Das Wertversprechen, die Customer Journey und die Customer Personas der digitalen Lösung (siehe Abschnitt 2.2.3) sind die Arbeitsprodukte, die wir in dieser Phase ausarbeiten. In einem festen Workshop-Raum können alle drei Arbeitsprodukte (Wertversprechen, Customer Journey Map und Personas) mit Post-its und Klebeband an der Wand befestigt werden.

Die Vision und die Situationsbeschreibung aus dem Digital Design Brief sind der Ausgangspunkt für die Arbeit in dieser Phase. Die folgende Abfolge von fünf Aktivitäten sollte durchlaufen werden.

Aktivität 1: Einfühlungsvermögen entwickeln und die Situation der potenziellen Kunden und Benutzer verstehen

Als Ausgangspunkt sollte das Team reale Personen identifizieren, die potenzielle Benutzer werden können oder bereits Kunden und/oder Benutzer sind. Das Team sollte diese Personen befragen, um ein umfassendes Verständnis ihrer Situation, ihrer Aufgaben, Bedürfnisse und Wünsche zu erhalten. Neben persönlichen Gesprächen ist die Feldforschung hier eine wichtige Informationsquelle. Das Team kann z. B. Menschen bei ihrer täglichen Arbeit beobachten oder versuchen, die Aufgaben der betroffenen Personen zu übernehmen (z. B. die Arbeit selbst zu erledigen).

Aktivität 2: Definieren Sie die wichtigsten Erkenntnisse über die Aufgaben, Probleme und Nutzen der Kunden aus heutiger Sicht

Wenn das Team ein ausreichendes Verständnis der potenziellen Kunden und Benutzer gesammelt hat, sollte es die Ergebnisse sammeln und die zentralen Erkenntnisse aus Aktivität 1 definieren. Für diese Aktivität kann das Team Persona-Vorlagen, Stakeholderlisten und die Kundenprofile aus dem Value Proposition Canvas und der Customer Journey Map verwenden (siehe Abschnitt 2.2.3).

Beginnen Sie die Arbeit im Team mit der Definition eines ersten Sets von Kunden-/Benutzerprofilen und erarbeiten Sie eine Persona-Vorlage für jedes Kundenprofil. Priorisieren Sie die definierten Kunden-Personas, um ihre Bedeutung für die Lösung zu verstehen.

Sobald die Personas definiert und priorisiert sind, arbeiten Sie an dem Kundenprofil des Value Proposition Canvas (2.2.3.4):

- Legen Sie im Feld Kundenaufgabe(n) eine Notiz für jeden Haupt- und Nebenauftrag an, bei dessen Erledigung Sie Ihrem Kunden helfen wollen.
- Vermerken Sie in im Feld Problem jedes Problem, das Ihr Kunde vor, während und nach der Erledigung des Auftrags erlebt oder erleben könnte.
- Legen Sie für jeden Nutzen, den Ihr Kunde erwartet, sich wünscht, oder der ihn überraschen würde, eine Notiz in der Nutzen-Box an.
- Erstellen Sie eine Notiz auf der Customer Journey Map für bestehende Aktivitäten und Kontaktpunkte.

Versuchen Sie im Rahmen dieser Aktivität, die definierten Personas, die Value Map und die bestehenden Customer Journeys zu integrieren, um ein einheitliches Verständnis der aktuellen Situation des Kunden zu erhalten:

- Bewerten Sie die aktuellen Erfahrungen in der Customer Journey Map: Wie erlebt der Kunde den Touchpoint (Berührungspunkt) heute? Werden die Erwartungen nicht erfüllt, werden sie erfüllt, oder passiert etwas, was der Kunde nicht erwartet hat?
- Identifizieren Sie Trustpoints und relevante Bedürfnisse: Welche Relevanz hat ein Touchpoint in der jeweiligen Situation?
- Übertragen Sie relevante Bedürfnisse und Probleme in ein erstes Wertversprechen: Was sind die wichtigsten Bedürfnisse und Probleme im entscheidenden Moment?

Es ist völlig normal, zwischen der Arbeit an den Persona-Vorlagen, den Kundenprofilen und den Customer Journey Maps zu wechseln, da die Arbeit an einem Arbeitsprodukt das Verständnis des Teams für die anderen verbessert.

Aktivität 3: Definieren und Ausarbeiten des Wertversprechens zusammen mit der Customer Journey aus einer Zukunftsperspektive

Sobald das Team anhand von Personas und Kundenprofilen einen guten Konsens über die aktuelle Situation des Kunden erreicht hat, kann es sich an die Zukunftsperspektive der digitalen Lösung machen. Diese Zukunftsperspektive sollte im Digital Design nicht unterschätzt werden, da das Potenzial innovativer digitaler Technologien nur aus der Zukunftsperspektive bewertet werden kann, da sie heute oft noch nicht vorhanden sind.

Das Team sollte damit beginnen, potenzielle Trends für künftige Erfahrungen und künftige Bedürfnisse zu erkunden. So kann beispielsweise die Verfügbarkeit einer zuverlässigen Sprachinteraktionstechnologie eine andere Interaktions- und Serviceerfahrung in einer digitalen Lösung schaffen. Ein weiterer Ansatz ist die Übertragung bestehender Dienste (z. B. Echtzeit-Paketverfolgung) in den Bereich der digitalen Lösung.

Die ermittelten Bedürfnisse können in die bestehende Value Map integriert werden. Ein pragmatischer Ansatz besteht darin, den Bedarf mit andersfarbigen Notizen zu dokumentieren, um anzuzeigen, dass es sich um zukünftige Bedürfnisse handelt. In einem zweiten Schritt kann das Team die in der Customer Journey Map identifizierten zukünftigen Bedürfnisse bewerten, um schließlich Probleme und Möglichkeiten zu identifizieren, die den Lösungsraum erweitern. Das Ergebnis dieses Arbeitsschrittes sind konkrete Fragen für die Zukunft.

Aktion 4: Erkunden der Lösungswege und entwickeln von Lösungsideen

Mit diesem verbesserten Verständnis der Probleme und Möglichkeiten im Lösungsraum kann das Team beginnen, alternative Lösungsrichtungen zu erkunden und Lösungsideen zu den Zukunftsfragen zu entwickeln.

Beginnen Sie die Arbeit mit der Value Map (siehe 2.2.3.4):

- Führen Sie alle Produkt- und Dienstleistungsideen auf, auf denen Ihr Wertversprechen beruht, indem Sie für jedes Element eine Notiz im Feld Produkte und Dienstleistungen der Value Map erstellen.
- Beschreiben Sie, wie Ihre digitale Lösung innerhalb einer Customer Journey Map einen Mehrwert schafft, indem sie entweder Probleme des Kunden beseitigt oder Nutzen für den Kunden schafft.

Alternative Lösungsideen können innerhalb einer einzigen Value Map oder mit mehreren Value Maps definiert werden. Dabei ist es wichtig zu erkennen, dass eine in der Value Map definierte Produkt-/Dienstleistungsidee nicht unbedingt die gesamte digitale Lösung definiert. Eine digitale Lösung besteht aus mehreren Produkt-/Dienstleistungsideen.

Wenn das Team ein gutes Verständnis für die Value Map hat, kann es mit der Arbeit an der Customer Journey Map beginnen. Erarbeiten Sie Customer Journey Maps, die die Produkt-/Dienstleistungsideen in die reale Situation potenzieller Kunden versetzen. Definieren Sie Customer Journey Maps für jede Persona, um ihre persönliche Reise und Erfahrung mit Ihrer Lösungsidee zu erkunden. Denken Sie daran, dass die Customer Journey Map ein sehr detailliertes Arbeitsprodukt ist, mit dem Sie sehr konkrete Situationen und Erfahrungen des Kunden beschreiben können. Berücksichtigen Sie bei der Definition der Customer Journey auch die verschiedenen Technologien, die Sie im Digital Design Brief definiert haben.

Ähnlich wie bei Aktivität 2 sind die Arbeiten an der Value Map und den Customer Journey Maps eng miteinander verbunden. Sie werden also auf jeden Fall zwischen den beiden Arbeitsprodukten hin und her wechseln.

Aktivität 5: Leichtgewichtige Evaluierung von Lösungsideen

Sobald das Team ein gutes Verständnis der alternativen Lösungsideen erreicht hat, sollte das Team seine Ideen bewerten. Da wir uns hier in einem frühen Stadium des Prozesses befinden, empfehlen wir einen leichtgewichtigen Evaluierungsansatz.

Das Team kann die Auftraggebenden, potenzielle Kunden und Benutzende sowie weitere Stakeholder in einen dauerhaften Workshop-Raum einladen. Dort kann das Team die Lösungsideen vorstellen und erhält sofortiges Feedback. Bei dieser Form der frühen Evaluierung ist es wichtig, dass das Team eine entspannte und offene Workshop-Atmosphäre schafft, damit offenes Feedback gegeben werden kann.

Führen Sie die fünf Aktivitäten mindestens dreimal durch, bevor Sie sich an eine detaillierte Evaluierung machen

Diese fünf Aktivitäten sollten mindestens dreimal durchgeführt werden, um das Verständnis und die Wertversprechen, Customer Journeys und definierten Personas weiter zu verbessern. Am Ende dieses Prozesses sollte ein klares Verständnis der Wertversprechen erreicht sein. Ist dies nicht der Fall, sollte der Vorgang wiederholt werden. Ein klares Verständnis des Wertversprechens ist erreicht, wenn das Team ein klares Wertversprechen präsentieren kann, das für Außenstehende leicht zu verstehen ist.

Tabelle 23 - Struktur einer Pitch-Präsentation mit Beispielinhalten von YPRC

Thema	Beispiel aus dem YPRC
Wer ist der Kunde, der von der digitalen Lösung profitieren wird?	YPRC richtet sich an Einsteiger im Langstreckenlauf.
Welchen Wert erwarten die Kunden von der Lösung?	Unsere Kunden erwarten ein professionelles Coaching für einen guten Start in den Langstreckenlauf.
Was ist der Hauptgrund für den Kunden, sich für unsere Lösung zu entscheiden?	YPRC bietet ein erschwingliches und individuelles persönliches Coaching-Erlebnis.
Worum geht es bei der digitalen Lösung und welches Wertversprechen bietet sie?	YPRC bietet einen Remote-Coaching-Service, der den Läufer über eine digitale Sprachverbindung mit einem Remote-Lauftrainer verbindet.
Wie erreicht die Lösung das Wertversprechen für den Kunden?	Mithilfe einer Smartwatch und einer Smartphone-App kann der Lauftrainer die Leistung und die Gesundheitsdaten des Läufers überwachen. Anhand dieser Daten kann der Coach über eine Sprachverbindung Ratschläge erteilen.

Thema	Beispiel aus dem YPRC
Wie erlebt ein Kunde unsere digitale Lösung?	Unsere Kunden schätzen die Individualität unseres Coaching-Angebots. Sie sind nicht auf feste Laufgruppen angewiesen. Stattdessen können sie einen individuellen Coaching-Termin buchen, wann es ihnen am besten passt.
Wie erlebt die Auftraggeber-Rolle unsere digitale Lösung?	Die Firma YPRC ist stolz darauf, die erste Firma auf dem Markt zu sein, die eine solche erschwingliche und individuelle Coaching-Lösung anbieten kann.
Wie sieht der Fahrplan für die Realisierung der digitalen Lösung aus?	Die erste Version von YPRC wird eine White-Label-Smartwatch zur Erfassung von Gesundheitsdaten verwenden. Die Entwicklung der Smartphone-App und des Coaching-Portals wird etwa 9 Monate dauern. Wir gehen davon aus, dass wir den Dienst innerhalb von 12 Monaten wohlgesonnenen Kunden anbieten können.
Was ist die endgültige Vision für die digitale Lösung?	Der YPRC soll zu einer marktführenden Lösung für Coaching-Dienste im Laufsport werden. Die Vision ist, zusätzlich zu den persönlichen Coaches einen digitalen Voice Coach anzubieten, der künstliche Intelligenz nutzt.

Nach drei Iterationen sollten die zwei oder drei besten Lösungsideen in einer Pitch-Präsentation für die Auftraggeber-Rolle und in einem fiktiven Videoprototyp (siehe Abschnitt 2.3) für eine genauere Evaluierung vorbereitet werden.

Die Pitch-Präsentation ist ein Format, das in der Startup-Community weit verbreitet ist (siehe z. B. [Ries2011]). Es handelt sich um eine kurze Präsentation der gesamten Idee, die hinter einem Startup steckt, und dient dazu, Investoren von der Unterstützung des Startups zu überzeugen. Im Rahmen des Bauprozesses für eine digitale Lösung empfehlen wir, die Pitch-Präsentation als Instrument für eine kurze und knackige Präsentation der digitalen Lösung zu nutzen. Die Präsentation sollte maximal 10 Minuten lang sein und die in Tabelle 23 aufgeführten Themen behandeln.

Zusätzlich zum Pitch sollte den verschiedenen potenziellen Kunden/Benutzern und weiteren Stakeholdern ein Werbevideo präsentiert werden, um deren Feedback einzuholen. Der konkrete Inhalt des Videos sollte zusammen mit den Bewertungszielen kurz beschrieben werden, um ein einfaches Lösungsevaluationskonzept zu erstellen. Darüber hinaus sollte das Team eine kurze Liste von Fragen definieren, welche die Stakeholder nach dem Ansehen des Videos beantworten sollten, um ein strukturiertes Feedback zu erhalten. Wenn die Lösungsideen akzeptiert werden, kann die nächste Phase in Angriff genommen werden. Wenn nicht, beginnt Phase 1 erneut.

5.2.2 Phase 2: Erarbeitung und Bewertung von Lösungskandidaten aus wirtschaftlicher Sicht

In der zweiten Phase empfehlen wir, sich auf das Geschäftsmodell zu konzentrieren, um einen angemessenen Detaillierungsgrad zu erhalten. Das Business Model Canvas (siehe Abschnitt 2.2.3) und ein System Design Canvas dienen als Leitfaden für diese Tätigkeit.

Erstellung eines separaten Business Model Canvases für jede Lösungsidee

Wir empfehlen, für jede Lösungsidee ein eigenes Business Model Canvas zu erstellen. In einem festen Workshop-Raum kann das Business Model Canvas mit Post-its und Klebeband an der Wand angebracht werden.

Das Business Model Canvas (siehe Abbildung 18 und Abbildung 58) ist ein wichtiges Instrument, um ein erstes und tiefes Verständnis für die Lösungsideen zu bekommen. Das Team beschreibt alle Sichten auf das Canvas: Kunden, Technologie und Umwelt. Das Team sammelt Informationen, befragt Experten, untersucht potenzielle Kunden und analysiert Hindernisse oder Potenziale des Geschäftsmodells sowie der Entwicklungsorganisation.

Es gibt keine bestimmte Reihenfolge, die für das Canvas einzuhalten ist. Ein guter Ausgangspunkt ist die Kundenperspektive, da das Unternehmen auf die Kunden ausgerichtet ist. Eine gute Reihenfolge ist:

- Füllen Sie die Kundensegmente und die Elemente des Wertversprechens mit den Ergebnissen aus Phase 1 (Wertversprechen).
- Legen Sie einen klaren Namen für die Geschäftsidee fest, um auch die Perspektive der Auftraggeber-Rolle anzusprechen. Der Name sollte den Handlungsbedarf widerspiegeln, der durch die Motivation für die digitale Lösung als Teil des Lösungsdesignkonzepts definiert wird.

Die anderen Elemente müssen vom Team ausgearbeitet werden. Falls erforderlich, sollten externe Fachleute hinzugezogen werden:

- Beschreiben Sie (Verkaufs-, Liefer-, Service-) Kanäle und Kundenbeziehungen
- Bewerten Sie die Einnahmequellen
- Beschreiben Sie Schlüsselressourcen, Schlüsselaktivitäten und Schlüsselpartnerschaften
- Bewerten Sie die Kostenstruktur

Iterieren Sie die Elemente des Business Model Canvas mehrere Male

Ein gutes Business Model Canvas wird durch mehrmaliges Wiederholen der verschiedenen Elemente erstellt. Sehr oft wird dabei der gesamte Inhalt des Canvas überarbeitet. Während diesen Iterationen sollte das gesamte Wertversprechen des Geschäftsmodells (die *Story* für die Kunden-Rolle) immer wieder hinterfragt werden. Das Geschäftsmodell kann der Kunden-Rolle nur dann einen guten Wert bieten, wenn die *Story* für die Kunden-Rolle klar ist.

Abbildung 58 fasst eine gute Arbeitsreihenfolge für die Elemente des Business Model Canvas und die Beziehungen zwischen ihnen zusammen. Bestimmte Arten von digitalen Lösungen (z. B. unternehmensinterne Systeme) erfordern kein vollständig ausgearbeitetes Business Model Canvas. Dennoch empfehlen wir, beim Canvas Template zu bleiben und die Teile so gut wie möglich zu füllen.

Die erstellten Business Model Canvases müssen gemeinsam mit dem Auftraggeber bewertet werden. Wie bei der Evaluierung des Value Proposition Canvas empfehlen wir, im Lösungsevaluationskonzept klare Bewertungsziele zu definieren und zu dokumentieren (z. B. Bewertung des Preises für eine bestimmte Leistung). Das Feedback aus der Evaluierung sollte in das Business Model Canvas einfließen.

Erstellung der Business Model Canvases in drei Runden mit verschiedenen Stakeholdern

Wir empfehlen, die Business Model Canvases in drei Runden zu erstellen. Die erste Runde sollte schnell gehen und nur eine grobe Skizze sein, die mit dem Auftraggeber und den relevanten Stakeholdern diskutiert werden kann. Die zweite und dritte Runde können ausführlicher sein. Während dieses Prozesses ist es wichtig, den Handlungsbedarf (siehe

2.2.2.1) der Kunden-/Auftraggeber-Rolle kontinuierlich zu erörtern und zu beobachten, ob sich der Handlungsbedarf ändert. Unserer Erfahrung nach ändert und konkretisiert sich der Handlungsbedarf der Auftraggeber-/Kunden-Rolle während der frühen Konzeptarbeit.

Wir empfehlen außerdem, das Problemverständnis (siehe Abbildung 56), den Handlungsbedarf und das Engagement der Stakeholder (siehe Abbildung 57) im Auge zu behalten. Eine Verbesserung des Problemverständnisses und des Engagements ist ein gutes Zeichen dafür, dass sich die Lösungsidee in eine gute Richtung entwickelt. Wird keine Verbesserung festgestellt, sollte das Umsetzungsteam versuchen, die Gründe dafür zu verstehen und die Erkenntnisse aus dieser Analyse in die weitere Entwicklung der digitalen Lösung einfließen zu lassen.

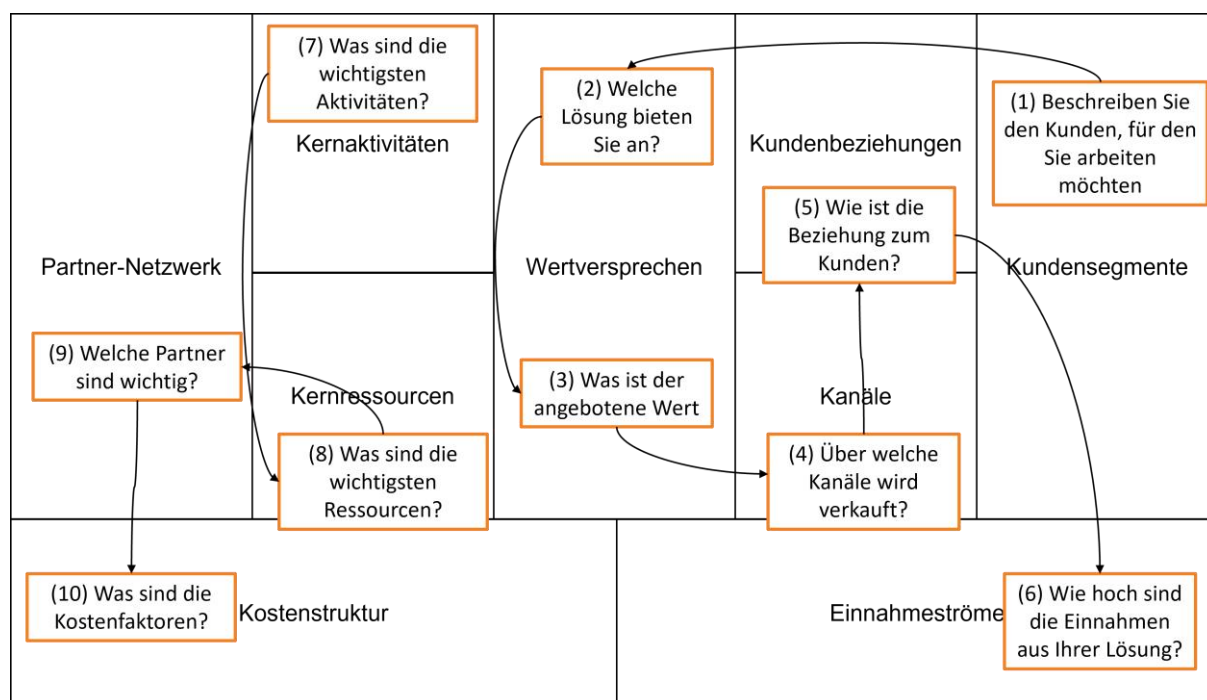


Abbildung 58 - Beziehungen, die die Entwicklung eines guten Business Model Canvas unterstützen

Ausarbeitung von Lösungskandidaten mit einem Systemdesign-Canvas

Mit den Informationen aus den Business Model Canvases kann das Team beginnen, ein erstes Designkonzept auf Systemebene für jede Idee auszuarbeiten.

In dieser Phase des Prozesses sind alternative Lösungsideen erwünscht und wertvoll. Daher sollte das Team alternative und leichtgewichtige Designkonzepte auf Systemebene ausarbeiten.

In einem permanenten Workshop-Raum kann die Vorlage für das Systemdesignkonzept (siehe Abschnitt 2.2.2, Tabelle 5) als Canvasstruktur anstelle einer Dokumentenstruktur verwendet werden. Wir empfehlen, mit der Beschreibung des Zwecks (Ziele) und der Funktion des Systems (Szenarien) zu beginnen. Auf der Grundlage der Szenarien lassen sich weitere Elemente des Systemdesignkonzepts ableiten. Die folgenden Leitlinien unterstützen die Erstellung eines Systemdesignkonzepts aus dem Lösungsdesignkonzept:

- Denken Sie an Benutzertypen, die das Wertversprechen befürworten werden. Benutzertypen können aus den Kundensegmenten abgeleitet werden, wenn der Kunde auch ein Benutzer ist. Beginnen Sie mit einem Benutzertyp für Ihre Kunden. Denken Sie daran, dass es weitere Benutzertypen geben kann, die für den Erfolg des Geschäfts wichtig sind (z. B. der Coach von YPRC). Andere mögliche Benutzertypen sind Kundenbetreuung oder Verwaltungspersonal.
- Vorhandene Objekte hängen von der Lösungsidee ab. Typische Beispiele für vorhandene Objekte sind Geräte, die der Kunde zur Ausführung einer digitalen Lösung verwendet (z. B. ein Smartphone, ein Tablet oder ein PC).
- Bestehende Systeme hängen von der Lösungsidee und dem Geschäftsmodell ab. Stellen Sie sich die Frage, welche Dienstleistungen für die Umsetzung der Lösung erforderlich sind? Typische Beispiele für bestehende Dienste sind Zahlungsanbieter und Systeme, die zusätzliche Dienste oder Daten (z. B. Kartendaten) bereitstellen.
- Die Definition von Elementen der digitalen Lösung ist schwierig. In einer vernetzten Welt bestehen die meisten digitalen Lösungen aus mindestens zwei Elementen: einem Element für den direkten Benutzer und einem Element, das den Server oder das Backend der digitalen Lösung darstellt.
- Qualitätsanforderungen und Randbedingungen können zu jedem Zeitpunkt des Prozesses auftreten. Wann immer ein Element der Lösung definiert wird, fragen Sie sich: Was sind wichtige Eigenschaften des neuen Elements? Was sind die Randbedingungen, die für das neue Element gelten?

Leitlinien für die Suche nach dem richtigen Detaillierungsgrad für das Systemdesign eines Lösungskandidaten

Ein typischer Einsteigerfehler bei der Erstellung eines Systemdesignkonzepts ist es, die Dinge zu kompliziert zu machen und zu versuchen, alles selbst zu bauen. Die folgenden Leitlinien helfen Ihnen dabei, den richtigen Detaillierungsgrad für das Systemdesign zu finden:

- Konzentrieren Sie sich auf eine einfache Lösung, die den Kernprozess der digitalen Lösung unterstützt, und definieren Sie die Details in den Elementdesignkonzepten. Denken Sie daran, dass es beim Systemdesignkonzept darum geht, die Gesamtstruktur der digitalen Lösung zu definieren.
- Die administrativen Aspekte der Lösung (Registrierung, Login, Logout) sind Sache der Elementdesignkonzepte. Besondere Anforderungen an die Sicherheit können in Form von kurzen textuellen Qualitätsanforderungen dokumentiert werden. Zum Beispiel ist das Schreiben eines Szenarios, das eine Zwei-Faktor-Authentifizierung des Benutzers im Detail beschreibt, eine Zeitverschwendung im Systemdesignkonzept. Stattdessen reicht eine kurze Qualitätsanforderung aus (z. B.: *Die App muss eine Zwei-Faktor-Authentifizierung für die Anmeldung bieten*).
- Vermeiden Sie die Definition von Ausnahmen und alternativen Verhaltensweisen auf der Systemebene. Beschreiben Sie Szenarien, die die allgemeine Funktion der Lösung veranschaulichen (manchmal auch als *Happy Path* bezeichnet), und überlassen Sie die Details der Elementebene (und den Use Cases).

In dieser Phase werden zusätzliche Beiträge von externen Fachleuten erforderlich sein. Zusätzliche Interviews und Workshops mit potenziellen Benutzern können ebenfalls erforderlich sein.

Um die verschiedenen Konzepte auf Systemebene mit den Benutzern zu bewerten, sollte das Team einfache interaktive Mock-ups als Prototypen der Lösung erstellen. Für diesen Zweck gibt es spezielle Werkzeuge, mit denen sich solche interaktiven Mock-ups schnell und einfach erstellen lassen. Verwenden Sie die Szenariobeschreibung als Drehbuch für den Prototyp und bereiten Sie die Mock-ups so vor, dass sie die Interaktion mit der digitalen Lösung während des Szenarios veranschaulichen. Außerdem sollte das Team Interviewfragen vorbereiten, die die Benutzer nach der Verwendung des Prototyps beantworten sollten. Alle diese Aspekte sollten im Konzept der Systemevaluation dokumentiert werden.

Mit diesem Ansatz können Sie auch „Wizard of Oz“-Prototypen erstellen, die die digitale Lösung für potenzielle Benutzer und weitere Stakeholder simulieren. Diese Prototypen ermöglichen es Ihnen, Rückmeldungen von den potenziellen Benutzern und weiteren Stakeholdern einzuholen und erste Probleme mit der Benutzbarkeit zu erkennen (siehe Abschnitt 4.1.4).

5.2.3 Phase 3: Annäherung an einen vielversprechenden Lösungskandidaten aus der Perspektive der Machbarkeit

Mit den Erkenntnissen aus den interaktiven Mock-ups kann in Phase 3 ein größerer Iterationszyklus beginnen, um die Machbarkeit der Lösungskandidaten zu prüfen. Das Feedback kann zu Änderungen des Lösungsdesignkonzepts (Wertversprechen, Customer Journey und Geschäftsmodell) oder des Systemdesignkonzepts führen.

Das Team sollte das Lösungsdesignkonzept und das Systemdesignkonzept iterieren und anhand von Pitch-Präsentationen, einfachen Videoprototypen oder interaktiven Mock-ups erneut bewerten. Sie können sogar beide Elemente kombinieren: Die interaktiven Mock-ups können zur Darstellung des digitalen Systems im Video verwendet werden. Das primäre Ziel ist es, ein tiefes Verständnis der Machbarkeit eines Lösungskandidaten in Bezug auf die Technologie (ist sie realisierbar?) und die Betriebsfähigkeit (können wir die Lösung betreiben?) zu erlangen.

Die Intensität der Evaluationsaktivitäten sollte entsprechend dem Grad des Problemverständnisses festgelegt werden. Als Faustregel gilt, dass ein geringeres Maß an Problemverständnis konkretere und realistischere Prototypen erfordert, um das Verständnis der relevanten Stakeholder zu verbessern. Zusätzlich zu den Prototypen sollten gute Storys vorbereitet werden, die die Anwendung des Prototyps in seinem späteren Kontext erklären. Ein weiterer Ansatz zur Bewertung der Gesamtlösungsidee ist der Lean-Startup-Ansatz (siehe Abschnitt 5.4).

Nach einer gewissen Anzahl von Iterationen sollte sich ein vielversprechendes Gesamtkonzept für die digitale Lösung herauskristallisieren. Wenn die Iterationen nicht zu einer stabilen Idee führen, sollte das Team erwägen, zum Scoping-Schritt zurückzukehren.

5.2.4 Phase 4: Abschließende Evaluierung des Lösungskandidaten mit dem Auftraggebenden

Sobald eine stabile Idee für die digitale Lösung und das zugrundeliegende System definiert ist, sollte das Team das Lösungsdesign- und das Systemdesignkonzept im Detail gemäß den definierten Vorlagen ausarbeiten (siehe Abschnitt 2.2.2). An dieser Stelle beginnt die eigentliche Arbeit an den Dokumenten.

Während der Ausarbeitung des Systemdesignkonzepts können sich wichtige Fragen im Zusammenhang mit der Machbarkeit von Details stellen. Einfache funktionale Prototypen sollten von Realisierungsexperten erstellt werden, um die Machbarkeit dieser Details zu bewerten.

Beispiel YPRC. In der YPRC-Fallstudie gibt es zwei gute Beispiele für solche Machbarkeitsfragen: Die Machbarkeit der Remote-Coaching-Funktion und die Machbarkeit des Coachings mit künstlicher Intelligenz (bitte lesen Sie die Fallstudie für weitere Einzelheiten). Solche frühen Prototypen sollten sich im Allgemeinen auf die technische Machbarkeit konzentrieren. Der Benutzer steht in dieser Phase nicht im Mittelpunkt.

Sobald das detaillierte Lösungsdesign- und Systemdesignkonzepte vorliegen, ist eine abschließende Evaluierung erforderlich:

- Besprechen Sie die Konzepte mit Ihrem Auftraggeber und den relevanten Stakeholdern.
- Verwenden Sie interaktive Mock-ups und/oder ein Werbevideo für potenzielle Benutzer/Kunden.

Die konkreten Verfahren (z. B. Review-Checkliste, Feedback-Fragen, verwendete Prototypen) und Review-Ergebnisse sollten im Lösungs- bzw. Systemevaluationskonzept dokumentiert werden. Die Dokumentation dieser Informationen ist aus Gründen der Rückverfolgbarkeit und zur Dokumentation des gemeinsamen Verständnisses und des Feedbacks aller Stakeholder besonders wichtig. Die Dokumentation dieser Details kann als Referenzpunkt während des Entwicklungsschritts besonders nützlich sein.

Neben der Evaluierung der Details der digitalen Lösung ist es auch notwendig, das Problemverständnis und das Engagement der relevanten Stakeholder zu bewerten. Die Evaluierung kann ein Nebenprodukt der Evaluierung der Details der digitalen Lösung sein. In dieser Phase des Bauprozesses ist es von großer Bedeutung, dass alle Stakeholder die Lösung zumindest im Allgemeinen verstanden haben. Außerdem sollte kein relevanter Stakeholder eine blockierende Haltung haben (C--, siehe Abbildung 57).

Wenn die Ergebnisse der Evaluierung (Phase 4) positiv sind, kann der Entwicklungs- und Betriebsschritt beginnen. Andernfalls werden weitere Iterationen empfohlen, vor allem, wenn es noch Stakeholder mit einem negativen Engagement gibt.

5.3 Leitlinien für den Entwicklungs- und Betriebsschritt

Das Ziel des Entwicklungs- und Betriebsschritts ist es, die digitale Lösung zum Leben zu erwecken und sie während des Betriebs zu warten und weiterzuentwickeln. Die wichtigsten Ergebnisse dieses Schrittes aus der Sicht der Gestaltung sind:

- Die digitale Lösung im Gebrauch
- Die Designkonzepte auf Elementebene, die jedes implementierte Element beschreiben (siehe Abschnitt 2.2.2)
- Ein überarbeiteter Design Brief, eine Lösungsdesign- und Systemdesignkonzept, die die detaillierten Entscheidungen aus der Entwicklungs- und Betriebsphase widerspiegeln

Es gibt viele Prozessmodelle für die Entwicklung einer digitalen Lösung

Die industrielle Praxis hat verschiedene Ansätze, Methoden und Prozessmodelle für die Entwicklung einer digitalen Lösung entwickelt. Die Fachleute kennen das breite Spektrum der Ansätze (z. B. V-Modell, Scaled Agile Framework, Rational Unified Process und Lean Startup). Für Einsteiger im Bereich des Digital Designs ist diese breite Palette oft überwältigend. Die wichtige Botschaft für Einsteiger ist, dass es keinen fertigen Ansatz für die Entwicklung einer digitalen Lösung gibt. Alle Ansätze stellen eine Art Rahmen dar, der auf die jeweilige Situation zugeschnitten werden muss.

Ein von Kanban und Scrum inspirierter Prozess für Einsteiger

Der in diesem Abschnitt vorgestellte Prozess wurde von Kanban und Scrum inspiriert und verwendet Elemente aus beiden Quellen, die sich unserer Meinung nach besonders gut für den Einstieg in den Bauprozess für digitale Lösungen eignen. Diese Elemente wurden so angepasst, dass sie bestens zum Digital Design passen. Der Prozess ist also weder Kanban noch Scrum in seiner reinsten Form.

Kanban ist ein Ansatz zur Steuerung von Softwareentwicklungsprozessen [Ande2010]. Scrum [ScSu2020] ist ein Framework zum Entwickeln und zur Wartung komplexer Produkte. Beide sind anerkannt und weit verbreitet. Sie haben darüber hinaus große Communities, die Einsteigern verschiedene Ressourcen für die Anpassung eines Entwicklungsprozesses zur Verfügung stellen. Ein weiterer praktischer Aspekt der Verwendung von Kanban und Scrum als Grundlage ist, dass sie von verschiedenen Software-Werkzeuge gut unterstützt werden.

Erwartungsmanagement zum vorgestellten Prozess

Bevor wir auf die Einzelheiten eingehen, möchten wir die Erwartungen von Lesern und Leserinnen ein wenig steuern. Der Entwicklungs- und Betriebsschritt des Bauprozesses ist bei weitem der komplizierteste und schwierigste Schritt. Im folgenden Abschnitt werden die verschiedenen Aspekte dieses Schrittes im Detail erläutert.

Das Ziel des DDP Foundation Level ist es, die grundlegenden Mechanismen dieses Schrittes und seine Beziehungen zum Digital Design zu vermitteln. Dieses Verständnis ist die Grundlage für das Erlernen des Umgangs mit einem solchen Prozess. Die zusätzlichen Erläuterungen sollen ein vollständiges Bild des Bauprozesses vermitteln.

Ein Leser oder eine Leserin ohne Erfahrung mit dem Aufbau einer digitalen Lösung muss eine erhebliche Lernkurve durchlaufen, um den gesamten Prozess zu verstehen. Zusätzlich zu der folgenden Einführung enthält das Material des DDP Foundation Level eine vollständige Fallstudie, die den vorgestellten Prozess in der Praxis zeigt. Wir empfehlen dringend, die Fallstudie eingehend zu studieren und sie als Referenzquelle für Ihre eigene Arbeit zu verwenden.

5.3.1 Überblick über die Struktur des Prozesses

Der Einfachheit halber gehen wir davon aus, dass die Elemente der digitalen Lösung von einem einzigen Team realisiert werden können. Die Entwicklung größerer digitaler Lösungen kann von den in diesem Handbuch vorgestellten Grundlagen aus skaliert werden. Dies erfordert jedoch zusätzliche Managementfähigkeiten, die über den Rahmen dieses Foundation Levels hinausgehen.

Verwaltung der Arbeit auf allen Abstraktionsebenen

In Anlehnung an die in Abschnitt 1.2 vorgestellten Abstraktionsebenen verwalten wir die Arbeit während des Entwicklungs- und Betriebsschritts auf drei Ebenen:

- Der Prozess auf *Lösungsebene* konzentriert sich auf die Auftraggeber-Rolle und zielt darauf ab, die Ziele des Auftraggebenden durch die digitale Lösung zu erreichen. In diesem Prozess geht es um die Kommunikation und Koordination zwischen den Personen, die die Rollen des Auftraggebers und des Product Owners einnehmen.
- Der Prozess auf *Systemebene* konzentriert sich auf die Kundenperspektive und zielt darauf ab, einen Wert für den Kunden durch das System zu realisieren. In diesem Prozess geht es um die Kommunikation und Koordination zwischen den Personen, die die Rollen des Auftraggebers und des Product Owners einnehmen.
- Der Prozess der *Elementebene* konzentriert sich auf die Benutzerperspektive und zielt darauf ab, einen Wert für den Benutzer zu realisieren. In diesem Prozess geht es um die Kommunikation und Koordination zwischen dem Umsetzungsteam und dem Product Owner.

Wenn wir über die Prozessebene sprechen, wird der Unterschied zwischen Auftraggeber-, Kunden- und Benutzer-Rolle wieder wichtig (siehe Abschnitt 1.2.3), da sich jede Ebene auf eine bestimmte Perspektive konzentriert.

Arbeitsprodukttypen für jede Ebene

Auf allen drei Ebenen verwendet der Prozess die folgenden Arbeitsprodukttypen, um die Arbeit zu verwalten:

- Workitem: Eine kohärenter und dokumentierter Arbeitsauftrag
- Backlog: Eine geordnete Liste von Workitems, die die Reihenfolge darstellt, in der die Workitems bearbeitet werden sollen
- Kanban-Board: Eine Visualisierung der verschiedenen Phasen eines Prozesses. Die Karten stellen die Arbeitsaufträge (Workitems) und die Spalten für die einzelnen Phasen des Prozesses dar. Es können Grenzen für unfertige Erzeugnisse eingeführt werden, um den Umfang der parallelen Arbeit in einer Spalte zu begrenzen. Außerdem können Zeilen verwendet werden, um ein Arbeitsauftrag einer bestimmten Rolle zuzuordnen.

Zusätzlich werden zwei zusätzliche Meilensteine definiert, die für Arbeitsaufträge auf allen drei Ebenen gelten:

- Definition of Ready: Definition allgemeiner Kriterien, die erfüllt sein müssen, damit ein mit einem Arbeitsauftrag begonnen werden kann
- Definition of Done: Definition allgemeiner Kriterien, die erfüllt sein müssen, damit ein Arbeitsauftrag als erledigt gilt

Die Einzelheiten zum Management der Arbeit auf den drei Ebenen werden in Abschnitt 5.3.2 zusammen mit einem Überblick über die relevanten Typen von Arbeitsaufträgen vorgestellt. Die Einzelheiten zu den jeweiligen Arbeitsaufträgen (Inhalt und Regeln für das Schreiben, einschließlich der Definition of Done und Definition of Ready) sind in Abschnitt 5.3.2.3 aufgeführt. Um die allgemeine Mechanik des Prozesses abzuschließen, werden in Abschnitt 5.3.3.1 die Beziehungen zwischen den verschiedenen Arbeitsaufträgen und den Designkonzepten zusammengefasst.

Vier Phasen des Entwicklungs- und Betriebsschritts

Um den Prozess zu strukturieren, definieren wir vier Phasen:

- Phase 1: Erste Release-Planung und Backlog-Vorbereitung. In dieser Phase erstellen der Product Owner und das Umsetzungsteam ein erstes Backlog, das für den Beginn der Entwicklung des ersten Releases ausreicht.
- Phase 2: Entwicklung des ersten Releases. In dieser Phase arbeiten das Umsetzungsteam und der Product Owner am ersten Release.
- Phase 3: Weitere Entwicklung während des Betriebs. In dieser Phase ist das erste Release der digitalen Lösung in Betrieb, das Umsetzungsteam wartet es und arbeitet an der Weiterentwicklung durch weitere Releases.
- Phase 4: Außerbetriebnahme. In dieser Phase wird die Lösung vom Markt genommen.

Wichtige Ereignisse für die Strukturierung der Arbeit

Für die Entwicklung des ersten Releases (Phase 2) und die weitere Entwicklung (Phase 3) werden die folgenden Ereignisse zur Strukturierung der Arbeit herangezogen:

- Daily: Eine kurze und zeitlich begrenzte tägliche Besprechung, um den Fortschritt seit dem letzten Daily zu überprüfen
- Release-Planung: Eine zeitlich begrenzte, regelmäßige Besprechung, in der Personen mit der Auftraggeber-Rolle, der Product Owner und das Team die Backlogs mit den enthaltenen Arbeitsaufträgen auf Lösungs- und Systemebene pflegen
- Iterationsplanung: Eine zeitlich begrenzte, regelmäßige Besprechung, in der Product Owner und Team das Backlog mit den enthaltenen Arbeitsaufträgen auf Elementebene pflegen
- Iteration: Ein fester Zeitraum, in dem das Umsetzungsteam an den Arbeitsaufträgen arbeitet
- Retrospektive: Ein regelmäßiges Treffen zur Verbesserung der eigenen Arbeitsweise, um den Prozess zu überprüfen und anzupassen
- Lösungsreview: Eine Präsentation und Überprüfung der Ergebnisse einer Iteration für den Auftraggebenden und andere relevante Stakeholder

Die erste Phase, Backlog-Preparation (Vorbereitung), ist eine besondere Phase. Die Einzelheiten dieser Phase werden in Abschnitt 5.3.4 beschrieben. Weitere Einzelheiten über die Entwicklung des ersten Releases (Phase 2) werden in Abschnitt 5.3.5 erörtert. In Abschnitt 5.3.6 wird die weitere Entwicklung (Phase 3) erörtert und in Abschnitt 5.3.7 wird der Prozess für den Entwicklungs- und Betriebsschritt mit der Außerbetriebnahme der digitalen Lösung (Phase 4) abgeschlossen.

5.3.2 Management der Arbeit auf allen Abstraktionsebenen

Im Folgenden werden das Management des Prozesses und die notwendigen Arbeitsprodukte auf den drei Abstraktionsebenen des Bauprozesses vorgestellt. Wir stellen auch die Arbeitsaufträge für jede Ebene und ihren jeweiligen Zweck vor. Weitere Einzelheiten zur Ausarbeitung der Arbeitsaufträge in Bezug auf die Designkonzepte werden in Abschnitt 5.3.2.3 vorgestellt. Wir beginnen auf der Elementebene und bewegen uns dann nach oben.

5.3.2.1 Management der Arbeit auf der Elementebene

Der Prozess auf der Elementebene konzentriert sich auf die Benutzerperspektive und zielt darauf ab, einen Mehrwert für den Benutzer zu schaffen; er dient der Kommunikation und Koordination zwischen dem Umsetzungsteam und dem Product Owner.

5.3.2.1.1 Arbeitsaufträge auf der Elementebene zur Dokumentation der zu erledigenden Arbeiten

Die Arbeit auf der Elementebene hat ein breites Spektrum, das wir mit den folgenden Arbeitsaufträgen erfassen, die die Elemente des digitalen Systems realisieren:

- User Story
- Technisches Workitem
- Konzept-Workitem
- Prototyp-Workitem
- Evaluations-Workitem
- Defekt

User Story: Eine Beschreibung eines Bedürfnisses aus der Sicht eines Benutzers zusammen mit dem erwarteten Nutzen, wenn dieses Bedürfnis erfüllt ist

Die User Story ist der zentrale Arbeitsauftrag, da er die Umsetzungsarbeit darstellt, die zur Realisierung der Bedürfnisse eines Benutzers führt. Die User Story ist ein sehr beliebtes Konzept (vgl. [Cohn2004]) und wird in verschiedenen Ansätzen verwendet. Der Vorteil der Strukturierung der Umsetzungsarbeit anhand des Wertes für Benutzer (User Value) ist, dass das Ergebnis einer umgesetzten User Story direkt sichtbar und für die Benutzer erlebbar ist. Auf diese Weise können Benutzer und andere Stakeholder sofortiges Feedback zu den realisierten Funktionen geben, was eine schnelle Evaluierung und Feedbackschleifen ermöglicht.

Technisches Workitem: Arbeitsauftrag zur Erarbeitung/Realisierung einer technischen Voraussetzung für die Realisierung einer User Story

Allerdings reichen User Storys allein nicht aus, da es auch Implementierungsarbeiten gibt, die nicht direkt mit den Bedürfnissen der Benutzer zusammenhängen (z. B. die Implementierung technischer Schnittstellen). Die Aufnahme solcher Arbeiten in eine User Story würde zu sehr großen Arbeitsposten führen, die nur schwer zu schätzen und zu verwalten sind. Um Implementierungsarbeiten zu erfassen, die nicht direkt mit einem Benutzerbedarf zusammenhängen, definieren wir das technische Workitem als Typ eines Arbeitsauftrags. Ein technisches Workitem kann dann zur Voraussetzung für den Beginn der Arbeit an einer User Story werden.

Konzept-Workitem: Arbeitsauftrag zur Ausarbeitung von konzeptuellen Details als Voraussetzung für die Realisierung einer User Story

Die Details, was in einer User Story oder einem technischen Workitem implementiert wird, werden durch Designkonzepte und/oder Realisierungskonzepte festgehalten. Diese Konzepte werden zum Zeitpunkt der Definition von User Storys oder technischen Workitems nicht unbedingt im Detail ausgearbeitet. Um die Arbeit zur Ausarbeitung der Konzepte zu erfassen und zu verwalten, definieren wir einen Konzept-Workitem-Typ. Ein Konzept-Workitem kann

dann die Voraussetzung für den Beginn der Arbeit an einer User Story oder einem technischen Workitem sein.

Prototyp-Workitem: Arbeitsauftrag zur Erstellung eines Prototyps eines Aspekts der digitalen Lösung

Neben der Umsetzung arbeitet das Umsetzungsteam oft auch an Prototypen. Um die Implementierungsarbeit an der Lösung von der Arbeit an den Prototypen zu unterscheiden, definieren wir das Prototyp-Workitem als Typ eines Arbeitsauftrags. Der konkrete Inhalt eines Prototyp-Workitems hängt von der Art des Prototyps ab. Das Ergebnis eines Prototypen-Workitems ist ein bestimmter Prototyp oder eine Erweiterung eines bestehenden Prototyps.

Evaluations-Workitem: Arbeitsauftrag zur Evaluierung eines Prototyps oder eines bereits realisierten Aspekts der digitalen Lösung

Prototypen werden häufig verwendet, um einen bestimmten Aspekt der digitalen Lösung zu bewerten. Zur Erfassung der Bewertungsarbeit definieren wir das Evaluations-Workitem als Typ eines Arbeitsauftrags. Evaluations-Workitems können auch dazu verwendet werden, einen bestimmten Aspekt der bereits implementierten digitalen Lösung zu bewerten.

Defekt: Arbeitsauftrag zu einem Fehler in der digitalen Lösung, der analysiert und behoben werden muss

Der letzte Typ von Arbeitsaufträgen auf der Elementebene ist der Defekt. Defekte sind ein normaler Bestandteil eines jeden Bauprozesses und sollten explizit behandelt werden. Mit einem Defekt wird der Aufwand für die Analyse und das Verständnis der Ursache(n) eines bestimmten Fehlers erfasst.

5.3.2.1.2 System-Backlog und System-Board zur Visualisierung des Fortschritts und der zu erledigenden Arbeit

Alle oben eingeführten Typen von Arbeitsaufträgen, werden im System-Backlog verwaltet. Dabei handelt es sich um eine Liste von Arbeitsaufträgen (auch Backlog-Items genannt), deren Reihenfolge vom Product Owner und dem Umsetzungsteam festgelegt wird. Die Reihenfolge der Backlog-Items stellt die Priorität und Arbeitsreihenfolge dar, in der die Elemente vom Umsetzungsteam und dem Product Owner bearbeitet werden. Die Reihenfolge der Elemente ist daher ein wichtiges Managementinstrument zur Kontrolle der Arbeit auf der Elementebene.

Zur Visualisierung und Verwaltung des Fortschritts auf der Elementebene verwenden wir das System-Board als Kanban-Board. Zusätzlich zu den Spalten sollte das System-Board eine Zeile für jedes Mitglied des Umsetzungsteams und für den Product Owner haben. Diese Zeilen zeigen, welche Person an welchem Arbeitsauftrag arbeitet. Darüber hinaus ist eine Zeile mit dem Titel *Fast Track* nützlich, um anzuzeigen, dass ein bestimmter Arbeitsauftrag kritisch ist und mit hoher Priorität bearbeitet werden sollte.



Abbildung 59 - System-Backlog und System-Board als Kanban-Board

Abbildung 59 zeigt einen abstrakten System-Backlog zusammen mit einem System Board als Kanban-Board. Die Arbeit mit Arbeitsaufträgen auf der Elementebene sieht folgendermaßen aus:

- *ToDo*: Jede User Story, die der Definition of Ready entspricht und einer bestimmten Person zugewiesen ist oder so wichtig ist, dass sie auf die Schnellspur (siehe oben) gesetzt wird, kommt auf die System Board.
- *In Bearbeitung*: Wenn eine Person an einer User Story arbeitet, wird die User Story in die Spalte *In Bearbeitung* eingetragen.
- *In Prüfung*: Wenn eine User Story die Definition of Done erfüllt, kann sie in die Spalte *In Prüfung* verschoben werden, was bedeutet, dass ein Teammitglied das Ergebnis der User Story prüfen kann.
- *Blockiert*: Wenn die Arbeit an einer User Story aus externen Gründen nicht fortgesetzt werden kann, wird sie in die Spalte *Blockiert* verschoben.
- *Erledigt*: Wenn eine User Story erfolgreich geprüft wurde, kann sie in die Spalte *Erledigt* verschoben werden. Wenn eine Iteration abgeschlossen ist, können alle User Storys in dieser Spalte entfernt werden.

5.3.2.2 Verwaltung der Arbeit auf der Systemebene

Der Prozess auf Systemebene konzentriert sich auf die Kundenperspektive mit dem Ziel, durch das System einen Mehrwert für den Kunden zu schaffen. Er dient der Kommunikation und Koordination zwischen den Personen, die die Rollen des Auftraggebers und des Product Owners einnehmen.

5.3.2.2.1 Epic als Arbeitsaufgabe auf der Systemebene

Auf der Systemebene wird die Arbeit mit Hilfe von Epics verwaltet. Das Epic ist ein Kernelement dieses Schritts des Bauprozesses und wird wie folgt definiert:

Epic: A work item that describes a characteristic of a digital system that provides value for stakeholders.

Für den in diesem Abschnitt beschriebenen Prozess konzentrieren wir uns in erster Linie auf den Kundennutzen, der durch das System als Teil eines Releases realisiert werden soll. Ein

Epic bezieht sich daher auf ein bestimmtes Ziel, das das digitale System als Teil des Wertversprechens der digitalen Lösung erreichen soll.

5.3.2.2 Story Map als Werkzeug zur Strukturierung und Visualisierung von Epics und User Storys

Um die Details eines Epics zu definieren, zu verstehen und zu verwalten, verwenden wir eine Technik namens *Story Mapping* [Patt2014]. Die Hauptstruktur einer Story Map ist eine zweidimensionale Anordnung von User Storys. Eine User Story repräsentiert ein bestimmtes Benutzerbedürfnis und wird auch auf der Elementebene verwendet (siehe Abschnitt 5.3.2.1). In der horizontalen Dimension der Story Map liegt der Schwerpunkt auf dem Hauptstrang, das heißt auf dem narrativen Ablauf des Systems (oder auf dem durch das System bereitgestellten Gesamtprozess). Mit Narrativer Fluss sind die verschiedenen Schritte oder Aktivitäten gemeint, die ein Benutzer mit dem System in einem Ende-zu-Ende Fluss durchführen kann. Die Use Cases und Szenarien des digitalen Systems sind eine gute Perspektive für diesen Ablauf. Die vertikale Dimension bietet Details für jeden Teil des Erzählflusses sowie eine Trennung der Elemente nach der Priorität der User Storys aus der Funktionsperspektive (die Reihenfolge der Storys definiert eine Priorität, Storys oben sind wichtiger als die darunter liegenden in derselben Spalte).

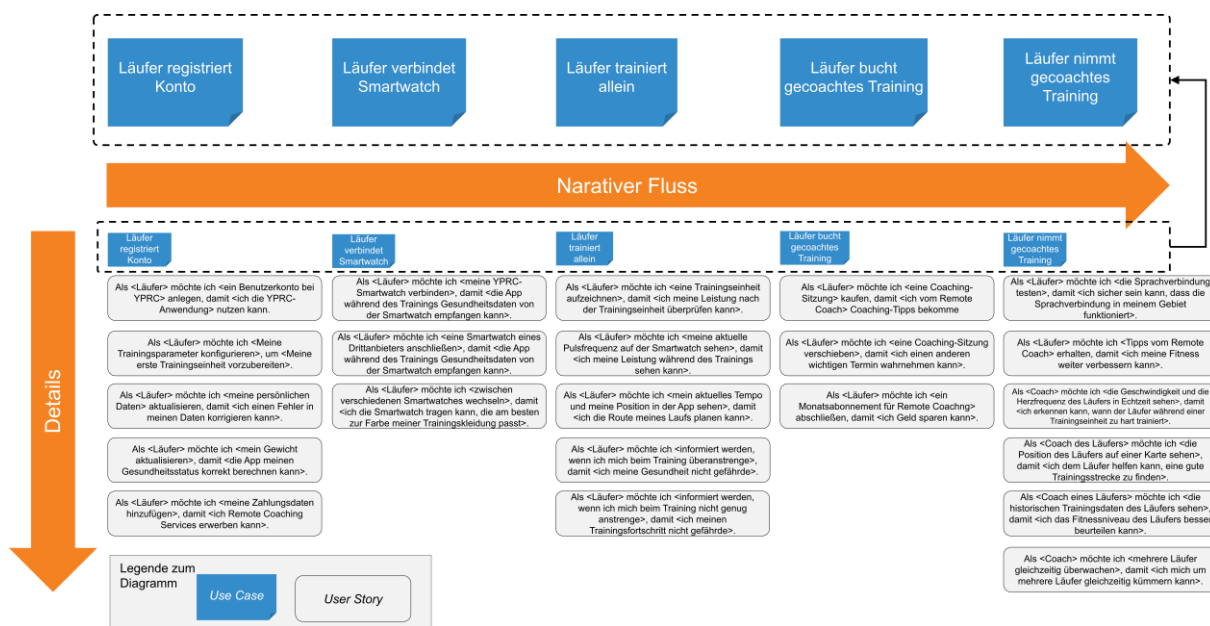


Abbildung 60 - Beispiel für eine Story Map aus der YPRC-Fallstudie

Abbildung 60 - Beispiel für eine Story Map aus der YPRC-Fallstudie zeigt eine Miniatur-Story Map aus der YPRC-Fallstudie. Die User Storys sind hier nicht relevant und sollen auch nicht lesbar sein. Die Use Cases sind zur besseren Lesbarkeit des Erzählflusses vergrößert dargestellt. Die vollständige Version finden Sie in den Fallstudienunterlagen.

Abbildung 61 zeigt eine Zwischenversion der YPRC Story Map (siehe Abbildung 60 - Beispiel für eine Story Map aus der YPRC-Fallstudie), bei der die Mitte vergrößert wurde, so dass die Titel der User Storys lesbar sind.

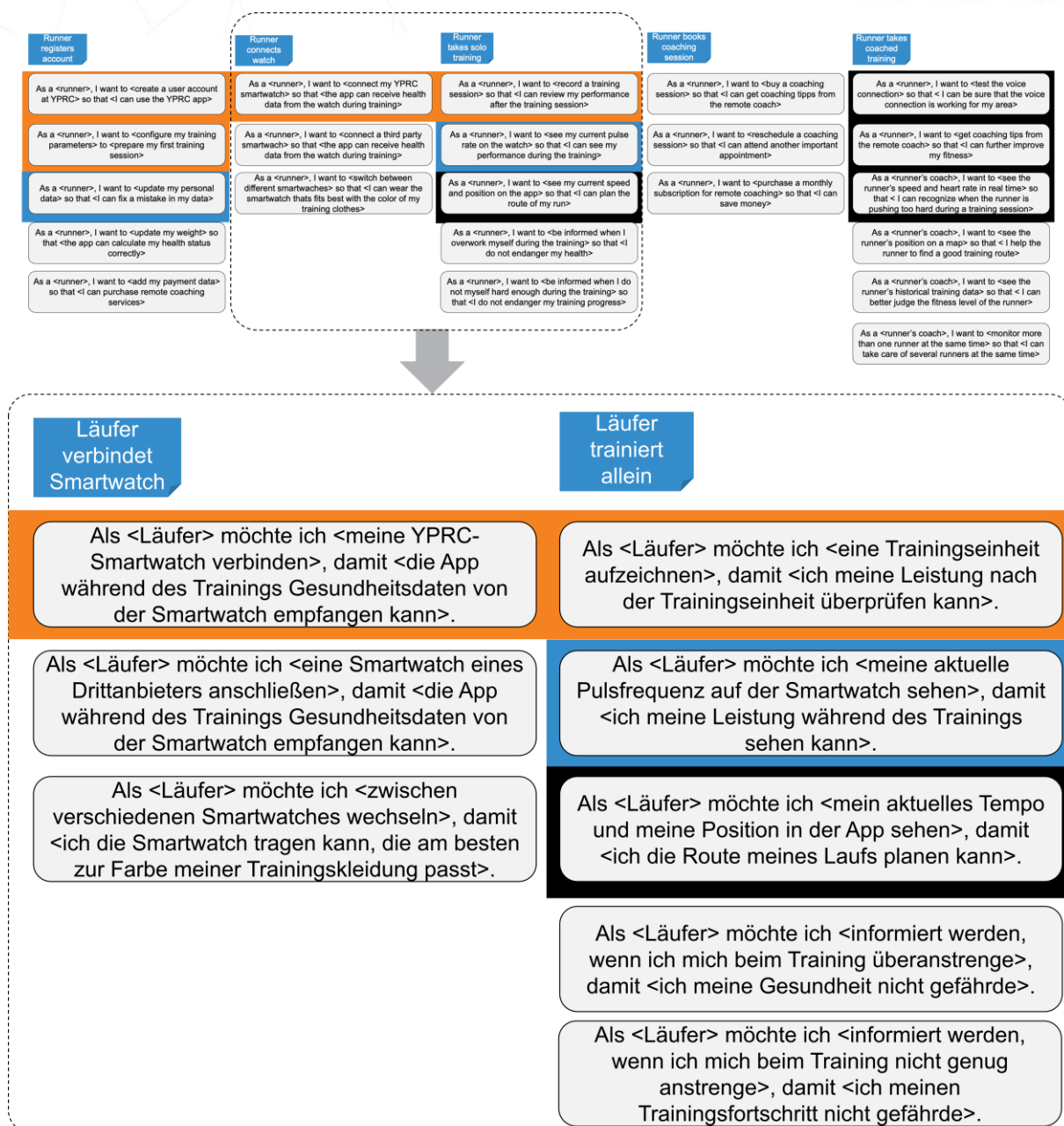


Abbildung 61 - Verwendung einer Story Map zur Priorisierung von User Storys

In diesem Beispiel sind die User Storys in drei Epics unterteilt:

- Orangerfarbenedes Epic: Die Storys mit orangefarbenem Hintergrund definieren eine erste und sehr einfache Funktionalität von YPRC, die aus vier User Storys besteht, die den Registrierungsprozess, die Verbindung mit der Smartwatch, die Möglichkeit, eine Trainingseinheit aufzuzeichnen und die Trainingsdaten anschließend zu überprüfen, beschreiben.
- Blaues Epic: Die blau hinterlegten Storys definieren weitere Funktionen des YPRC, die aus zwei zusätzlichen User Storys bestehen. Das Epic soll dem Läufer die Möglichkeit geben, seine persönlichen Daten in der App zu ändern und seine aktuelle Pulsfrequenz auf der Smartwatch zu sehen.

- **Schwarzes Epic:** Die schwarz hinterlegten Storys definieren weitere Funktionalitäten des YPRC, die aus vier User Storys bestehen. Das Epic soll die aktuelle Geschwindigkeit und Position des Läufers auf einer Karte anzeigen und eine erste Version der Ferncoaching-Funktion bieten.

Die drei Epics sind aus funktionaler Sicht natürlich sehr klein, könnten aber in einem einzigen Release zusammengefasst werden. In diesem Beispiel geht es nicht um eine realistische Menge an Funktionen, sondern darum, den Wert der Story Map für die Definition von Epics zu verstehen. Die exemplarische Story Map ist eher klein. Das bedeutet aber nicht, dass Story Maps für größere Lösungen nicht nützlich sind. Es ist möglich, wirklich große Story Maps mit mehreren Use Cases zu erstellen. In einer solchen Situation kann die Story Map leicht große Wände eines Raumes einnehmen.

Die Story Map kann für die anfängliche Definition von User Storys verwendet werden (siehe Abschnitt 5.3.4), sie ist aber auch eine Technik zur Dokumentation und Verwaltung der Entwicklung während des gesamten Entwicklungsprozesses. Wir empfehlen daher, die Story Map während des gesamten Entwicklungsprozesses beizubehalten. Mit dieser Perspektive ist die Story Map ein wichtiges Werkzeug, das Ihnen hilft, sich auf die Lösungsperspektive zu konzentrieren und die detaillierte Ansicht der System- und Elementdesignkonzepte zu ergänzen.

5.3.2.2.3 Epic Board zur Visualisierung des Fortschritts und der zu erledigenden Arbeit

Abbildung 62 - Story Map und Epic Board zeigt die Struktur eines Epic Boards. Die Story Map dient als Werkzeug zur Pflege und Definition der Epics während des Bauprozesses, und das Epic Board wird zum Verwalten der Arbeit während des Entwicklungs- und Betriebsprozesses verwendet.



Abbildung 62 - Story Map und Epic Board

Im Folgenden werden die Spalten des Epic Boards besprochen:

- **ToDo:** Jedes Epic, das die Definition of Ready (siehe oben) erfüllt, wird auf dem Epic Board in der *ToDo-Spalte* eingetragen. Die Reihenfolge der Epics in dieser Spalte kann verwendet werden, um die Priorität der Epics und der damit verbundenen Arbeitsaufträge anzuzeigen.
- **Vorbereitung:** Sobald das Umsetzungsteam mit der Arbeit an konzeptuellen oder technischen Arbeitsaufträgen beginnt, die zu einem Epic gehören, wird das Epic in die Spalte *Vorbereitung* verschoben. Dies ist ein Hinweis für den Product Owner und den Auftraggebenden, dass die Details eines Epics ausgearbeitet wurden und die Vorbereitung der Realisierung im Gange ist.

- *Realisierung*: Wenn das Umsetzungsteam mit der Arbeit an User Storys zu einem bestimmten Epic beginnt, wird das Epic in die Spalte *Realisierung* verschoben. Es ist nicht notwendig, dass alle Konzept-Workitems für dieses bestimmte Epic erledigt sind. Nur die Konzept-Workitems, die sich auf die in Arbeit befindlichen User Storys beziehen, müssen bearbeitet werden.
- *Evaluierung*: Wenn alle User Storys, technischen Arbeitsaufträge und Konzept-Workitems erledigt sind, wird das Epic in die Spalte *Evaluierung* verschoben, und die abschließende Bewertung eines bestimmten Epics sollte beginnen. Es ist wichtig zu wissen, dass sich dieses Kriterium für die Aufnahme eines Epics in diese Spalte von den anderen Spalten unterscheidet. Für diese Phase des Bauprozesses eines Epics sollten spezielle Evaluations-Workitems definiert werden, die bewerten, ob die digitale Lösung die im Epic definierten Akzeptanzkriterien erfüllt.
- *Bereit für den Betrieb*: Sobald die Definition of Done für Epics erfüllt ist (d.h. alle Evaluations-Workitems wurden erledigt und die Ergebnisse der Evaluation zeigen, dass die Akzeptanzkriterien erfüllt sind), kann das Epic in die Spalte „betriebsbereit“ verschoben werden. Sobald das Epic in Betrieb ist, wird das Epic vom Board entfernt (siehe Abschnitt 5.3.5). Das Entfernen des Epics vom Board dient hauptsächlich dazu, das Board so schlank wie möglich zu halten.

Neben den oben beschriebenen Spalten gibt es auf dem Epic Board noch die Spalte *Gesperrt*. Ein Epic wird in diese Spalte verschoben, wenn ein mit diesem Epic verbundener Arbeitsauftrag auf Element- oder Lösungsebene gesperrt ist.

Diese Art der Arbeit mit Epics macht das Epic zum zentralen Verwaltungsinstrument, das die verschiedenen anderen Arbeitsaufträge integriert. Die Pflege dieser verschiedenen Beziehungen zwischen Epics und Arbeitsaufträgen ist Teil des Managements und nicht des Designs. Aus der Perspektive des Digital Designs ist es jedoch wichtig, die Beziehung zu und die Auswirkungen der Definition von Epics auf Designkonzepte zu verstehen, wie sie in Abbildung 67 - Beziehung zwischen epischen Konzepten und Designkonzepten hervorgehoben werden.

5.3.2.3 Management der Arbeit auf der Lösungsebene

Der Prozess auf Lösungsebene konzentriert sich auf die Auftraggeber-Rolle und zielt darauf ab, die Ziele des Auftraggebenden durch die digitale Lösung zu erreichen; er dient der Kommunikation und Koordination zwischen der Auftraggeber-Rolle und dem Product Owner.

Das Management der Arbeit auf Lösungsebene besteht aus zwei Teilen: Management des Release-Plans und Management der Arbeit auf Lösungsebene.

5.3.2.3.1 Release-Plan als übergeordnetes Managementinstrument

Der Release-Plan ist ein übergeordnetes Managementinstrument, das die für die digitale Lösung geplanten Releases definiert (die bereits erstellten Releases können auch in der Release-Liste gepflegt werden). Wie oben definiert, ist ein Release gleichzeitig eine bestimmte Instanz der digitalen Lösung und ein Zeitraum, in dem die bestimmte Instanz der digitalen Lösung realisiert wird. Die Arbeit am Release-Plan findet im Rahmen der Release-Planung statt (siehe Abschnitt 5.3.1). Einzelheiten zur Definition eines bestimmten Releases in Bezug auf die Funktionalität (welche Epics sollen Teil des Releases sein) und in Bezug auf Zeit und Budget (welches Budget und welcher Zeitrahmen wird für das Release erwartet) werden in Abschnitt 5.3.4 erörtert.

5.3.2.3.2 *Arbeitsaufträge auf der Lösungsebene*

Die konkrete Arbeit auf der Lösungsebene hängt von der jeweiligen Art der digitalen Lösung ab und kann sehr unterschiedlich sein. Wir definieren daher nur abstrakte Arbeitsaufträge. Ein Lösungs-Workitem ist ein Arbeitsauftrag, der Ressourcen oder andere Mittel bereitstellt, die für die Entwicklung oder den Betrieb der digitalen Lösung erforderlich sind. Beispiele für solche Arbeitsaufträge auf Lösungsebene sind:

- Organisation eines Kunden- oder Benutzertests für die digitale Lösung
- Beschaffung von Personal oder Material, das zum Aufbau der digitalen Lösung erforderlich ist
- Beschaffung von technischen Ressourcen (z. B. Rechenzentren), die für den Betrieb einer digitalen Lösung erforderlich sind

5.3.2.3.3 *Lösungs-Backlog und Lösungsboard zur Visualisierung und zum Management der zu erledigenden Arbeit*

Zum Management der Arbeit werden ein Lösungs-Backlog und ein Lösungsboard verwendet. Das Lösungs-Backlog ist eine vom Product Owner geordnete Liste von Arbeitsaufträgen, die zur Realisierung eines Releases der digitalen Lösung abgearbeitet werden muss. Der Product Owner bestimmt die Reihenfolge der Arbeitsaufträge in der Liste.

Das Lösungsboard ist ein Kanban-Board zum Management der Arbeitsaufträge auf Lösungsebene und hat die folgenden Spalten:

- *ToDo*: Arbeitsaufträge, die der Definition of Ready entsprechen
- *In Arbeit*: Arbeitsaufträge in Arbeit
- *In Überprüfung*: Arbeitsaufträge, die überprüft werden
- *Blockiert*: Arbeitsaufträge, die blockiert sind
- *Erledigt*: Arbeitsaufträge, die die Kriterien der Definition of Done erfüllen (Arbeitsaufträge werden am Ende eines Releases entfernt)

5.3.2.4 *Das Gesamtbild zum Management von Arbeit auf allen Abstraktionsebenen*

In den vorangegangenen Abschnitten haben wir mehrere Arbeitsprodukte für das Management der Arbeit auf den drei Abstraktionsebenen vorgestellt. Abbildung 63 - Beziehungen zwischen den Arbeitsprodukten auf den drei Ebenen zeigt die Arbeitsprodukte, die auf den drei Ebenen eingeführt wurden.

Ein Aspekt, der auf den ersten Blick verwirrend sein kann, ist die Tatsache, dass die Arbeitsprodukte nicht symmetrisch definiert sind. Wir haben zum Beispiel ein Lösungs- und ein System-Backlog definiert, aber kein Element-Backlog. Wir haben auch kein Element-Board definiert. Der Grund dafür ist, dass es nicht notwendig ist, die Arbeit auf der System- und der Elementebene zu trennen, da das System aus lauter Elementen besteht. Die Arbeit an einem bestimmten Element bedeutet also eine Arbeit am System. Dennoch ist es notwendig, zwischen dem Management der detaillierten Arbeit an den Elementen (System-Backlog und System-Board) und der langfristigen Perspektive der Entwicklung des Systems als Ganzes zu unterscheiden. Für die langfristige Perspektive haben wir die Story Map und das Epic Board auf der Systemebene eingeführt.

Story Map als Kern für das Management von Entwicklung und Betrieb

Die Story Map ist das zentrale Werkzeug für das Management der Entwicklung und des Betriebs einer digitalen Lösung - sie setzt Kundennutzen (definiert durch Epics) mit Wert für den Benutzer (User Value; definiert durch User Storys) in Beziehung. Mit der Story Map erhält der Product Owner einen Überblick darüber, was die digitale Lösung erreichen soll und wie das digitale System zum Erreichen des Kundennutzens beitragen wird.

Die Arbeit an der digitalen Lösung und dem digitalen System wird in Bezug auf die Epics priorisiert. Aus der Perspektive der Lösungsebene wird ein Release der digitalen Lösung durch eine Reihe von Epics definiert. Aus der Perspektive der Systemebene und der Elementebene wird die Arbeit an den User Storys und den zugehörigen Arbeitsaufträgen entsprechend der Priorität der Epics priorisiert. Der Priorisierungsprozess sollte keinesfalls als Top-Down-Prozess verstanden werden. Die Prioritäten auf allen drei Ebenen können aus verschiedenen Gründen geändert werden. Bei der Entscheidung über Prioritäten sollte der Product Owner immer das Lösungs-Backlog, die Story Map und das System-Backlog zusammen betrachten, um die optimale Priorität zu finden.

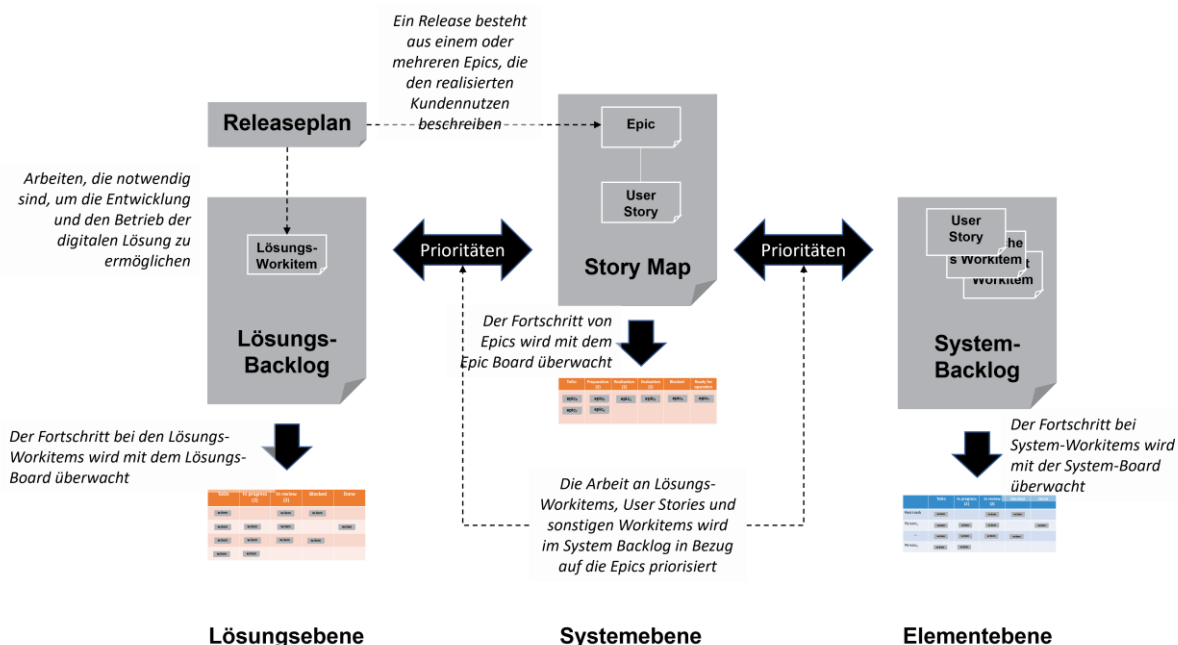


Abbildung 63 - Beziehungen zwischen den Arbeitsprodukten auf den drei Ebenen

Eine letzte Bemerkung zu den Arbeitsprodukten in Abbildung 63 - Beziehungen zwischen den Arbeitsprodukten auf den drei Ebenen: Die in diesem Abschnitt definierten Arbeitsprodukte wurden unter der Annahme definiert, dass ein einziges Team zusammen mit einem Product Owner an der digitalen Lösung arbeiten kann. Wenn die Größe und Komplexität einer digitalen Lösung die Koordination mehrerer Teams und mehrerer Product Owner erfordert, empfehlen wir, den hier vorgestellten Prozess nicht zu verwenden, da die Skalierung eines Bauprozesses mit mehreren Teams zusätzliche Managementstrukturen erfordert, die über einen Bauprozess auf Foundation Level Niveau hinausgehen.

5.3.3 Leitlinien für die Definition von Arbeitsaufträgen im Bauprozess

Im Folgenden finden Sie ausführliche Leitlinien für das Schreiben der unterschiedlichen Typen von Arbeitsaufträgen (Workitems), einschließlich einer detaillierten Vorlage, Schreibregeln

sowie einer Definition of Ready und Definition of Done. Diese Leitlinien gelten sowohl für die erstmalige Erstellung von Arbeitsaufträgen als auch für die Erstellung von Arbeitsaufträgen während der gesamten Entwicklung. Neben den aus Sicht von Digital Design relevanten Typen von Arbeitsaufträgen enthält dieser Abschnitt auch kurze Leitlinien zu weiteren Typen von Arbeitsaufträgen. Diese gehen zwar über den Rahmen des DDP Foundation Level Niveau hinaus, sind aber wichtig für das Verständnis des gesamten Bauprozesses.

Bevor wir auf die Details der verschiedenen Vorlagen für die Arbeitsaufträge eingehen, geben wir in Abschnitt 5.3.3.1 einen Überblick über die Beziehungen zwischen Arbeitsprodukten und Designkonzepten. Anschließend stellen wir in Abschnitt 5.3.3.2 eine allgemeine Vorlage für Arbeitsaufträge vor und fahren dann mit den Details zu den jeweiligen Typen fort.

5.3.3.1 Überblick über die Beziehungen zwischen Arbeitsprodukten und Designkonzepten

Im vorangegangenen Abschnitt haben wir die Arbeitsprodukte auf den drei Ebenen definiert. Bevor wir uns mit den detaillierten Leitlinien befassen, wollen wir einen Blick auf die Perspektive des Digital Designs auf diese Arbeitsprodukte werfen und die Beziehungen zwischen den Arbeitsprodukten und den Designkonzepten auf einer allgemeinen Ebene diskutieren. Das Verständnis dieser Beziehungen ist wichtig, um zu verstehen, wie Digital Design in das Management der Realisierungsarbeit integriert wird.

Abbildung 64 zeigt zwei Sichtweisen: die Sichtweise auf Designkonzepte und die Sichtweise auf Arbeitsprodukte. Außerdem zeigt die Abbildung die Beziehungen zwischen den Arbeitsprodukten und den Designkonzepten.

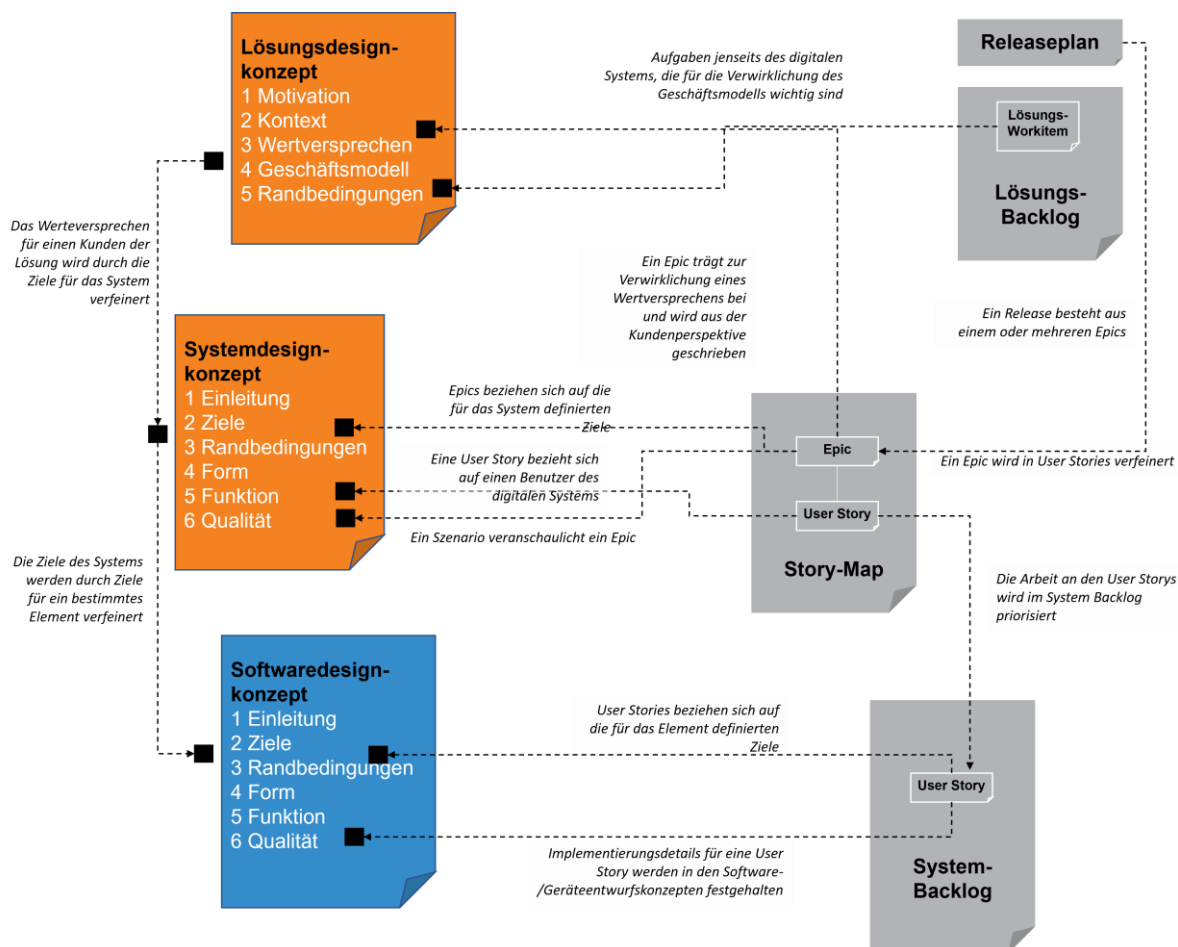


Abbildung 64 - Beziehungen zwischen Workitem-Management und Designkonzepten

Wir beginnen mit der Perspektive auf die Arbeitsprodukte. Ein Release definiert die langfristige Perspektive des Bauprozesses. Ein Release besteht aus einem oder mehreren Epics, und ein Epic wird in User Storys weiter verfeinert. Das bedeutet, dass ein bestimmtes Release den Kundennutzen schafft, der durch die Epics und die entsprechenden User Storys definiert ist. Eine ähnliche Hierarchie wird durch die Designkonzepte definiert. Das im Lösungsdesignkonzept definierte Wertversprechen wird durch die im Systemdesignkonzept definierten Ziele verfeinert, die wiederum durch Ziele auf der Elementebene verfeinert werden.

Diese Beziehungen implizieren, dass die Arbeit an Releases, Epics und User Storys die Arbeit an Wertversprechen, Systemzielen und Elementzielen bedeutet. Das heißt zum Beispiel, dass die Verfeinerung von User Storys die Verfeinerung von Zielen eines Elements und die Verfeinerung des Use Case eines Elements die Änderung einer oder mehrerer User Storys bedeutet. Ein DDP sollte diese Abhängigkeiten bei der Realisierung einer digitalen Lösung im Auge behalten, da er sonst den Anschluss an die Realisierungsaktivitäten verliert. Diese Beziehungen werden in Abschnitt 5.3.3 erläutert und unterstützen die Wahrung der Konsistenz zwischen Arbeitsprodukten und Designkonzepten. Das Verständnis dieser Zusammenhänge ist die Voraussetzung dafür, dass beide Perspektiven miteinander in Einklang gebracht werden können.

Wir möchten betonen, dass die Wahrung der Konsistenz zwischen Designkonzepten und Arbeitsprodukten kein Selbstzweck ist, sondern ein Instrument zur Integration der Design- und der Managementperspektive des Bauprozesses. Darüber hinaus möchten wir betonen, dass es unmöglich ist, eine vollständige Konsistenz zu erreichen, da Designkonzepte und Prozessartefakte ständig weiterentwickelt und überarbeitet werden. Dennoch ist der Aufwand, der betrieben wird, um beide Perspektiven so konsistent wie möglich zu halten, keine vergeudete Zeit, denn es ist das richtige Instrument, um die Managementperspektive auf den Bauprozess mit der Designperspektive in Einklang zu bringen.

In der YPRC-Fallstudie führen wir eine ergänzende Werkzeugunterstützung ein, die die Wahrung der Konsistenz zwischen beiden Perspektiven unterstützt.

5.3.3.2 Allgemeine Vorlage für Arbeitsaufträge

Im Allgemeinen besteht eine Workitem-Vorlage aus den folgenden Abschnitten:

- Bezeichner und Titel
- Abschnitt mit detaillierten Informationen über die zu erledigenden Arbeiten
 - Tätigkeitsbeschreibung
 - Vorbedingungen
 - Akzeptanzkriterien
 - Relevante Elemente
 - Epic-Referenz
- Abschnitt mit Managementinformationen
 - Bearbeiter
 - Ersteller
 - Release
 - Geschätzter Aufwand
 - Geleisteter Aufwand
 - Verbleibender Aufwand

Ähnlich wie bei den Bausteinen der Designkonzepte empfehlen wir die Verwendung eindeutiger Bezeichner für jeden Arbeitsauftrag, um die Rückverfolgbarkeit zwischen den Arbeitsaufträgen zu unterstützen. Die Festlegung eines angemessenen Titels, der eine klare Beschreibung der Arbeit liefert, wird die Verwaltung von Arbeitsaufträgen weiter unterstützen.

Detaillierte Informationen über die zu erledigenden Arbeiten

Die Detailinformationen zu einem Arbeitsauftrag bestehen aus der Aufgabenbeschreibung, die vom jeweiligen Typ des Arbeitsauftrags abhängt, sowie vier allgemeinen Abschnitten.

Die *Vorbedingungen* werden verwendet, um andere Aufgaben zu dokumentieren, die erledigt werden müssen, bevor die aktuelle Aufgabe bearbeitet werden kann. Die *Akzeptanzkriterien* dienen dazu, einzelne Kriterien zu definieren, die erfüllt sein müssen, damit der Arbeitsauftrag als erledigt gilt. Akzeptanzkriterien ergänzen die allgemeinen Kriterien, die durch die Definition of Done bereitgestellt werden, und sind ein nützliches Werkzeug, um eindeutige Kriterien zu definieren, die vom Arbeitsauftrag erfüllt werden müssen. Die *relevanten Elemente* verweisen auf die Elemente der digitalen Lösung, von denen angenommen wird, dass sie von der Aufgabe betroffen sind. Diese Referenz ist nützlich, um den Überblick über die zu einem bestimmten Element der digitalen Lösung gehörenden Arbeitsaufträge zu behalten. Die *Epic-Referenz* wird verwendet, um einen Arbeitsauftrag mit einem bestimmten Epic in Verbindung zu bringen. Diese Referenz ermöglicht einen Überblick über alle Arbeitsaufträge, die zu einem bestimmten Epic beitragen.

Managementinformationen

Die Managementinformationen sind wichtig für die Steuerung der Arbeiten während des Bauprozesses. Es ist nützlich, zwischen dem *Bearbeiter* und dem *Ersteller* zu unterscheiden, um den Überblick über die Person zu behalten, die den Arbeitsauftrag erstellt hat (z. B. um Fragen zu stellen).

Die *Release-Angabe* ist wichtig für die Dokumentation des Releases, zu dem eine Aufgabe beiträgt. Diese Informationen können z. B. dazu verwendet werden, den Arbeitsaufwand für ein bestimmtes Release zu kontrollieren.

Der *geschätzte Aufwand*, der *geleistete Aufwand* und der *verbleibende Aufwand* sind nützlich, um den Überblick über den Arbeitsaufwand zu behalten (siehe Abschnitt 5.3.4.3).

Die oben beschriebene Liste von Abschnitten für einen Arbeitsauftrag ist eine eher minimale Liste, und Fachleute im Management werden sich mehrerer anderer Aspekte bewusst sein. Experten sollten daher bedenken, dass der hier beschriebene Bauprozess für Einsteiger gedacht ist.

Zur Erinnerung: Arbeitsaufträge sind vorübergehend

Da es sich bei dem Arbeitsauftrag um ein temporäres Arbeitsprodukt handelt, sollten die Detailinformationen des Arbeitsauftrags nur Informationen enthalten, die für die jeweilige Aufgabe von besonderer Bedeutung sind. Alle Informationen zur digitalen Lösung, die erhalten bleiben müssen, sollten in den Konzepten dokumentiert werden. Verweise zwischen Arbeitsaufträgen und Konzepten werden verwendet, um diese Trennung zu erreichen und um weitere Details zu liefern.

Im Folgenden geben wir detaillierte Leitlinien für die Aufgabenbeschreibung jedes Typs eines Arbeitsauftrags.

5.3.3.3 Lösungs-Workitems

Lösungs-Workitems sind ein Werkzeug zur Erfassung aller Arbeiten auf der Lösungsebene. Der Arbeitsbereich auf der Lösungsebene ist sehr breit gefächert (siehe 5.3.2.3), und wir definieren daher eine sehr allgemeine Vorlage für ein Lösungs-Workitem. Die Aufgabenbeschreibung der Lösungsaufgabe sollte ausreichend detailliert sein, damit der Bearbeiter die Aufgabe bearbeiten kann. Akzeptanzkriterien, Definition of Done und Definition of Ready müssen je nach der jeweiligen digitalen Lösung festgelegt werden.

Beschaffung einer Testumgebung für den Zahlungsanbieter "pay friend"	
Voraussetzungen	<ul style="list-style-type: none"> Keine
Aufgabenbeschreibung	<ul style="list-style-type: none"> Verhandeln Sie eine Testumgebung für den Zahlungsanbieter "pay friend", um den Anwendungsfall Coachingkauf UC-10 in der Entwicklungsumgebung zu evaluieren
Akzeptanzkriterien	<ul style="list-style-type: none"> Die IP-Adresse für die Testumgebung ist verfügbar Testkonten für drei Benutzer sind verfügbar Verwaltungskonto zur Überprüfung der Zahlung ist vorhanden
Relevante Elemente	<ul style="list-style-type: none"> ESys-2 Zahlungsanbieter
Epic-Referenz	<ul style="list-style-type: none"> YP-17 Als <Sportler> möchte ich <von einem Remote Coach gewarnt werden>, damit <ich mich während einer Trainingseinheit nicht zu sehr anstrenge>.

Abbildung 65 - Beispiel für eine Lösungsaufgabe aus der YPRC-Fallstudie

Obwohl dies nur eine sehr abstrakte Beschreibung eines Lösungs-Workitems ist, können Lösungsaufgaben sehr wichtig für den Bauprozess einer digitalen Lösung sein. Abbildung 65 - Beispiel für eine Lösungsaufgabe aus der YPRC-Fallstudie zeigt ein Beispiel für eine Lösungsaufgabe aus der YPRC-Fallstudie, die sich mit einem bestimmten Zahlungsanbieter befasst, der in den YPRC eingeschlossen sein soll.

In dem Beispiel geht es um die Organisation einer Testumgebung für den Zahlungsanbieter. Ein solcher Arbeitsauftrag ist typisch für die Lösungsebene während des Bauprozesses, da es sich mit Ressourcen befasst, die für die Arbeit an der digitalen Lösung notwendig sind.

5.3.3.4 Epics in Bezug auf Lösungs- und Systemdesignkonzept

Das Epic dient in unserem Bauprozess zwei Zwecken. Dies sind:

- Eine kurze und einfache Beschreibung eines Wertes, den das digitale System einem Kunden der digitalen Lösung bietet
- Ein Management-Werkzeug zur Organisation und Priorisierung der Entwicklungsarbeit auf Systemebene

Für Einsteiger im Bereich des Digital Designs ist die Kombination dieser beiden Aspekte (Mischung aus Inhalt und Verwaltung) oft schwer zu verstehen. Die Stärke dieser Kombination besteht darin, dass der Bauprozess auf der Systemebene am tatsächlichen Kundennutzen ausgerichtet wird. Mit diesem Ansatz ist es möglich, die Prioritäten der Entwicklungsarbeit mit dem Auftraggebenden auf der Grundlage des Kundennutzens zu besprechen. Das bedeutet,

dass der Auftraggebende und der Product Owner entscheiden können, welcher Kundennutzen wichtiger ist und vom Team zuerst realisiert werden soll.

In unserem Bauprozess werden Epics durch die Gruppierung einer Reihe von User Storys definiert, um einen sinnvollen Kundennutzen zu schaffen (siehe Abschnitt 5.3.2.2). Das bedeutet, dass die Details eines Epics mit Hilfe von User Storys definiert werden.

Hinweise zur Definition des Titels von Epics

Um ein Epic zu identifizieren, brauchen wir einen passenden Titel, der den Kundennutzen widerspiegelt. Wir empfehlen die Verwendung der folgenden Vorlage für den Titel eines Epics:

Als <Kundentyp> möchte ich <eine bestimmte Systemfunktion>, damit <ein bestimmtes Systemziel> erreicht wird.

Abbildung 66 - Beispiel für ein Epic aus der YPRC-Fallstudie zeigt eine beispielhafte Darstellung der YPRC-Fallstudie.

Als <Sportler> möchte ich <meine Trainingsdaten aufzeichnen>, damit <ich meine Leistung nach dem Training analysieren kann>.	
Voraussetzungen	<ul style="list-style-type: none"> Keine
Ziel des Systems	<ul style="list-style-type: none"> G-4 Analysieren der Leistung nach der Trainingseinheit
Szenario	<ul style="list-style-type: none"> Scen-3 Der Läufer führt ein Training durch und überprüft die Daten nach dem Training
Akzeptanzkriterien	<p>Der Sportler kann nach der Trainingseinheit folgende Daten einsehen</p> <ul style="list-style-type: none"> Minimale, durchschnittliche und maximale Pulsfrequenz Mindest-, Durchschnitts- und Höchstgeschwindigkeit Laufstrecke Höchste und niedrigste Höhe Die Daten werden in der App, aber nicht im Portal gespeichert
Relevante Elemente	<ul style="list-style-type: none"> DDev-1 Smartwatch des Läufers DDSys-1 Runner's App

Abbildung 66 - Beispiel für ein Epic aus der YPRC-Fallstudie

Um die Konsistenz zwischen den Epics und den Designkonzepten zu wahren, empfehlen wir die folgenden Leitlinien für den Titel des Epics:

- Der Teil <Kundentyp> sollte sich auf ein Kundensegment aus dem Lösungsdesignkonzept beziehen.
- Der <Systemziel>-Teil des Epics sollte als Ziel im Systemdesignkonzept dokumentiert werden.
- Der Teil <eine Systemfunktion> des Epics sollte als Szenario im Systemdesignkonzept dokumentiert werden.

Die in Abbildung 67 - Beziehung zwischen epischen Konzepten und Designkonzepten visualisierten Referenzen erlauben es, die Aufgabenbeschreibung kompakt zu dokumentieren, da die wichtigen Details des Epics in den Designkonzepten enthalten sind.

Szenario aus Systemdesignkonzepten als Quelle für Akzeptanzkriterien

Ein Szenario eine wichtige Quelle für die Festlegung der *Akzeptanzkriterien* des Epics. Die folgenden Leitlinien sind nützlich für die Festlegung von Akzeptanzkriterien für ein Epic:

- Verwenden Sie Qualitätsanforderungen oder Randbedingungen auf der Systemebene.
- Definieren Sie nur solche Akzeptanzkriterien, deren Erfüllung zu dem im Systemdesignkonzept genannten Ziel beiträgt.
- Nennen Sie beobachtbares Verhalten oder Funktionen, die im Szenario beschrieben werden.
- Verwenden Sie den gleichen Wortlaut im Szenario und in den Akzeptanzkriterien.

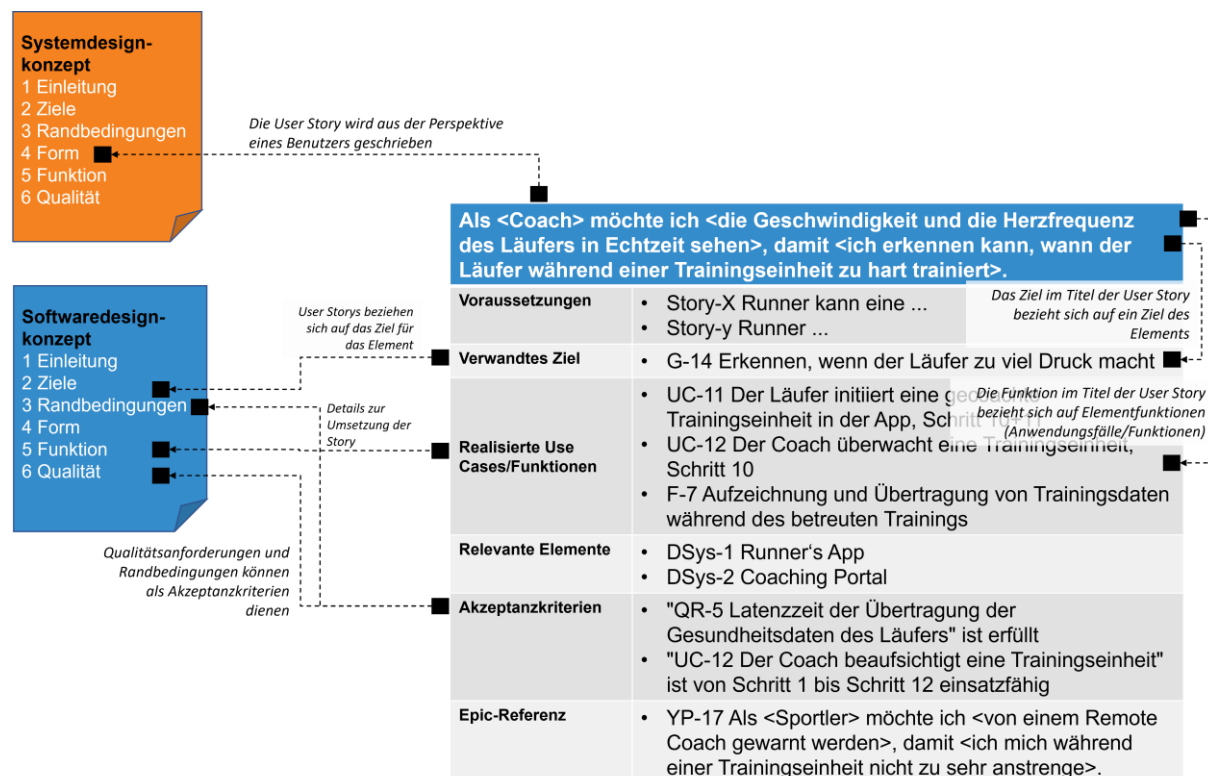


Abbildung 67 - Beziehung zwischen epischen Konzepten und Designkonzepten

Hinweise für die Definition of Ready

Für die Definition of Ready eines Epics empfehlen wir die folgenden Leitlinien:

- Die Epic Beschreibung steht im Einklang mit den Designkonzepten.
- Das Epic wurde durch User Storys verfeinert und die User Storys entsprechen der Definition of Ready.
- Der Auftraggebende ist mit der Änderung der Designkonzepte einverstanden (falls Änderungen erforderlich waren).

Hinweise für die Definition of Done

Für die Definition of Done für ein Epic empfehlen wir die folgenden Leitlinien:

- Die Erfüllung der Akzeptanzkriterien wurde nachgewiesen (z. B. durch spezielle Evaluations-Workitems, siehe unten).
- Alle Arbeitsaufträge im Zusammenhang mit dem Epic entsprechen der Definition of Done.
- Die realisierte digitale Lösung steht im Einklang mit den Designkonzepten.

Praxistipp

Die Dokumentation und Pflege der Beziehungen zwischen Epics und Designkonzepten hat einen wichtigen Vorteil. Sie ermöglicht Ihnen, die Details und die Struktur der Ziele und Szenarien im Systemdesignkonzept zu überdenken. Oft passen die in der Epic-Erstellung definierten Ziele und Funktionen nicht 1:1 zu den Zielen und Szenarien im Systemdesignkonzept. Sie sind oft anders geschnitten oder aus einer anderen Perspektive beschrieben. Dies ist kein Fehler oder Defekt im konzeptuellen Schritt, in dem die Lösungs- und Systemdesignkonzepte ursprünglich erstellt wurden, sondern spiegelt einfach das wachsende Verständnis für die digitale Lösung wider. Wir empfehlen daher, die Ziele und Szenarien im Systemdesignkonzept an den neuen Wissensstand und die Struktur der Epics anzupassen. Auf diese Weise sind die Epics und das Systemdesignkonzept konsistent, und der auf den Epics basierende neue Wissensstand wird auch in den Designkonzepten dokumentiert.

Für Einsteiger im Bereich des Digital Designs scheint diese Arbeit eine Menge Aufwand zu bedeuten. Insbesondere für diejenigen, die täglich mit Epics und Designkonzepten arbeiten, scheint die Aktualisierungsarbeit Zeitverschwendung zu sein, da sie die Details kennen und wissen, was zu tun ist. Dabei sollten Sie bedenken, dass die Designkonzepte das Langzeitgedächtnis des Bauprozesses sind (siehe Abschnitt 2.2.1) und daher auf dem neuesten Stand gehalten werden sollten. Ein weiterer praktischer Hinweis aus unserer Erfahrung ist, dass die richtige Softwareunterstützung der Schlüssel zur Minimierung des Aufwands für die Pflege des Designkonzepts ist. In der YPRC-Fallstudie erläutern wir mögliche Ansätze zur Werkzeugunterstützung in bestimmten Situationen.

5.3.3.5 User Storys in Bezug auf Elementdesignkonzepte

Die User Story dient in unserem Entwicklungsprozess zwei Zwecken. Diese sind:

- Eine kurze und einfache Beschreibung einer Funktion, die ein Benutzer haben möchte
- Ein Management-Werkzeug zur Organisation und Priorisierung der Entwicklungsarbeit auf der Elementebene (siehe Abschnitt 5.3.2.1)

Genau wie bei den Epics ist die Kombination dieser beiden Aspekte (Mischung aus Inhalt und Management) in einer User Story für Einsteiger im Digital Design schwer zu verstehen. Die Stärke dieser Kombination besteht darin, dass der Bauprozess auf der Elementebene entsprechend den tatsächlichen Bedürfnissen der Benutzer gesteuert wird. Diese Art, die Entwicklungsarbeit mit User Storys zu steuern, bedeutet, dass der Nutzen/Wert für den Benutzer während des gesamten Bauprozesses immer im Blick ist.

Hinweise zur Definition des Titels von User Storys

Wir empfehlen, die folgende Vorlage für den Titel einer User Story zu verwenden:

Als <Benutzertyp> möchte ich <eine Funktion>, um <ein Ziel>.

Wie bei den Epics sollten sich die Teile einer User Story auf Designkonzepte beziehen (siehe Abbildung 68 - Beziehung zwischen User Story und Designkonzepten):

- Der Teil <Benutzertyp> sollte sich auf einen im Systemdesignkonzept definierten Benutzertyp beziehen.
- Der Teil <eine Funktion> sollte sich auf eine bestimmte Funktion eines Elements beziehen, die mit dem Erreichen des Ziels zusammenhängt.
- Der Teil <ein Ziel> sollte sich auf ein Benutzerziel beziehen, das für die entsprechende Funktion im Elementdesignkonzept dokumentiert ist.

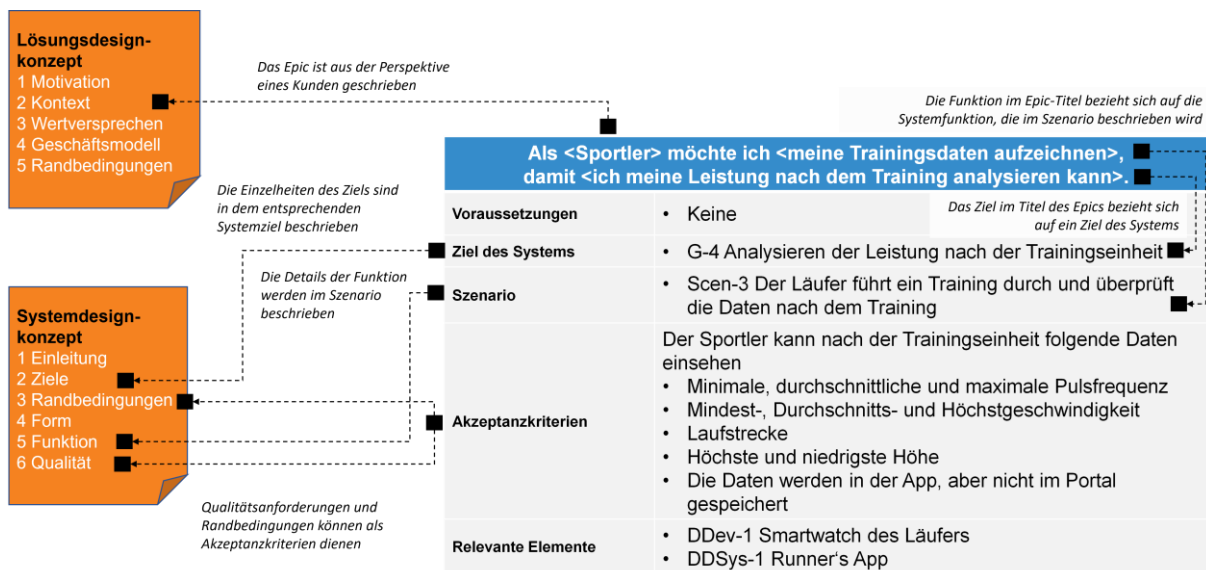


Abbildung 68 - Beziehung zwischen User Story und Designkonzepten

Beispiel YPRC. Die folgende User Story ist ein Beispiel aus der YPRC-Fallstudie (siehe Abbildung 69 zur vollständigen Story):

Als <Lauftainer> möchte ich <die Geschwindigkeit und die Herzfrequenz des Läufers in Echtzeit> sehen, damit <ich erkennen kann, wann der Läufer während einer Trainingseinheit zu hart trainiert>.

Als <Coach> möchte ich <die Geschwindigkeit und die Herzfrequenz des Läufers in Echtzeit sehen>, damit <ich erkennen kann, wann der Läufer während einer Trainingseinheit zu hart trainiert>.

Voraussetzungen	<ul style="list-style-type: none"> • Story-X Runner kann eine ... • Story-Y Runner ...
Verwandtes Ziel	<ul style="list-style-type: none"> • G-14 Erkennen, wenn der Läufer zu viel Druck macht
Realisierte Use Cases/Funktionen	<ul style="list-style-type: none"> • UC-11 Der Läufer initiiert eine gecoachte Trainingseinheit in der App, Schritt 10+11 • UC-12 Der Coach überwacht eine Trainingseinheit Schritt 10 • F-7 Aufzeichnung und Übertragung von Trainingsdaten während des betreuten Trainings
Relevante Elemente	<ul style="list-style-type: none"> • DSys-1 Runner's App • DSys-2 Coaching Portal
Akzeptanzkriterien	<ul style="list-style-type: none"> • "QR-5 Latenzzeit der Übertragung der Gesundheitsdaten des Läufers" ist erfüllt • "UC-12 Der Coach beaufsichtigt eine Trainingseinheit" ist von Schritt 1 bis Schritt 12 einsatzfähig
Epic-Referenz	<ul style="list-style-type: none"> • YP-17 Als <Sportler> möchte ich <von einem Remote Coach gewarnt werden>, damit <ich mich während einer Trainingseinheit nicht zu sehr anstreng>

Abbildung 69 - Beispiel für eine User Story aus der YPRC-Fallstudie

Hinweise zur Aufgabenbeschreibung

Mit dem Titel entsteht eine erste Vorstellung von einem Teil der digitalen Lösung. Für die eigentliche Entwicklungsarbeit sind jedoch weitere Details erforderlich, die in einer Aufgabenbeschreibung aufgeführt sind. Wir empfehlen, in der User Story eine sehr kurze Erklärung abzugeben, die für weitere Details auf die Elementdesignkonzepte verweist. Die Beschreibung sollte Informationen über die realisierten Use Cases (Schritte)/Funktionen, die Referenzen zu den Use Cases/Funktionen und die relevanten Use Case-Schritte oder (Teil-)Funktionen, die durch diese User Story realisiert werden, enthalten

Die wichtigsten Bezugspunkte für die Aufgabenbeschreibung von User Storys sind Ziele, Use Cases und Funktionen im Elementdesignkonzept. Der Grund dafür ist, dass die Realisierung einer User Story einen Mehrwert für den Benutzer bieten muss, der durch das Ziel definiert ist. Der Mehrwert in Bezug auf die Elementdesignkonzepte ist die wahrnehmbare oder zugrundeliegende Funktion, die mit Hilfe von Use Cases und Funktionen beschrieben wird. Es ist wichtig, sich vor Augen zu halten, dass die Beziehung zwischen den User Storys und den Designkonzepten in beide Richtungen geht. Während der Ausarbeitung einer User Story und der Diskussion mit den Umsetzungsteams ergeben sich neue Erkenntnisse, die in die entsprechenden Designkonzepte und User Storys einfließen müssen. Dies führt zu einer regelmäßigen Aktualisierung der Elementdesignkonzepte.

Die Beziehung zu anderen Aspekten der Form, die in den Elementdesignkonzepten definiert sind (z. B. User Interfaces, Software-Schnittstellen oder Entitäten), wird durch die Verweise von Use Cases/Funktionen auf diese Elemente definiert (konstruktive Beziehungen, siehe Abschnitt 2.2.6). Diese Referenzen müssen nicht in die User Storys eingeführt werden, da das gesamte Umsetzungsteam mit den Elementdesignkonzepten arbeitet und diese Informationen aus den Elementdesignkonzepten entnehmen kann.

Dies bedeutet nicht, dass die tatsächliche Umsetzung von Formaspekten einer digitalen Lösung (z. B. Software-Schnittstellen) nicht Teil des Backlogs ist. Zu diesem Zweck stellen wir das technische Workitem vor (siehe Abschnitt 5.3.3.9).

Hinweise zu den Akzeptanzkriterien

Akzeptanzkriterien in User Storys beschreiben Bedingungen, die vom Umsetzungsteam nachgewiesen werden müssen, wenn die User Story implementiert wurde. Sie dienen auch als abstrakte Liste von Testfällen, anhand derer der Product Owner feststellen kann, ob die User Story richtig implementiert wurde. Gute Kandidaten für Akzeptanzkriterien sind Qualitätsanforderungen und Randbedingungen aus den System- und Elementdesignkonzepten.

Andere gute Quellen für Akzeptanzkriterien sind Verweise auf Use Cases, die funktionieren müssen, wenn die User Story implementiert wurde. Das Beispiel in Abbildung 69 erwähnt den Use Case UC-12 mit den Schritten 1 bis 12 als Teil der Akzeptanzkriterien. Das bedeutet, dass nach der Umsetzung dieser User Story (die sich auf Schritt 10 konzentriert), der Coaching-Prozess von Schritt 1 bis 12 funktionieren muss. Dies ist möglich, da die anderen Schritte bereits durch vorangegangene User Storys umgesetzt wurden.

Die Epic Referenz der User Story wird verwendet, um das Epic anzugeben, zu dem die Realisierung der User Story beiträgt. Durch den Epic-Bezug wird ein indirekter Bezug zum Systemdesignkonzept hergestellt, da das Epic auf ein Ziel aus dem Systemdesignkonzept verweist. Aus Gründen der Konsistenz sollte das Systemziel auch durch das Elementziel referenziert werden, auf das sich die User Story bezieht.

Hinweise für die Definition of Ready

Eine gute Praxis für die *Definition of Ready* für User Storys ist das INVEST-Akronym [Wake2003]:

- I - Independent (unabhängig): User Storys sollten so unabhängig wie möglich voneinander sein. Dies erleichtert die parallele Durchführung in einem oder mehreren Teams.
- N - Negotiable (verhandelbar): Eine User Story stellt keinen festen Vertrag dar - sie lässt Raum für Diskussionen über die Details.
- V - Valuable (wertvoll): Eine User Story bringt dem Benutzer/Kunden einen echten Wert.
- E - Estimable (schätzbar): Die Beschreibung ist so detailliert, dass der Realisierungsaufwand richtig abgeschätzt werden kann.
- S - Small (klein): Eine User Story ist so klein, dass sie in einer Iteration umgesetzt werden kann.
- T - Testable (testbar): Der Detaillierungsgrad ist ausreichend, um eine Prüfung der Implementierung zu ermöglichen.

Diese Kriterien zu erfüllen, ist eine echte Herausforderung für Einsteiger im Digital Design. Das richtige Maß an Unabhängigkeit, Größe und Detailgenauigkeit zu finden, erfordert die Zusammenarbeit mit dem Umsetzungsteam und ist ein iterativer Prozess, der zwischen Product Owner und Umsetzungsteam hin und her geht.

Hinweise für die Definition of Done

Die *Definition of Done* für eine User Story sollte Kriterien aus der Design-, Konstruktions- und Realisierungsperspektive beinhalten. Aus der Sicht des Designs sollten die Kriterien für die Erledigung einige Qualitätssicherungsmaßnahmen enthalten, die zeigen, dass die Akzeptanzkriterien erfüllt sind (z. B. Testfälle oder explizite Verweise auf Use Case Schritte, siehe Abbildung 69). Die Kriterien für die Durchführung aus Sicht der Konstruktion und der Realisierung werden von Fachleuten aus diesen Bereichen festgelegt. Sie können die Durchführung von Unit Tests oder die Dokumentation technischer Details (z. B. Architekturdokumentation) umfassen.

5.3.3.6 Konzept-Workitems für die Arbeit an Designkonzepten

Konzept-Workitems werden verwendet, um Konzeptarbeit zu planen, die für zukünftige Iterationen und für die Definition of Ready von User Storys und anderen Arbeitsaufträgen notwendig ist. Typische konzeptuelle Aufgaben sind:

- Ausarbeitung neuer Designs für User Interfaces
- Ausarbeitung von neuen Entitäten
- Ausarbeitung von Use Cases
- Verfeinerung bestehender Designkonzepte

Konzept-Workitems beziehen sich daher auf den oder die Bausteine des zu erarbeitenden Designkonzepts und auf andere Bausteine, die sich als nützlicher Input für das Konzept-Workitem erweisen (siehe Abbildung 70 - Beziehung zwischen Konzept-Workitems und Designkonzepten).

Der Titel bezieht sich auf die Bausteine, die ausgearbeitet werden müssen

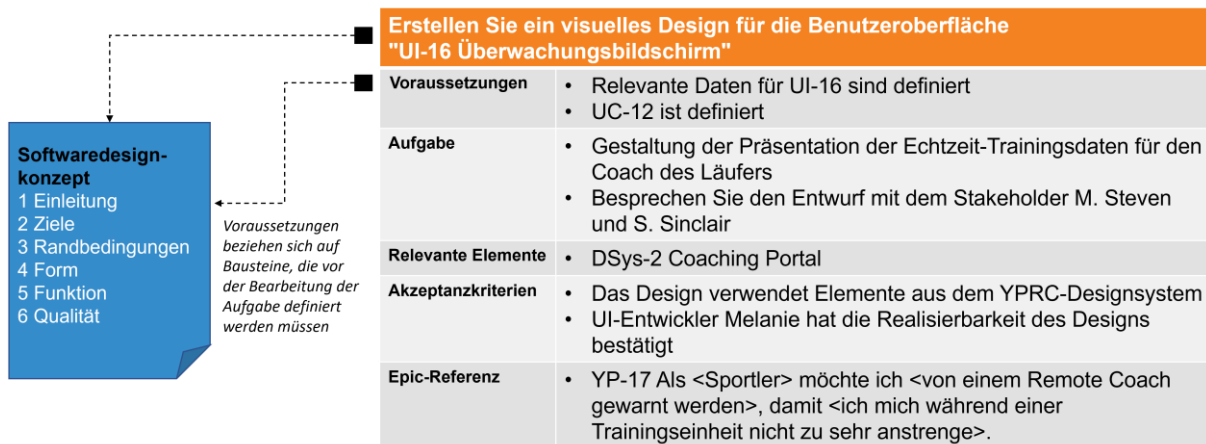


Abbildung 70 - Beziehung zwischen Konzept-Workitems und Designkonzepten

Hinweise zur Definition des Titels und der Aufgabenbeschreibung

Der Titel eines Konzept-Workitems sollte eine kurze Zusammenfassung der Aufgabe enthalten - wenn möglich, sollte das Konzept-Workitem bereits auf den oder die zu erarbeitenden Bausteine verweisen. Das unter Abbildung 71 - Beispiel für eine Konzept-Workitem aus der YPRC-Fallstudie gezeigte Beispiel bezieht sich auf das User Interface UI-16, das ausgearbeitet werden soll. Das Beispiel ist ein sehr feingranulares Konzept-Workitem (die Gestaltung eines bestimmten Bildschirms des User Interface).

Erstellen Sie ein visuelles Design für die Benutzeroberfläche "UI-16 Überwachungsbildschirm"	
Voraussetzungen	<ul style="list-style-type: none"> Relevante Daten für UI-16 sind definiert UC-12 ist definiert
Aufgabe	<ul style="list-style-type: none"> Gestaltung der Präsentation der Echtzeit-Trainingsdaten für den Coach des Läufers Besprechen Sie den Entwurf mit dem Stakeholder M. Steven und S. Sinclair
Relevante Elemente	<ul style="list-style-type: none"> DSys-2 Coaching Portal
Akzeptanzkriterien	<ul style="list-style-type: none"> Das Design verwendet Elemente aus dem YPRC-Designsystem. UI-Entwickler Melanie hat die Realisierbarkeit des Designs bestätigt
Epic-Referenz	<ul style="list-style-type: none"> YP-17 Als <Sportler> möchte ich <von einem Remote Coach gewarnt werden>, damit <ich mich während einer Trainingseinheit nicht zu sehr anstreng>.

Abbildung 71 - Beispiel für eine Konzept-Workitem aus der YPRC-Fallstudie

Konzept-Workitems können natürlich auch mit einem viel breiteren Umfang definiert werden (z. B. einen ganzen Prozess ausarbeiten). Beachten Sie jedoch, dass unser Bauprozess davon ausgeht, dass die Arbeit an den Konzept-Workitems im Rahmen der Iteration gemeinsam mit dem gesamten Umsetzungsteam erfolgt. Wir empfehlen daher, die Konzept-Workitems unter diesem Gesichtspunkt zu definieren. Einsteiger sollten mit der Definition eher

feingranularer Aufgaben beginnen, um ein Gefühl dafür zu bekommen, wie viel Arbeit in einem Konzept-Workitem erledigt werden kann.

Hinweise zu den Akzeptanzkriterien

Die Akzeptanzkriterien für ein Konzept-Workitem beziehen sich immer auf die erstellten Designkonzepte. Gute Kandidaten für Akzeptanzkriterien sind Überprüfungen durch Teammitglieder (z. B. aus technischer Sicht) oder durch relevante Stakeholder.

Hinweise für die Definition of Ready

Die Definition of Ready für ein Konzept-Workitem wird hauptsächlich durch die für die Aufgabe definierten Voraussetzungen bestimmt. Denken Sie daran, dass es nicht notwendig ist, die Definition of Ready unmittelbar nach der Erstellung des Konzept-Workitems zu erfüllen. Die Definition of Ready muss erfüllt sein, wenn die Arbeit an der Aufgabe kurz vor dem Beginn steht.

Hinweise für die Definition of Done

Die Definition of Done für ein Konzept-Workitem kann folgende Aspekte umfassen:

- Der Product Owner hat das erarbeitete Konzept geprüft und akzeptiert.
- Alle Verfolgbarkeitsbeziehungen zwischen dem neuen Konzeptteil und den bestehenden Teilen wurden überprüft.
- Die Auswirkungen des neuen Inhalts auf bereits definierte User Storys wurden geprüft.
- Die Erstellung neuer User Storys, die den Inhalt widerspiegeln, wurde überprüft.
- Prüfung auf notwendige Aktualisierungen des Systemdesignkonzepts, des Lösungsdesignkonzepts oder sogar des Digital Design Briefs (falls erforderlich) ist erfolgt.

Mit diesen Aspekten kann die Definition of Done für Konzept-Workitems genutzt werden, um qualitativ hochwertige Designkonzepte zu erhalten und einen gemeinsamen Standard für die Arbeit mit diesen Designkonzepten zu definieren.

5.3.3.7 Prototyp-Workitem

Prototyp-Workitems werden verwendet, um die Prototyping-Arbeiten als Teil der Iteration zu verwalten. Diese Art der Arbeit mit Prototyp-Workitems folgt der gleichen Idee, die für Konzept-Workitems eingeführt wurde.

Hinweise zur Definition des Titels und der Aufgabenbeschreibung

Wenn bei der Entwicklung einer digitalen Lösung die Erstellung von Prototypen erforderlich ist, muss die Arbeit an den Prototypen auf die anderen Aufgaben abgestimmt werden. Denken Sie daran, dass das Prototyping ein leistungsfähiges Werkzeug ist (siehe Abschnitt 2.3), das Ressourcen spart, da es die Entwicklung der Lösung in die falsche Richtung verhindern kann. Durch die Verwendung von Prototyp-Workitems verstehen der Product Owner und das Team die Auswirkungen auf den Fortschritt der Entwicklungsarbeit, da die Arbeit an einem Prototyp die Ressourcen des Umsetzungsteams verbraucht, aber langfristig Ressourcen einsparen kann.

Erstellen Sie ein interaktives High-Fidelity-Mock-up einer graphischen Visualisierung der Daten des Läufers, die die aktuellen Daten und den Verlauf während der Trainingseinheit zeigt

Voraussetzungen	<ul style="list-style-type: none"> • Drei Datensätze von Laufeinheiten sind verfügbar (mindestens 5 Minuten pro Trainingsdatensatz)
Aufgabe	<ul style="list-style-type: none"> • Einigung auf einen technischen Rahmen für die graphische Visualisierung zusammen mit dem Umsetzungsteam • Entwurf einer grafischen Visualisierung des Trainingsdatenflusses • Realisierung einer Demonstration, die die drei Datensätze in Echtzeit verwendet
Relevante Elemente	<ul style="list-style-type: none"> • DSys-2 Coaching Portal
Akzeptanzkriterien	<ul style="list-style-type: none"> • Die drei Datensätze werden in einer Demonstration in Echtzeit visualisiert
Epic-Referenz	<ul style="list-style-type: none"> • YP-17 Als <Sportler> möchte ich <von einem Remote Coach gewarnt werden>, damit <ich mich während einer Trainingseinheit nicht zu sehr anstreuge>.

Abbildung 72 - Beispiel eines Prototyp-Workitems aus der YPRC-Fallstudie

Abbildung 72 - Beispiel eines Prototyp-Workitems aus der YPRC-Fallstudie zeigt ein beispielhaftes Prototyp-Workitem aus der YPRC-Fallstudie. Das Beispiel stellt einen Prototyp eines User Interface vor, der eine alternative Visualisierungsidee für die Daten des Läufers für den Trainer konkretisiert, um zu verstehen, ob der graphbasierte Visualisierungsansatz für den Trainer wertvoller ist als der textbasierte. Dieser Prototyp wird von einem Designexperten (hier wahrscheinlich für Interaktionsdesign) aus dem Team erstellt. Die Arbeitsaufträge des Prototyps hängen jedoch von der jeweiligen Zielsetzung des Prototyps ab (siehe Abschnitt 2.3.2). Es kann sich auch um funktionale Prototypen handeln, die sich auf die Machbarkeit einer Funktion einer digitalen Lösung konzentrieren. Solche Prototypen werden natürlich von den Konstruktions-/Realisierungsexperten des Umsetzungsteams erstellt.

Hinweise zu den Akzeptanzkriterien

Akzeptanzkriterien für einen Prototyp sollten sich auf die Qualitätsanforderungen beziehen, die der Prototyp erfüllen sollte. Beachten Sie, dass die Qualitätsanforderungen an einen Prototyp je nach Zielsetzung des Prototyps unterschiedlich sein können. Die in Abschnitt 2.3 vorgestellten Ziele und Kategorien von Prototypen bieten gute Leitlinien für die Definition geeigneter Qualitätsanforderungen für ein Prototyp-Workitem.

Zusätzlich zu den Qualitätsanforderungen sollten die Ergebnisse des Prototyping (entsprechend den im Prototyping-Workitem definierten Aufgaben) von Teammitgliedern (z. B. aus technischer Sicht) oder von relevanten Stakeholdern überprüft werden.

Hinweise für die Definition of Ready und die Definition of Done

Die Definition of Ready und die Definition of Done hängen von dem konkreten Prototyp-Workitem ab. Daher ist es schwierig, allgemeine Regeln für beide zu definieren. Als Faustregel gilt, dass die Voraussetzungen so formuliert werden sollten, dass sie die Bereitschaft zur Bearbeitung des Prototyp-Workitems definieren. Die Definition of Done sollte die Erfüllung der Akzeptanzkriterien einschließen.

5.3.3.8 Evaluations-Workitems

Evaluations-Workitems werden verwendet, um die Evaluierung eines bestimmten Aspekts der digitalen Lösung, z. B. ein Usability-Test von Teilen der Lösung oder eines Prototyps als Teil der Iterationsarbeit zu verwalten. Die Evaluations-Workitems konzentrieren sich auf spezielle Fragen und ersetzen keine systematischen Qualitätssicherungsmaßnahmen der gesamten

digitalen Lösung wie Systemtests oder Benutzerakzeptanztests. Diese Tests müssen unabhängig voneinander geplant werden. Die Verfahren sollten in dem entsprechenden Evaluationskonzept dokumentiert werden.

Evaluierung der graphischen Visualisierung der Daten des Läufers mit den Stakeholdern X, Y und Z, um über die Umsetzung zu entscheiden

Voraussetzungen	<ul style="list-style-type: none"> • Prototyp einer graphischen Visualisierung ist verfügbar
Aufgabe	<ul style="list-style-type: none"> • Planen Sie einen kurzen Usability-Test für den Prototyp • Besuchen Sie X, Y und Z und präsentieren Sie den Prototyp, führen Sie den Usability-Test durch • Laden Sie X, Y und Z zu einem gemeinsamen Workshop mit dem PO ein, um die Ergebnisse des Usability-Tests zu diskutieren und über die Umsetzung des Prototyps zu entscheiden
Relevante Elemente	<ul style="list-style-type: none"> • DSys-2 Coaching Portal
Akzeptanzkriterien	<ul style="list-style-type: none"> • Der Workshop gibt eine klare Empfehlung für oder gegen die Implementierung der graphischen Visualisierung
Epic-Referenz	<ul style="list-style-type: none"> • YP-17 Als <Sportler> möchte ich <von einem Remote Coach gewarnt werden>, damit <ich mich während einer Trainingseinheit nicht zu sehr anstreng>.

Abbildung 73 - Beispiel für ein Evaluations-Workitem aus der YPRC-Fallstudie

Abbildung 73 - Beispiel für ein Evaluations-Workitem aus der YPRC-Fallstudie zeigt ein beispielhaftes Evaluations-Workitem aus der YPRC-Fallstudie. Das Beispiel zeigt das Evaluations-Workitem, das anschließend auf die Erstellung des Prototyps des User Interface von Abbildung 72 - Beispiel eines Prototyp-Workitem aus der YPRC-Fallstudie folgt.

Hinweise zur Aufgabenbeschreibung

Die Details eines Evaluations-Workitem sollten sich im Sinne eines Evaluationskonzepts auf das Evaluationskonzept des Elements beziehen. Wir empfehlen, wichtige Evaluations-Workitem (z. B. Usability-Tests oder detaillierte User Acceptance Tests) nicht nur in Form von Backlog-Items zu dokumentieren, sondern auch als Teil des Evaluationskonzepts der Lösung, des Systems oder des Elements. Ein Systemtestfall, der verschiedene Elemente des Systems umfasst, gehört beispielsweise in den Systemevaluationstest, und ein detaillierter Usability-Test eines Elements gehört in das betreffende Elementevaluationskonzept.

Hinweise zu den Akzeptanzkriterien

Akzeptanzkriterien für Evaluations-Workitem haben immer einen anderen Schwerpunkt als andere Arbeitsaufträge, da sich die Evaluations-Workitem auf Erkenntnisse über ein bestehendes Artefakt konzentrieren. Wir empfehlen daher, Akzeptanzkriterien für Evaluations-Workitem auf der Grundlage der erwarteten Ergebnisqualität der Evaluationsaufgabe zu definieren. Zum Beispiel fordert Abbildung 73 - Beispiel für ein Evaluations-Workitem aus der YPRC-Fallstudie eine klare Empfehlung für oder gegen die Implementierung der graphbasierten Visualisierung.

Hinweise für die Definition of Ready und die Definition of Done

Die Definition of Ready und die Definition of Done hängen von dem konkreten Evaluations-Workitem ab. Daher ist es schwierig, allgemeine Regeln für beide zu definieren. Als Faustregel gilt, dass die Voraussetzungen so formuliert werden sollten, dass sie die Bereitschaft zur Durchführung der Evaluierung definieren (z. B. erforderliche Testdaten vorhanden). Die Definition of Done sollte die Erfüllung der Akzeptanzkriterien einschließen.

5.3.3.9 Technische Workitems

Technische Workitems gehören in den Bereich der Konstruktion und Realisierung. Wir stellen sie daher hier nur kurz vor.

Beispiele für technische Workitems sind:

- Ausarbeitung eines Realisierungskonzepts (oder eines Teils davon)
- Implementierung von technischen Schnittstellen
- Implementierung von Datenstrukturen
- Überarbeitung der bereits implementierten Softwareelemente

Technische Workitems dienen dazu, die Entwicklungsarbeit zu verwalten, die als Voraussetzung für die Umsetzung einer User Story notwendig ist. Obwohl technische Workitems nicht zum Kernbereich des Digital Designs gehören, können sie Beziehungen zu Designkonzepten haben (siehe Abbildung 74 - Beziehung zwischen technischen Workitems und Designkonzepten).

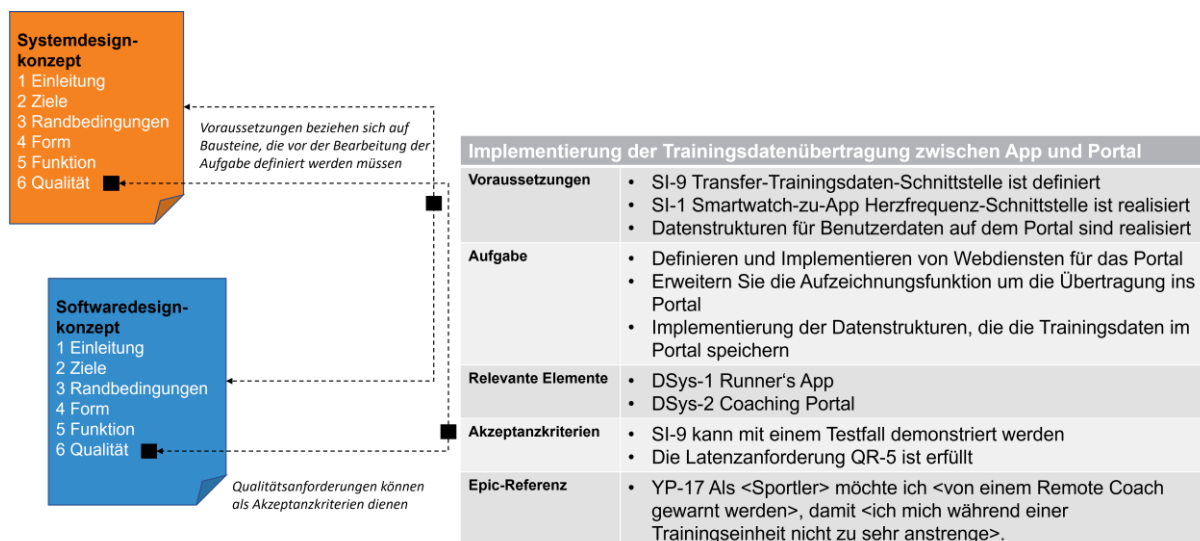


Abbildung 74 - Beziehung zwischen technischen Workitems und Designkonzepten

Hinweise zu den Akzeptanzkriterien

Zum Beispiel können sich die Akzeptanzkriterien eines technischen Workitems auf Qualitätsanforderungen auf System- oder Elementebene oder auf Testfälle beziehen, die das Ergebnis auf technischer Ebene demonstrieren. Die Voraussetzungen eines technischen Workitems können sich auf Bausteine von Designkonzepten als Input für das technische Workitem beziehen.

Implementierung der Trainingsdatenübertragung zwischen App und Portal	
Voraussetzungen	<ul style="list-style-type: none"> • SI-9 Transfer-Trainingsdaten-Schnittstelle ist definiert • SI-1 Smartwatch-zu-App Herzfrequenz-Schnittstelle ist realisiert • Datenstrukturen für Benutzerdaten auf dem Portal sind realisiert
Aufgabe	<ul style="list-style-type: none"> • Definieren und Implementieren von Webdiensten für das Portal • Erweitern Sie die Aufzeichnungsfunktion um die Übertragung ins Portal • Implementierung der Datenstrukturen, die die Trainingsdaten im Portal speichern
Relevante Elemente	<ul style="list-style-type: none"> • DSys-1 Runner's App • DSys-2 Coaching Portal
Akzeptanzkriterien	<ul style="list-style-type: none"> • SI-9 kann mit einem Testfall demonstriert werden • Die Latenzanforderung QR-5 ist erfüllt
Epic-Referenz	<ul style="list-style-type: none"> • YP-17 Als <Sportler> möchte ich <von einem Remote Coach gewarnt werden>, damit <ich mich während einer Trainingseinheit nicht zu sehr anstrenge>

Abbildung 75 - Beispiel für ein technisches Workitem aus der YPRC-Fallstudie

Darüber hinaus erfordert die Rolle des Product Owners ein grundlegendes Verständnis der technischen Workitems, um eine effektive Kommunikation mit den Teammitgliedern zu ermöglichen, die an der Konstruktion und Realisierung der digitalen Lösung arbeiten. Abbildung 75 - Beispiel für ein technisches Workitem aus der YPRC-Fallstudie zeigt ein Beispiel für ein technisches Workitem, das sich mit der Implementierung einer bestimmten Schnittstelle beschäftigt.

Hinweise für die Definition of Ready und die Definition of Done

Die Definition of Ready und die Definition of Done müssen gemeinsam mit den Fachleuten für Bau und Realisierung festgelegt werden. Die Definition of Done einer technischen Aufgabe umfasst einige Qualitätssicherungsmaßnahmen, die zeigen, dass die Akzeptanzkriterien erfüllt sind (z. B. durch eine Reihe von Testfällen). Darüber hinaus empfehlen wir, wo immer möglich, Verweise auf die Designkonzepte in den technischen Workitems als Referenzpunkte zu verwenden. Diese Arbeitsweise unterstützt die Abstimmung der technischen Arbeit mit den Designkonzepten.

5.3.3.10 Niederschreiben von Defekten

Unabhängig vom spezifischen Entwicklungsansatz werden in der digitalen Lösung unweigerlich Defekte (Defects) festgestellt werden. Wir empfehlen, Defekte als bestimmten Typ eines Arbeitsauftrags zu verwalten. Abbildung 76 - Beispiel für ein Defekt-Workitem aus der YPRC-Fallstudie zeigt ein Beispiel aus der YPRC-Fallstudie, das als Vorlage dienen kann.

Pulsdaten über 255 führen dazu, dass die Runner's App einfriert	
Voraussetzungen	<ul style="list-style-type: none"> Die Smartwatch misst einen Puls von über 255 während einer Trainingseinheit
Beschreibung des Defekts	<ul style="list-style-type: none"> Die Smartwatch des Läufers zeigt eine Pulsfrequenz von über 255 an (Fehlfunktion des Sensors?) Der Trainingsbildschirm der Läufer-App (UI-5) zeigt einen leeren Pulswert und friert ein Der Läufer muss die App beenden und neu starten, um das Training fortzusetzen. Die zuvor aufgezeichneten Trainingsdaten gehen verloren
Betroffene Elemente	<ul style="list-style-type: none"> DSys-1 Runner's App UI-5 Trainingsstatus-Bildschirm
Erwartetes Verhalten	<ul style="list-style-type: none"> Die App sollte diesen außergewöhnlichen Pulswert ohne Einfrieren anzeigen und aufzeichnen
Epic-Referenz	<ul style="list-style-type: none"> YP-17 Als <Sportler> möchte ich <von einem Remote Coach gewarnt werden>, damit <ich mich während einer Trainingseinheit nicht zu sehr anstreuge>

Abbildung 76 - Beispiel für ein Defekt-Workitem aus der YPRC-Fallstudie

Hinweise zur Definition des Titels und der Beschreibung

Der Titel sollte eine kurze Beschreibung des Defekts enthalten, um eine Vorstellung von dem Defekt zu vermitteln. Die Beschreibung besteht aus Voraussetzungen, die notwendig sind, um den Defekt zu reproduzieren. Diese Informationen sind besonders wichtig, um die Ursache des Defekts zu verstehen und zu analysieren. Außerdem sollte die Beschreibung des Defekts eine verständliche Erklärung des Defekts enthalten. Der Teil *Betroffene Elemente* sollte einige Hinweise auf Designkonzepte enthalten, die mit dem Defekt in Zusammenhang stehen könnten.

Praxistipp

Im obigen Beispiel friert ein Bildschirm der Läufer-App ein. Deshalb werden die App und das konkrete User Interface erwähnt. Wie die Voraussetzungen können auch diese Informationen die Analyse des Defekts unterstützen. Der letzte Teil Erwartetes Verhalten enthält eine kurze Beschreibung des Verhaltens, das anstelle des Defekts zu erwarten wäre. Diese Informationen sind notwendig, um einen Hinweis darauf zu geben, wie der Defekt behoben werden sollte.

Es müssen keine zusätzlichen Aufgaben für Defekte erstellt werden, die leicht behoben werden können. Stellt sich bei der Analyse eines Defekts jedoch heraus, dass es sich um einen komplexen Defekt handelt, werden aus der Analyse neue Workitems (z. B. ein technisches Workitem oder eine User Story) erstellt.

5.3.4 Phase 1: Erste Release-Planung und Backlog-Vorbereitung

Bevor mit der eigentlichen Entwicklung begonnen werden kann, sind eine erste Releaseliste und erste Backlogs erforderlich. Die Ausgangspunkte für diese Arbeit sind das Lösungsdesignkonzept und das Systemdesignkonzept, die im konzeptuellen Schritt erstellt wurden (siehe Abschnitt 5.2). Ein naiver Ansatz für Phase 1 wäre eine vollständige Ausarbeitung der in Abschnitt 2.2.2 vorgestellten Elementdesignkonzepte. Dieser ist nicht sinnvoll, da die detaillierte Erstellung von Elementdesignkonzepten einen hohen Aufwand erfordert und den Beginn der Umsetzung verzögert.

Unserer Erfahrung nach ist das folgende Verfahren für Einsteiger im Digital Design praktischer:

- 1) Ausarbeitung von Element-Design-Canvases für jedes Element, das im Systemdesignkonzept definiert ist.
- 2) Ausarbeitung einer Story Map mit User Storys.
- 3) Priorisierung und Gruppierung der User Storys in Epics, die einen erkennbaren und realisierbaren Kundennutzen definieren.
- 4) Erarbeitung aller Arbeitsaufträge, die notwendig sind, um die Definition of Ready für alle User Storys der 2-3 relevantesten Epics zu erreichen.

Durch dieses Vorgehen sollte ein Elementdesign-Backlog mit einer ausreichenden Menge an Arbeit für ein typisches Umsetzungsteam mit 6-7 Personen entstehen. Weitere Epics können parallel zu den anderen Entwicklungsarbeiten ausgearbeitet werden.

5.3.4.1 Erarbeitung eines Element-Design-Canvases für jedes Element

Ähnlich wie die Arbeit mit dem Wertversprechen und dem Business Model Canvas im konzeptuellen Schritt, kann die Arbeit an den Elementdesignkonzepten in einer canvas-orientierten Arbeitsweise beginnen. Abbildung 77 zeigt eine einfache Struktur für ein solches Canvas.

Hinweise für die Arbeit an Element-Design-Canvases

Das Canvas ist ein temporäres Artefakt und dient der Strukturierung und Diskussion der Ideen für ein bestimmtes Element. Einsteigern empfehlen wir, die Elemente auf dem Canvas in der folgenden Reihenfolge zu definieren:

1. Definieren Sie die Hauptziele, die das Element erreichen soll.
2. Definieren Sie die Ausgangsfunktion des Elements:
 - Benennen Sie Use Cases (skizzieren Sie das Hauptszenario und kritische Alternativszenarien, falls der Use Case unklar ist).
 - Benennen Sie die Funktionen, die für die Use Cases erforderlich sind (skizzieren Sie Funktionsdetails, wenn die Funktion nicht aus dem Namen hervorgeht).
3. Definieren Sie die ursprüngliche Form des Elements auf der Grundlage der Use Cases und Funktionen:
 - Benennen Sie Datenstrukturen in Form von Entitäten (Attribute werden während der Sprints erstellt).
 - Benennen Sie wichtige User Interfaces (falls zutreffend).
 - Benennen Sie wichtige technische Schnittstellen (falls zutreffend).
 - Benennen Sie Teile eines Geräts (falls zutreffend).
4. Visualisieren Sie wichtige Datenflüsse zwischen Schnittstellen, Use Cases, Funktionen und Entitäten.
5. Erstellen Sie eine erste Liste von Qualitätsanforderungen und Randbedingungen für das Element.
6. Iterieren Sie alle Elemente, bis sich das Team auf ein erstes gemeinsames Verständnis des Elements geeinigt hat, und beziehen Sie das Lösungs- und Systemdesignkonzept als Referenzpunkt mit ein (z. B. vergleichen Sie das Element mit den vorgeschlagenen Werten aus dem Value Proposition Canvas).

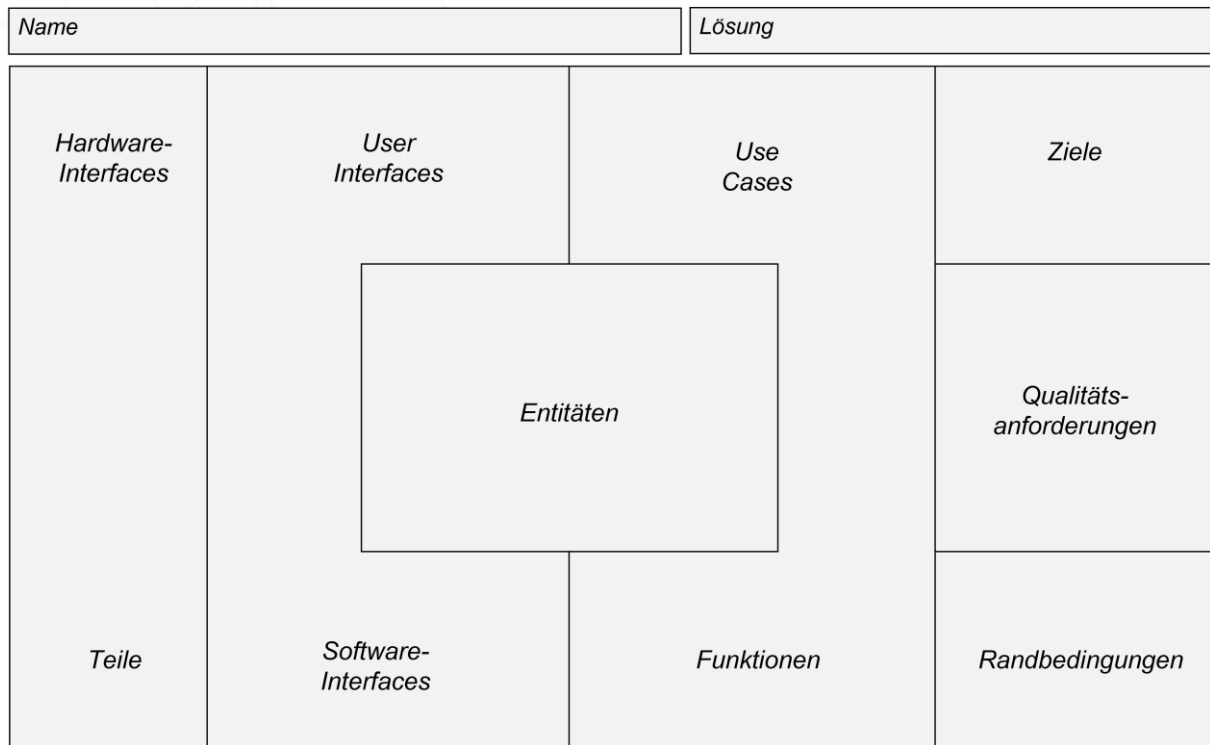


Abbildung 77 - Eine einfache Struktur für ein Element-Design-Canvas

Es ist wichtig, sich darüber im Klaren zu sein, dass ein solches Canvas keine vollständige und konsistente Beschreibung des betrachteten Elements sein soll; es ist ein Ausgangspunkt für die Diskussion.

Weitere Details (z. B. Attribute der Entitäten, detaillierte Gestaltung eines User Interface) werden im Rahmen der Sprints und während der Ausarbeitung der User Storys definiert. Mit dieser Arbeit wird im Rahmen der Sprintarbeit ein vollständig detailliertes Elementdesignkonzept erstellt (siehe unten).

Wir empfehlen, dass das gesamte Umsetzungsteam zusammen mit dem Product Owner am Element-Design-Canvas arbeitet. Es werden weitere Realisierungskonzepte erstellt (z. B. eine erste technische Architektur). Ihre Struktur und ihr Inhalt hängen von den technischen Details der digitalen Lösung ab. Wichtig dabei ist, dass der Product Owner zusammen mit dem Umsetzungsteam das Backlog und alle anderen notwendigen Grundlagen für den Beginn der Entwicklungsarbeit vorbereitet (z. B. Einrichten von Entwicklungsumgebungen).

Beispiel YPRC. Abbildung 78 zeigt ein beispielhaftes Canvas für die Läufer-App aus der YPRC-Fallstudie. Auf den ersten Blick wirkt das Canvas recht chaotisch und verwirrend, vor allem wegen der vielen Elemente und Verbindungen. Wie bei jeder Canvas-Technik ist jedoch zu beachten, dass das Canvas ein Denkwerkzeug für eine Gruppe von Personen ist, die das Canvas gemeinsam im Workshop-Modus erstellen und bearbeiten. Ein genauerer Blick auf das Canvas offenbart wichtige Strukturen für die App. Zum Beispiel ist die Entität „Trainingsdaten“ in verschiedene Funktionen integriert und der „Internetzugang“ ist eine zentrale Schnittstelle, um die Ziele der App zu erfüllen.

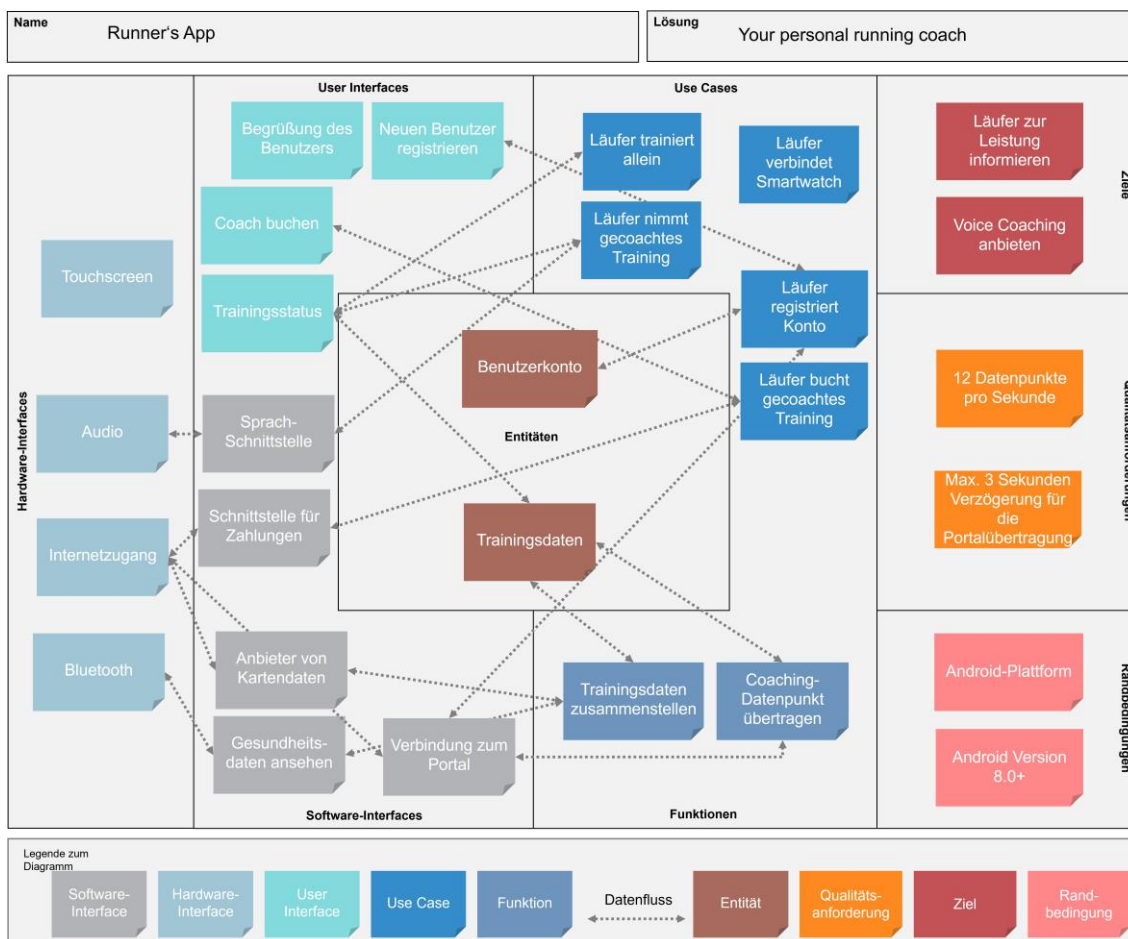


Abbildung 78 - Ein beispielhaftes Element-Design-Canvas aus der YPRC-Fallstudie

Berücksichtigung der Validierung wichtiger Ideen eines Elements

Es ist möglich, dass während der Konzeptarbeit an den ersten Konzepten ein dringendes Bedürfnis entsteht, eine Idee zu validieren (z. B. eine Funktion oder einen Aspekt der Benutzbarkeit).

Die folgenden Prototypen (siehe Abschnitt 2.3) werden als Validierungstechniken für Einsteiger empfohlen:

- Interaktive Mock-ups zur Validierung alternativer Ideen für die Interaktion
- High-Fidelity Mock-ups zur Validierung alternativer Ideen für die visuelle Gestaltung des User Interface
- Funktionale Prototypen zur Validierung kritischer technischer Aspekte

In dieser Phase des Bauprozesses erfordert die Entwicklung spezieller Prototypen jedoch eine sorgfältige Abwägung von Kosten und Nutzen. Die Entwicklung eines speziellen Prototyps kann teurer sein als die echte Implementierung und Validierung einer Funktion in der Praxis. Eine allgemeine Regel für solche Situationen ist schwer zu definieren - es ist eine Frage der Risikotoleranz. Daher sollte der Product Owner diese Frage ausführlich mit dem Umsetzungsteam erörtern, um einen guten Ansatz zu finden.

5.3.4.2 Erstellen einer Story Map zur Definition von User Storys

User Storys können aus den Zielen, Use Cases und Funktionen, die im Element-Design-Canvas beschrieben sind, abgeleitet und mit Hilfe einer Story Map (siehe Abschnitt 5.3.2.2) strukturiert werden. Wir empfehlen, eine erste Story Map auf der Grundlage des Wissens der beteiligten Personen und ohne Berücksichtigung der vorhandenen Konzepte (Lösungsdesignkonzept, Systemdesignkonzepte und Element-Design-Canvases) zu erstellen. Der Grund dafür ist einfach: In dieser Phase des Bauprozesses entwickelt das Team ein erstes Verständnis für die Elemente der Lösung. Indem die Konzepte in diesem Schritt nicht verwendet werden, kann das Team testen, inwieweit es die Elemente versteht.

Diese Arbeitsweise mag auf den ersten Blick unsystematisch erscheinen, weil sie sich nicht auf die bestehenden Konzepte stützt. Aber bedenken Sie, dass Konzepte als externes Gedächtnis dienen und dass diese Situation (die Definition der Story Map) ein guter Zeitpunkt ist, um eine neue Sicht auf das Verständnis des Umsetzungsteams für die digitale Lösung zu erhalten. Dieser neue Blick kann neue Einblicke und neue Ideen für die Funktionsweise der digitalen Lösung bieten. Beachten Sie, dass neue Erkenntnisse und Ideen bewertet werden sollten.

Sobald eine erste Story Map erstellt wurde, können die definierten Storys - einschließlich des Ablaufs der Storys - mit den bestehenden Konzepten verglichen werden, während an den Details der User Storys gearbeitet wird. In diesem Schritt können das Team und der Product Owner an der Story Map, dem System-Backlog und allen Konzepten arbeiten, um ein konsistentes und kohärentes Verständnis der digitalen Lösung zu erreichen. Sobald dieses Verständnis auf einer ausreichenden Detailebene erreicht ist, ist es wichtig, einen Schritt zurückzutreten, um die gesamte Lösung zu betrachten, indem man z. B. die Pressemitteilung aus der Zukunft als Bezugspunkt verwendet und eine Evaluierung der neuen Erkenntnisse und Ideen plant.

5.3.4.3 Definieren von Arbeitsaufträgen sowie Schätzen, Priorisieren und Verwalten des System-Backlogs

Sobald das initiale System-Backlog definiert ist, muss es nach Prioritäten geordnet werden.

DEEP als Instrument zur Priorisierung

Ein gutes Akronym, das die Merkmale des Backlogs beschreibt, ist *DEEP* [Cohn2004]:

- D - Angemessen detailliert (Detailed appropriately) bedeutet, dass die obersten Backlog-Items angemessen detailliert sein müssen. Backlog-Items, die weiter unten im Backlog stehen und eine geringere Priorität haben, müssen zu gegebener Zeit möglicherweise weiter ausgearbeitet werden.
- E - Geschätzt (Estimated) bedeutet, dass der Aufwand für die Bearbeitung von Backlog-Items geschätzt werden sollte. Im Folgenden finden Sie Hinweise zur Schätzung von Backlogs anhand von T-Shirt-Größen.

- E - Emergent (oder entstehend) bedeutet, dass das Backlog ein sich veränderndes Artefakt ist, das kontinuierlich überarbeitet wird.
- P - Priorisiert bedeutet, dass das Backlog immer eine Priorität hat. Die einfachste Art, die Priorität zu definieren, ist, das Backlog als eine Aufgabenliste zu verstehen, bei der die Priorität durch die Reihenfolge der Elemente bestimmt wird.

Alle diese Merkmale sind eng miteinander verbunden. Zum Beispiel erfordert die Schätzung eine gewisse Detailgenauigkeit. Darüber hinaus ist es wichtig zu erkennen, dass die Priorisierung des Backlogs sowohl die User Storys als auch die Design-, Technologie-, Prototyping- und Evaluations-Workitems so abstimmen muss, dass eine korrekte Umsetzung möglich ist. Wenn beispielsweise die Umsetzung einer User Story Details aus einem noch nicht fertiggestellten Konzept der Elementebene erfordert, muss die entsprechende Designaufgabe im Backlog eine höhere Priorität haben als die User Story.

Hinweise zur Aufwandsschätzung

Die Aufwandsschätzung für Backlog-Elemente ist eine echte Herausforderung, nicht nur für die Arbeit an User Storys, sondern auch für alle anderen oben vorgestellten Arten von Backlog-Items. Es gibt viele Techniken zur Aufwandsschätzung (vgl. [McCo2014]). Unserer Erfahrung nach ist die Schätzung anhand von T-Shirt-Größen (vgl. [McCo2014]) ein guter Ausgangspunkt, um ein Gefühl für die Aufwandsschätzung zu bekommen. Bei dieser Technik wird jedes Backlog-Item durch eine T-Shirt-Größe (S, M, L, XL) kategorisiert, die den Aufwand für diese Aufgabe angibt. Mit jedem Größenschritt verdoppelt sich der angenommene Aufwand in etwa, d. h. eine S-User-Story erfordert die Hälfte der Arbeit einer M-User-Story. Ein Vergleich des Aufwands von Backlog-Items ist nur bei Elementen desselben Typs (z. B. User Storys oder Konzept-Workitems) ratsam. Wenn eine budgetorientierte Schätzung obligatorisch ist (z. B. für die Budgetplanung), sollte sich das Team auf ein Basisbudget für die S-Größe für jeden Typ einigen. Auf diese Weise kann das Team eine grobe Schätzung des Budgets vornehmen.

Beispiel YPRC. Abbildung 79 zeigt einen Auszug aus dem System-Backlog der YPRC-Fallstudie, der mit Hilfe der Story Map priorisiert wurde. An der Spitze des Backlogs werden vier User Storys vorgestellt, die sich mit der Entwicklung der ersten Funktionen der Läufer-App befassen (siehe Abbildung 61, orangefarbenes Epic). Aus dem Backlog können wir schließen, dass alle notwendigen Voraussetzungen für diese User Storys erfüllt sind, d.h. die technische Basis für die App ist geschaffen und das Team kann mit der Arbeit an den durch die User Storys definierten Funktionen und Use Cases beginnen.

Die User Story an Position 19 ist die User Story, die sich mit dem Trainer des Läufers und der Echtzeit-Datenvisualisierung beschäftigt (schwarze Gruppe in Abbildung 61). Für die Umsetzung dieser User Story sind noch konzeptuelle und technische Arbeiten erforderlich. Dies ergibt sich aus der Priorität der beiden verwandten Aufgaben an Position 12 (Realisierung der Schnittstelle) und Position 13 (visuelle Gestaltung des User Interface).

Pos	ItemID	Arbeitsaufgabe Typ	Zusammenfassung	T-Shirt
1	US-103	User Story	Als <Läufer> möchte ich mich <im Portal registrieren>, um <die Anwendung> zu nutzen	L
2	US-107	User Story	Als <Läufer> möchte ich <Meine Trainingsparameter konfigurieren>, um <mein erstes Training vorzubereiten>.	XL
3	US-113	User Story	Als <Läufer> möchte ich <meine YPRC-Smartwatch verbinden>, damit <die App während des Trainings Gesundheitsdaten von der Smartwatch empfangen kann>	L
4	US-125	User Story	Als <Läufer> möchte ich <eine Trainingseinheit aufzeichnen>, damit <ich meine Leistung nach der Trainingseinheit überprüfen kann>.	XL
...				
12	T-2	Technisch	Implementierung der Trainingsdatenübertragung zwischen App und Portal	M
13	C-7	Konzept	Erstellen Sie ein visuelles Design für die Benutzeroberfläche "UI-16 Überwachungsbildschirm"	L
..				
19	US-243	User Story	Als <Coach> möchte ich <die Geschwindigkeit und die Herzfrequenz des Läufers in Echtzeit sehen>, damit <ich erkennen kann, wann der Läufer während einer Trainingseinheit zu hart trainiert>.	M
...				

Der große Vorteil dieses Ansatzes besteht darin, dass das Umsetzungsteam keine konkreten Zahlen diskutieren muss, sondern stattdessen den relativen Umfang der Arbeiten besprechen kann. Wenn man mit dieser Technologie arbeitet, wird das Umsetzungsteam früher oder später sein Verständnis für die Arbeit kalibrieren und ein Gefühl dafür entwickeln, wie viel Zeit für eine bestimmte T-Shirt-Größe wirklich benötigt wird.

Praxistipp

Zusammen mit der Story Map ist das System-Backlog ein mächtiges Werkzeug, um die Entwicklung einer digitalen Lösung auf System- und Elementebene im Detail zu steuern. Dieses integrierte Management aller Tätigkeitsbereiche in der Entwicklungs- und Betriebsphase erfordert jedoch Ausbildung und Erfahrung. Unsere Erfahrung zeigt, dass diese Arbeitsweise mit einer Story Map, die das Gesamtbild und die detaillierten Aufgaben in einem Backlog zusammenfasst, ein echter Vorteil für multidisziplinäre Teams ist. Das Backlog liefert zu jedem Zeitpunkt ein klares und transparentes Bild der zu erledigenden Arbeit für das gesamte Team und die Story Map bietet einen guten Überblick über die längerfristige Entwicklungsperspektive. Und schließlich sorgen die Verweise auf die detaillierten Designkonzepte dafür, dass die wichtigen Details der digitalen Lösung dokumentiert werden.

Die Verwaltung des System-Backlogs und der Story Map wird zu einer schwierigen Aufgabe, wenn das Backlog über eine Anzahl von 20-30 Backlog-Items hinauswächst. Eine solche Anzahl von Punkten wird selbst bei sehr einfachen digitalen Lösungen leicht erreicht werden. Wir empfehlen daher die Verwendung spezieller Software-Werkzeuge, um das Product Backlog zusammen mit den verschiedenen Konzepten zu verwalten. Unserer Erfahrung nach sind Werkzeuge zur Verfolgung von Problemen für die Verwaltung des Backlogs und Wikis für die Verwaltung der Konzepte ein guter Ausgangspunkt. Die YPRC-Fallstudie zeigt Ihnen einen möglichen Ansatz für die Unterstützung durch Werkzeuge.

5.3.5 Phase 2: Entwicklung des ersten Releases der digitalen Lösung

Mit dem initial priorisierten Backlog kann die erste Iteration beginnen, um die Reihe von Iterationen zur Entwicklung der einzelnen Elemente zu initialisieren und damit das erste Release der digitalen Lösung zu realisieren.

Hinweise zur Iterationslänge

Wir empfehlen eine Iterationsdauer von drei Wochen als Ausgangspunkt für Einsteiger. Wir empfehlen außerdem, eine Iteration immer am Montag zu beginnen und am Freitag zu beenden. Dies gibt dem Prozess eine klare Struktur und ermöglicht eine langfristige verlässliche Terminplanung für das Team und andere Stakeholder.

Timebox-Empfehlung für jede Veranstaltung

Innerhalb einer Iteration empfehlen wir die folgenden Zeitabschnitte (Timeboxes) für die Ereignisse:

- Ein 15-minütiges Daily (siehe Abschnitt 5.3.1) an jedem Tag der Iteration außer dem letzten Freitag. Ein gutes Zeitfenster für das Daily ist die Zeit vor der Mittagspause.
- Am ersten und dritten Mittwochnachmittag jedes Monats treffen sich das Team und der Product Owner zu einem vierstündigen Timebox-Meeting für die Iterationsplanung.
- Am zweiten Mittwochnachmittag treffen sich das Team, der Product Owner und der Auftraggebende zur Release-Planung.
- Der letzte Freitag der Iteration ist für das Lösungsreview und die Retrospektive reserviert:
 - Das Lösungsreview findet am Vormittag in einer zweistündigen Timebox statt. Wenn Sie das Lösungsreview direkt vor dem Mittagessen ansetzen, haben Sie am Morgen etwas Zeit für die Vorbereitung.
 - Die Retrospektive findet am Nachmittag in einer dreistündigen Timebox statt.
 - Das Daily wird an diesem Tag übersprungen, da das gesamte Team mit dem Lösungsreview und der Retrospektive beschäftigt ist.

Die oben genannten Timebox-Empfehlungen sind ein guter Ausgangspunkt für Einsteiger. Es ist wichtig zu wissen, dass eine Timebox ein festes Ende der Veranstaltung definiert; das bedeutet nicht, dass die Timebox bei jeder Veranstaltung vollständig genutzt werden muss. Die Retrospektive ist das geeignete Ereignis, um den Zeitrahmen anderer Ereignisse und die Dauer der Iteration zu überprüfen. Wenn das Umsetzungsteam die Notwendigkeit sieht, den Zeitrahmen für ein bestimmtes Ereignis zu verlängern oder zu verkürzen, sollte das Team dies gemeinsam mit dem Product Owner besprechen.

Praxistipp

Die Arbeitsweise folgt nun den oben beschriebenen Ereignissen. Das Umsetzungsteam arbeitet an den Aufgaben aus dem System-Backlog und nutzt das tägliche Treffen zur Zusammenarbeit und Koordination. Der Product Owner arbeitet zusammen mit dem Umsetzungsteam gleichzeitig an den anderen Arbeitsaufträgen.

Am Ende einer Iteration wird das Lösungsreview genutzt, um dem Auftraggebenden, dem Product Owner und anderen wichtigen Stakeholdern die Ergebnisse zu präsentieren. Der Product Owner entscheidet, ob die vorgestellte Implementierung akzeptabel ist und Teil des Produktinkrements werden soll. Die Leitlinien für diese Entscheidung ist natürlich die Liste der Annahmekriterien. Auch die Ergebnisse der anderen Aufgaben (wie z. B. konzeptuelle Aufgaben, Prototyping Aufgaben und technische Aufgaben) sollten während des Lösungsreviews kurz vorgestellt werden, um das gesamte Team über die erzielten Fortschritte zu informieren.

Nach der Iterationsüberprüfung findet die Retrospektive statt. Die konkrete Form hängt von verschiedenen Faktoren ab. Die Retrospektive ist besonders nützlich, da Einsteiger voneinander lernen können und um Verbesserungen für die nächsten Sprints zu planen.

Mit den Ergebnissen des Lösungsreviews und der Retrospektive kann die nächste Iterationsplanung in Angriff genommen werden und die Iteration beginnt erneut.

Die Story Map aus der Vorbereitungsphase (siehe Abschnitt 5.3.4) kann verwendet werden, um die Roadmap für die Entwicklung der gesamten digitalen Lösung beizubehalten und die Prioritäten im Product Backlog aus der Lösungsperspektive heraus auszurichten.

Wenn durch diesen Prozess ein akzeptables erstes Release der Lösung erstellt wurde, wird dieses in Produktion gegeben. Die konkrete Vorgehensweise hängt von der Art der Lösung ab und muss mit den relevanten Stakeholdern festgelegt werden.

5.3.6 Phase 3: Weitere Entwicklung während des Betriebs

Sobald die digitale Lösung in Betrieb ist, ändert sich die Arbeitsweise grundlegend, da die Aktivitäten der Fehlerbehebung, der Einarbeitung des Feedbacks von Stakeholdern und der weiteren Evolution der digitalen Lösung gegeneinander priorisiert werden müssen.

Neue Typen von Arbeitsaufträgen zum Management der Arbeit im operativen Betrieb

Wir empfehlen, zu diesem Zweck neue Typen von Arbeitsaufträgen einzuführen:

- Externe Bugs: Beschreibung von Defekten in der digitalen Lösung, die von Kunden/Benutzern gemeldet wurden
- Verbesserungsideen: Beschreibung einer Verbesserung, die die Benutzer/Stakeholder in die digitale Lösung aufnehmen möchten

Einsteigern empfehlen wir, Defekte in der digitalen Lösung in der Produktion als dedizierte Elemente (externe Bugs) im Product Backlog zu erfassen. Sie können auf dieselbe Weise wie Defekte dokumentiert werden (siehe Abschnitt 5.3.3.10). Die separate Bezeichnung hilft, sie von internen Defekten zu unterscheiden, die vom Bau- oder Testteam gefunden wurden.

Zusätzliches Feedback von Stakeholdern kann als Verbesserungsaufgabe erfasst werden. Diese Aufgaben können auf dieselbe Weise beschrieben werden wie andere Konzeptaufgaben (siehe Abschnitt 5.3.3.6). Wir empfehlen auch hier, diese Aufgaben als

Verbesserungsideen zu bezeichnen, um sie von internen konzeptuellen Aufgaben zu unterscheiden. Externe Bugs und Verbesserungsideen können dann im Backlog priorisiert und so schnell wie möglich und nötig bearbeitet werden.

Praxistipp

Als Faustregel gilt, dass ein Umsetzungsteam, das an einer digitalen Lösung in der Produktion arbeitet, etwa 30 % seiner Arbeitskapazität für die Behebung kritischer externer Fehler aus der Produktion reservieren sollte. Wenn diese Kapazität während des Sprints nicht genutzt wird, können weitere Aufgaben aus dem Backlog bearbeitet werden. Unsere Erfahrung zeigt jedoch, dass eine solche Kapazitätsreserve nützlich ist, insbesondere wenn längerfristige Prognosen über den Fortschritt des Umsetzungsteams erforderlich sind.

5.3.7 Phase 4: Außerbetriebnahme

Auf den ersten Blick mag es seltsam erscheinen, in einem Handbuch über Digital Design über die Außerbetriebnahme einer digitalen Lösung zu sprechen. Die Außerbetriebnahme einer digitalen Lösung gehört zum Betrieb einer digitalen Lösung und sollte ebenfalls berücksichtigt werden. Wir wollen in diesem Abschnitt nicht ins Detail gehen, sondern nur ein Bewusstsein dafür schaffen, dass auch diese Phase zum Bauprozess gehört.

Heutzutage kann eine digitale Lösung nicht einfach abgeschaltet werden und vom Markt verschwinden. Ein solches Verhalten würde den Kunden und Benutzern schaden und auch dem Unternehmen, das die digitale Lösung bereitgestellt hat.

Praxistipp

Auch die Außerbetriebnahme einer digitalen Lösung muss geplant werden, und aus der Sicht des Digital Designs sind die folgenden Aspekte wichtig:

- Finden Sie heraus, wer die aktuellen Benutzer des Systems sind. Besonders in großen Organisationen steigt die Zahl der unbekanntenen Benutzer eines Systems oder Dienstes ständig an.
- Bereiten Sie die Kunden und Benutzer im Voraus auf die Außerbetriebnahme vor. Informieren Sie die Kunden/Benutzer zum Beispiel rechtzeitig darüber, dass die Lösung außer Betrieb genommen wird.
- Stellen Sie Kunden/Benutzern Mittel bereit, mit denen sie auf ihre Daten achten können. Je nach Art der digitalen Lösung können Daten enthalten sein, die für die Kunden von echter Bedeutung sind (z. B. Gesundheitsdaten vieler Jahre, wenn YPRC seit Jahren auf dem Markt ist). Die Kunden sollten die Möglichkeit haben, ihre persönlichen Daten aus der digitalen Lösung herauszunehmen. Dies kann bedeuten, dass eine spezielle Datenexportfunktionalität realisiert werden muss.
- Planen Sie gegebenenfalls die Migration der Daten auf einen neuen Dienst oder halten Sie zumindest die vorhandenen Daten für eine spätere Nutzung vor, sofern dies datenschutzrechtlich zulässig ist.
- Machen Sie deutlich, dass sowohl der Dienst als auch die Daten nach der Stilllegung nicht mehr verfügbar/verwendbar sein werden.
- Planen Sie eventuell eine Beobachtungsphase (oder sogar einen Parallelbetrieb) nach der Außerbetriebnahme des Produkts/der Dienstleistung ein, um festzustellen, ob es unerwartete Nebeneffekte gibt.

5.4 Lean Startup als alternativer Ansatz

In den drei vorangegangenen Abschnitten haben wir einen detaillierten Bauprozess für digitale Lösungen vorgestellt. Dieser Prozess folgt einem strukturierten Ansatz, um eine digitale Lösung zu verstehen und zu realisieren. Ein Kernmerkmal des zuvor vorgestellten Prozesses ist, dass die digitale Lösung in Betrieb geht, sobald eine erste vollständige Version realisiert wurde. Wir glauben, dass ein DDP auf Foundation Level Niveau sich der Tatsache bewusst sein sollte, dass dies nicht der einzige Ansatz für den Bau einer digitalen Lösung ist. Im Folgenden stellen wir *Lean Startup* als einen Ansatz vor, der eine andere Philosophie verfolgt.

Lean Startup [Ries2011] ist ein methodischer Rahmen, der darauf abzielt, die Produktentwicklungszyklen durch eine Kombination von auf Geschäftshypothesen basierenden Experimenten, iterativen Produktreleases und validiertem Lernen zu verkürzen. Die zentrale Hypothese ist, dass Start-up-Unternehmen, die ihre Zeit in die iterative Entwicklung eines Produkts investieren, um die Bedürfnisse der frühen Kunden zu erfüllen, die Marktrisiken verringern und die Notwendigkeit einer hohen Anfangsfinanzierung sowie teure Produkteinführungen und Fehlschläge vermeiden können.

Bezogen auf den in Abschnitt 2.1 vorgestellten Bauprozess hat die Wahl eines Lean-Startup-Ansatzes erhebliche Auswirkungen auf alle Phasen des Bauprozesses. Das Scoping und der konzeptuelle Schritt werden auf ein Minimum reduziert, um die eigentliche Entwicklung der digitalen Lösung so früh wie möglich zu beginnen und ein Minimal Viable Product zu realisieren.

Minimal Viable Product als Kernelement von Lean Startup

Eine zentrale Herausforderung beim Lean Startup ist die korrekte Definition des *Minimal Viable Products (MVP)*. Unserer Erfahrung nach besteht ein häufiges Missverständnis bei der Vorstellung eines MVP darin, dass es sich um eine digitale Lösung handelt, die implementiert werden muss. Ein solcher Ansatz führt zu einem schwachen und oft erfolglosen MVP.

Mit dem Verständnis, dass eine digitale Lösung auch ohne Implementierungsaufwand erstellt werden kann (siehe Kapitel 1), kann ein anderer Ansatz für ein MVP gewählt werden. Das MVP kann durch die Wiederverwendung bestehender und verfügbarer digitaler Technologien erstellt werden. Die YPRC-Fallstudie kann als guter Bezugspunkt dienen, um dies deutlich zu machen.

Beispiel YPRC. Während der gesamten Story hat das Team nie die Tatsache in Frage gestellt, dass eine Software implementiert werden muss, um YPRC zu realisieren. Zur Veranschaulichung der Fallstudie ist dies in Ordnung, und wir verwenden dieses Beispiel, um zu erklären, wie ein MVP für YPRC aussehen könnte. Ein Hauptbestandteil des YPRC-Geschäftsmodells ist der Verkauf von Remote-Coaching über eine Sprachverbindung. Ein radikales MVP für diesen Teil des Geschäftsmodells könnte wie folgt aussehen:

- Eine einfache Website, die es den Benutzern ermöglicht, eine Remote-Coaching-Sitzung zu buchen. Die Bezahlung einer solchen Sitzung kann über den bestehenden Zahlungsanbieter oder per Kreditkarte erfolgen.
- Das eigentliche Remote-Coaching erfolgt mit einem Handy-Anruf und ohne zusätzliche digitale Datenübertragung.
- Der Läufer kann einen Instant Messenger mit Standortfreigabefunktion verwenden, um seine Position und seine Laufstrecke mit dem Remote-Coach zu teilen.
- Der Lauftrainer kann mit dem Läufer sprechen und seinen Gesundheitszustand anhand des Klangs seiner Stimme und der Art und Weise, wie der Läufer atmet, bestimmen.

Dieses MVP ist wirklich radikal, zumal der digitale Teil recht primitiv ist. Die Qualität des Coachings könnte aufgrund einer schwachen Datengrundlage eingeschränkt sein. Im Sinne des Lean-Startup-Konzepts kann diese Lösung jedoch innerhalb weniger Tage umgesetzt und eingeführt werden. Nach einigen Wochen im Betrieb wurden viele Erfahrungen gesammelt, die für die Entwicklung einer nächsten Version des MVP sehr nützlich sein werden - dieses Mal vielleicht mit Echtzeitübertragung von Gesundheitsdaten.

Praxistipp

Lean Startup ist besonders dann sinnvoll, wenn die Erfolgchancen einer digitalen Lösung mit anderen Methoden nicht angemessen beurteilt werden können. In diesem Fall wird die Entwicklung sehr schnell angegangen, um mit minimalen Ressourcen eine einsatzfähige digitale Lösung als Minimum Viable Product (MVP) auf den Markt zu bringen. Anhand des Feedbacks von realen Kunden kann dann abgeschätzt werden, ob die Lösung und das Geschäftsmodell erfolgreich sein können und wie die Lösung weiterentwickelt werden kann.

5.5 Schlussfolgerungen zum Bauprozess für Einsteiger

Dieser Abschnitt über den Bauprozess ist beim ersten Lesen sicherlich überwältigend. Der Aufbau einer digitalen Lösung ist in der Tat ein großes und kompliziertes Unterfangen.

Der erste Schritt zum Aufbau digitaler Lösungen besteht darin, zu wissen, was man aufbauen möchte. Genau das ist die Kompetenz von Digital Design. Der zweite Schritt ist die Umsetzung der Ideen. Und das erfordert Prozesskompetenz, von der ersten Idee bis zur tatsächlichen Umsetzung. Dies ist auch ein Teil des Digital Designs.

Der in diesem Kapitel vorgestellte Prozess soll zeigen, wie Design und tatsächliche Umsetzung zusammenwirken und wie ein möglicher Prozess funktioniert. Vor allem aber sollte dieses Kapitel zeigen, dass der Prozess zwar kompliziert, aber überschaubar ist.

6 Gutes Digital Design verwirklichen

Zum Abschluss dieses Handbuchs möchten wir erörtern, wie sich die Prinzipien des guten Digital Designs (siehe Abschnitt 1.4.2) auf den Inhalt des Handbuchs beziehen:

- P1: Gutes Digital Design ist nützlich und gebrauchbar
- P2: Gutes Digital Design ist elegant und ästhetisch
- P3: Gutes Digital Design ist evolutionär
- P4: Gutes Digital Design ist explorativ
- P5: Gutes Digital Design nimmt den ganzen Menschen in den Fokus
- P6: Gutes Digital Design antizipiert die Auswirkungen seiner Ergebnisse
- P7: Gutes Digital Design achtet den Datenschutz und die Datensicherheit
- P8: Gutes Digital Design ist nachhaltig und schafft Nachhaltigkeit
- P9: Gutes Digital Design würdigt Analoges und Digitales in gleicher Weise
- P10: Gutes Digital Design setzt Digitales nur dort ein, wo es erforderlich ist

Dieser Abschnitt fasst den Inhalt des Handbuchs zusammen und zeigt dem Leser, wie wichtig die verschiedenen Inhalte für ein gutes Digital Design sind.

6.1 Beiträge des Digital Design Professional

Zur Einstellung jedes DDPs sollte es gehören, gutes Digital Design zu erreichen. Die Kenntnis und das Verständnis der zehn Prinzipien ist die Grundlage für gutes Digital Design. Die in diesem Handbuch vorgestellten Ideen leisten die folgenden wichtigen Beiträge zum Verwirklichen vom gutem Digital Design.

Gutes Digital Design im Bauprozess

Während des Bauprozesses (siehe Abschnitt 2.1 und Kapitel 5) nutzt ein DDP die zehn Prinzipien als Leitlinien für alle Entscheidungen hinsichtlich der digitalen Lösung. Die folgende Liste gibt einen Überblick über die Schritte des Bauprozesses in Bezug auf die zehn Prinzipien:

- Der Scoping-Schritt ist sehr wichtig, um die Hauptmotivation für den Aufbau einer neuen digitalen Lösung zu verstehen. Die Prinzipien P9 und P10 sollten als Leitfaden für die gesamte Diskussion im Scoping-Schritt dienen. Eine gültige Antwort des Scoping-Schrittes könnte zum Beispiel lauten, dass die Entwicklung einer digitalen Lösung eine gute Idee war, aber nicht wirklich notwendig ist, da die bestehende analoge Lösung ausreicht. Darüber hinaus sollten die Auswirkungen einer geplanten digitalen Lösung im Voraus so detailliert wie möglich erörtert werden, um die Auswirkungen der Lösung zu antizipieren (P6). Einige Auswirkungen werden sich jedoch erst später zeigen. Ein besonderer Aspekt ist dabei der Datenschutz, der von Anfang an berücksichtigt werden muss (P7).
- Während des konzeptuellen Schritts werden wichtige Entscheidungen in Bezug auf den Nutzen und die Benutzbarkeit (Usability) getroffen (P1), da die Hauptfunktionen der digitalen Lösung zusammen mit dem Wertversprechen definiert werden. Jede Funktion sollte unter Berücksichtigung der Prinzipien P9 und P10 bewertet werden, um nur die Funktionen zu definieren, die wirklich notwendig sind. Das bedeutet, dass die Auswirkungen der definierten Funktionen (P6) zusammen mit Datenschutz- und Sicherheitsfragen (P7) berücksichtigt werden sollten. Wichtige Entscheidungen in

Bezug auf die Nachhaltigkeit einer digitalen Lösung werden bei der Festlegung der Einzelheiten des Geschäftsmodells getroffen (P8). In der konzeptuellen Phase werden schließlich die Grundlagen für die evolutionäre und explorative Entwicklung der digitalen Lösung definiert. Dabei sind Form und Funktion der digitalen Lösung ebenso wichtig wie die Entscheidung für die Haupttechnologie.

- In der Entwicklungs- und Betriebsphase müssen Nützlichkeit, Benutzbarkeit, Eleganz und Ästhetik (P1 und P2) sowie Nachhaltigkeit (P8) im Vordergrund stehen, da die vielen kleinen Design- und Realisierungsentscheidungen in Bezug auf Form (z. B. User Interface), Funktion (z. B. gespeicherte Daten) und Qualität (z. B. Reaktionszeit) enorme Auswirkungen auf diese Prinzipien haben werden. Dennoch sind auch alle anderen Prinzipien wichtig. Um gutes Digital Design zu erreichen, sollte ein DDP nicht zögern, Kernentscheidungen der vorangegangenen Schritte in Frage zu stellen. Wenn neue Erkenntnisse auftauchen, die bedeuten, dass ein wichtiges Prinzip verletzt wird, muss diese Entscheidung hinterfragt und möglicherweise revidiert werden.

Gutes Digital Design und Konzeptarbeit

Mit den Designkonzepten (siehe Abschnitt 2.2.2) sollten die beabsichtigten positiven Vorteile, der Nutzen, die Benutzbarkeit, die Eleganz und die Ästhetik (P1 und P2) sowie die Entwicklungs- (P3) und Erkundungsrichtungen (P4) deutlich gemacht werden. Die Grundlagen für Nützlichkeit und Benutzbarkeit werden in den Designkonzepten und den Gedanken der DDPs gelegt, die die Designkonzepte gestalten.

In gut definierten Designkonzepten sind die oben genannten Aspekte bereits sichtbar. Ebenso sollte die Eleganz und Ästhetik einer digitalen Lösung in den Designkonzepten sichtbar werden. Diese Sichtbarkeit kann in verschiedenen Details zum Ausdruck kommen, z. B. in einem sorgfältig gestalteten User Interface, einem ausgeklügelten Formular mit minimaler Interaktion oder in effizienten Datenstrukturen.

Datenschutz und Sicherheit (P7) beginnen ebenfalls in den Designkonzepten. Es ist ein weit verbreitetes Missverständnis, dass es sich dabei nur um ein technologisches Problem handelt. Jede Entscheidung in Bezug auf Form oder Funktion kann Auswirkungen auf den Datenschutz und die Sicherheit haben. Die wichtigsten Fragen müssen lauten:

- Ist diese Funktion oder Entität für meine digitale Lösung wirklich notwendig?
- Ist dieses Attribut für die Lösung wichtig und müssen wir es wirklich speichern?
- Wenn wir bestimmte Daten speichern müssen, wann wollen wir sie löschen?
- Ist es möglich, die Lösung so zu gestalten, dass diese Daten nicht erforderlich sind?

Gutes Digital Design und Prototypen

Die Erstellung von Prototypen (siehe 2.3) ist eine wichtige Technik, um die definierte Lösung anhand der zehn Prinzipien zu bewerten. So können beispielsweise Nützlichkeit und Benutzbarkeit (P1) bereits in einem sehr frühen Stadium des Prozesses mit Prototypen des User Interface bewertet werden. Das Gleiche gilt für die Nachhaltigkeit (P8) - hier können funktionale Prototypen verwendet werden, um Fragen in dieser Richtung zu bewerten. Darüber hinaus ermöglichen Prototypen die Erkundung von (grundlegend anderen) alternativen Designrichtungen und Lösungen. Auch ein evolutionärer Entwicklungsansatz kann auf Prototypen beruhen.

Gutes Digital Design und digitale Technologie

Digitale Technologie (Kapitel 3) hat Auswirkungen auf mehrere Prinzipien. Zum Beispiel ist die evolutionäre und explorative Entwicklung von der Flexibilität der Technologie abhängig und erfordert einen schnellen und flexiblen Konstruktions- und Realisierungsprozess. Nützlichkeit und Benutzbarkeit (P1) stehen in direktem Zusammenhang mit digitaler Technologie, insbesondere wenn es sich um Interaktionstechnologie handelt. Schließlich sollte die Entscheidung für oder gegen eine bestimmte Technologie immer unter dem Gesichtspunkt der Nachhaltigkeit (P8) und des Datenschutzes/der Datensicherheit (P7) getroffen werden.

Gutes Digital Design und Human Factors

Das Verständnis und die Berücksichtigung von Human Factors (siehe Abschnitt 4.1) ist wichtig für die Entwicklung nützlicher und nutzbarer digitaler Lösungen (P1), aber auch für das Verständnis der Auswirkungen der digitalen Lösung auf das Verhalten und die Erfahrungen der Menschen (P5).

Gutes Digital Design und digitale Geschäftsmodelle

Das Geschäftsmodell (siehe Abschnitt 4.2) ist die Grundlage für die Schaffung einer wirtschaftlich erfolgreichen digitalen Lösung und die Grundlage für die Beschaffung der notwendigen finanziellen Mittel für die Entwicklung einer digitalen Lösung. Das Geschäftsmodell steht jedoch auch im Mittelpunkt, wenn es um die Auswirkungen einer digitalen Lösung geht (P6). Es kann auch die Ursache für negative Entwicklungen sein (P5), da die neue Lösung Auswirkungen auf andere Unternehmen haben oder zu unethischem Geschäftsverhalten führen kann. So können beispielsweise Plattform-Geschäftsmodelle zu prekären Beschäftigungsverhältnissen führen oder andere negative Auswirkungen auf den Markt haben.

Das Geschäftsmodell ist auch ein wichtiges Mittel, um zu verstehen, ob eine digitale Lösung wirklich notwendig ist (P10) und ob analoge oder digitale Mittel besser geeignet sind (P9). Sorgfältig definierte Einnahmequellen und Kostentreiber sprechen hier eine deutliche Sprache. Schließlich sollte das Geschäftsmodell Nachhaltigkeit berücksichtigen (P8). Auch wenn keine direkten Kosten oder Gewinne in Bezug auf die Nachhaltigkeit entstehen, so ist die Nachhaltigkeit einer digitalen Lösung mittlerweile ein wichtiger Faktor für jede digitale Lösung. Darüber hinaus sind bestimmte Unternehmen rechtlich nicht für bestimmte negative Auswirkungen verantwortlich und fühlen sich dafür auch nicht zuständig, z. B. die Atomindustrie oder die Batterieindustrie für den Abbau seltener Erdstoffe unter schlechten Bedingungen für die Beschäftigten. Letztendlich ist jedoch das Geschäftsmodell dieser Unternehmen für die negativen Auswirkungen verantwortlich.

Gutes Digital Design und People Management

Schließlich ist auch das People Management (4.3) wichtig, um alle relevanten Stakeholder einzubeziehen und das Engagement für die Entwicklung der angestrebten digitalen Lösung gemäß den zehn Prinzipien zu erhalten. Das People Management wird auch bei der Zusammenstellung des gesamten Umsetzungsteams wichtig, um das breite Kompetenzspektrum abzudecken, das für gutes Digital Design erforderlich ist.

6.2 Die Bedeutung von praktischer Erfahrung und Heuristiken

Das wichtigste Instrument für DDPs sind ihre praktischen Erfahrungen und ihre eigene Reflexion darüber. Wir möchten an dieser Stelle jeden DDP ermutigen, seinen individuellen Lernweg aktiv zu gestalten und hier einige Impulse geben, um gutes Digital Design zu erreichen.

Für Ihre eigene Reflexion kann es hilfreich sein, Ihre eigenen Erfahrungen als Designer mit einer theoretischen Beschreibung der Designpraxis zu vergleichen oder die gesellschaftlichen Diskurse, die Ihr eigenes Handeln umgeben, genauer zu betrachten. Konkrete Ansatzpunkte hierfür wären:

- [Dors2003] gibt eine breite Einführung in das Berufsbild des Designers.

Kritische Texte über die Entwicklung der digitalen Technologie und ihre Auswirkungen auf die Gesellschaft (z. B.

- [Lani2011]) schärfen das Bewusstsein für die möglichen Auswirkungen einer Lösung.

Es ist auch wertvoll, sich mit den praktischen Erfahrungen etablierter Designer auseinanderzusetzen und sich von der Reflexion konkreter Designergebnisse inspirieren zu lassen. Beispiele hierfür sind:

- Texte über die Arbeit herausragender Designer (z. B. [Rams2016]) verbessern Ihr eigenes ästhetisches Wissen und können als Inspirationsquelle dienen
- Inspiration durch Designpreise, z. B. Webby Awards: <https://www.webbyawards.com/>
- Inspiration durch Portfolios renommierter Designagenturen (z. B. Frog Design: <https://www.frogdesign.com/work>)

Eine besondere Kategorie für das Lernen aus den Erfahrungen anderer ist die Heuristik. Heuristik kann definiert werden als eine „Strategie, die einen Teil der Informationen ignoriert, mit dem Ziel, Entscheidungen schneller, genügsamer und/oder genauer zu treffen als mit komplexeren Methoden“ [GiGa2011]. Für den DDP gibt es die konkrete Möglichkeit, Heuristiken als Konzeptmodifikatoren zu verwenden, die schnell zu einer potenziellen Lösung führen und so die Chance für ein neuartiges Design bieten. Ein guter Überblick über solche Designheuristiken und die Erläuterung ihre Verwendung findet sich in [Desi2018].

Darüber hinaus kann der DDP in der Praxis von Usability-Heuristiken profitieren, die vor allem wahrnehmungspsychologische Grundlagen leicht handhabbar machen. Zwei sehr wertvolle Beiträge sind:

- [Wein2011] ist eine Sammlung von 100 Heuristiken aus dem Bereich Human Factors für Designer.
- Eine Sammlung von Heuristiken, die die Benutzbarkeit von Software verbessern, findet sich in [Niel1994].

6.3 Die Wichtigkeit von Teamarbeit

Die in diesem Kapitel vorgestellten Ideen zeigen, dass gutes Digital Design ein Thema für den gesamten Bauprozess ist und eine enge Zusammenarbeit zwischen Management, Design, Konstruktion und Realisierung erfordert. DDPs sollten daher ihre eigenen Kompetenzen (und die Kompetenzen ihrer Teams) kontinuierlich anhand der zehn Prinzipien bewerten und bei Bedarf zusätzliche Fachleute hinzuziehen. Beispiele für Fachleute sind:

- Datenschutzfachleute für den Schutz von Daten und Privatsphäre (P7)
- Fachleute für visuelles Design für ein elegantes und ästhetisches visuelles Design (P2)
- Fachleute für Ergonomie und Benutzbarkeit (Usability), um angenehme, nützliche und benutzbare digitale Lösungen zu schaffen (P1)
- Requirements Engineers zur Identifizierung von Stakeholdern und zum Verständnis der Bedürfnisse der Stakeholder (P1 und P5)
- Sozialwissenschaftler und insbesondere Ethnologen, um die Auswirkungen einer digitalen Lösung zu antizipieren (P6)
- Fachleute für Nachhaltigkeit zur Bewertung und Verbesserung der Nachhaltigkeit einer digitalen Lösung (P8)
- Fachleute für Design und Konstruktion, um eine elegante und ästhetische digitale Lösung auf den wahrnehmbaren und zugrundeliegenden Ebenen zu erreichen (P2)

Schließlich bleibt noch eine wichtige Feststellung. Dieses Handbuch zeigt, dass Digital Design ein sehr vielfältiges Berufsbild ist, das unterschiedliche Fähigkeiten erfordert. Wir glauben, dass es möglich ist, die Bedeutung und den Zusammenhang zwischen all diesen Fähigkeiten zu verstehen. Wir glauben außerdem, dass es möglich ist, einige dieser Fähigkeiten zu beherrschen. Ein Meister des gesamten Spektrums an Fähigkeiten zu werden, ist möglich, aber nur für außergewöhnliche Talente. Für einen Durchschnittsmenschen, wie wir als Autorinnen und Autoren uns selbst sehen, bleibt daher folgendes Fazit für dieses Kapitel und das Handbuch:

**Gutes Digital Design kann nur durch transdisziplinäre Teamarbeit erreicht werden,
aber nur mit einem Team,
das die Vielfalt an Fähigkeiten des Digital Designs abdecken kann.**

I. Literaturverzeichnis

- [Alex2005] Alexander, I. F.: A Taxonomy of Stakeholders: Human Roles in System Development. International Journal of Technology and Human Interaction, Vol 1, 1, 2005, pages 23-59.
- [AllS1977] Alexander, C., Ishikawa, S., Siverstein, M.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, 1977.
- [Ande2010] Anderson, D. J.: Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press, 2010.
- [BaCoWh1999] Baghai, M., Coley, S., White, D.: The Alchemy of Growth – Kickstarting and Sustaining Growth in Your Company. Orion Business Books, 1999.
- [Barn2011] Barnum, C. M.: Usability Testing Essentials. Ready, Set...Test! Morgan Kaufmann, 2011.
- [BBKL1978] Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G., Merritt, M.: Characteristics of Software Quality. North Holland, 1978.
- [BCDE2015] Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Prenzenstadler, B., Seyff, N., Venters, C.: Sustainability Design and Software: The Karlskrona Manifesto. International Conference on Software Engineering (ICSE), 467-476, 2015.
- [BeBa2007] Beckman, S., Barry, M.: Innovation as a Learning Process: Embedding Design Thinking, California Management Review, 2007.
- [BeHo1998] Beyer, H., Holtzblatt, K.: Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann, 1998.
- [Bitk2017] Bitkom e. V.: Rollenideal Digital Design - Erfolgreiche Digitalisierung und Digitale Transformation erfordern ein Umdenken in der Softwareentwicklung, 2017, <https://www.bitkom.org/Bitkom/Publikationen/Rollenideal-Digital-Design.html>, last accessed 2021/03/11.
- [Bloo2018] Bloomberg, J.: Digitization, Digitalization, And Digital Transformation: Confuse Them At Your Peril <https://www.forbes.com/sites/jasonbloomberg/2018/04/29/digitization-digitalization-and-digital-transformation-confuse-them-at-your-peril/#6f4658632f2c>, last accessed 2021/03/11.
- [BrDP2005] Broy, M., Deißeböck, F., Pizka, M.: A Holistic Approach to Software Quality at Work. Third World Congress for Software Quality, Munich, Germany, 2005.
- [Brow2009] Brown, T.: Change by Design. Harper Business, 2009.
- [Budi2018] Budiu, R.: Change Blindness in UX: Definition. <https://www.nngroup.com/articles/change-blindness-definition>, 2018, last accessed 2021/05/03.
- [Chou2013] Choudary, S. P.: Why Business Models Fail: Pipes vs. Platforms. Wired Magazine, 2013.
- [Cohn2004] Cohn, M.: User Stories Applied: For Agile Software Development. Addison Wesley Professional, 2004
- [Coop2004] Cooper, A.: The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity. 2nd Edition, Que, Indianapolis, 2004.
- [CPRE2020] IREB Certified Professional for Requirements Engineering - Foundation Level Syllabus, Version 3.0.1 October 7, 2020.

- [CPUX2018] UXQB Certified Professional for Usability and User Experience - Curriculum and Glossary Version 3.15, March 23, 2018
- [CRCN2014] Cooper, A., Reimann, R., Cronin, D., Noessel, C.: About Face: The Essentials of Interaction Design. 4th Edition, John Wiley & Sons, 2014.
- [Cros2006] Cross, N.: Designerly Ways of Knowing. Birkhäuser, 2006.
- [Demi2000] Deming, W. E.: Out of the Crisis. Reprint. MIT Press, 2018.
- [Desi2018] Design Heuristics: Strategies to Inspire Ideas. <https://www.designheuristics.com/>, 2018, last accessed 2021/03/11.
- [Dick2019] Dickel, S.: Prototyping Society. Bielefeld: transcript, 2019.
- [Dors1997] Dorst, K.: Describing Design – A Comparison of Paradigms. Delft University Press, Delft, The Netherlands, 1997.
- [Dors2003] Dorst, K.: Understanding Design: 150 Reflections on Being a Designer. BIS, 2003.
- [Duck2013] Dueck G.: E-Man: Die neuen virtuellen Herrscher. Vieweg+Teubner Verlag, 2013
- [ErMa2008] Erlhoff, M., Marshall, T. (Eds.): Design Dictionary: Perspectives on Design Terminology. Birkhäuser, 2008.
- [Fla2018] Flaherty, K.: Why Personas Fail. <https://www.nngroup.com/articles/why-personas-fail>, last accessed 2021/05/02.
- [Floy1984] Floyd, C.: A Systematic Look at Prototyping. In: Budde R., Kuhlenkamp K., Mathiassen L., Züllighoven H. (eds): Approaches to Prototyping. Springer, Berlin, Heidelberg, 1984.
- [Fros2020] Frost, B.: Atomic Design. <https://atomicdesign.bradfrost.com>, last accessed 2021/03/11.
- [FrPr2009] Freemann, S., Pryce, N.: Growing Object-Oriented Software, Guided by Tests. Addison-Wesley, 2009.
- [Gass2013] Gassmann, O.: The St. Gallen Business Model Navigator, BMI Pattern Cards <https://bmilab.com/resources>, last accessed 2021/03/11.
- [GBGSV2013] Grapenthin, S., Book, M., Gruhn, V., Schneider, C., Völker, K.: Reducing Complexity Using an Interaction Room: An Experience Report. SIGDOC 2013, pp. 71-76, 2013.
- [GeSE2017] Geissdoerfer, M., Savaget, P., Evans, S.: The Cambridge Business Model Innovation Process. Procedia Manufacturing 8:262–269, 2017.
- [GiGa2011] Gigerenzer, G., Gaissmaier, W.: Heuristic Decision Making. Annual Review of Psychology 62(1):451–482, 2011
- [Glas2006] Glass, R. L.: Software Creativity 2.0. developer.* books, 2006.
- [Glin2007] Glinz, M.: On Non-Functional Requirements. 15th IEEE International Requirements Engineering Conference (RE'07), Delhi, India, 21–26, 2007
- [Glin2008] Glinz, M.: A Risk-Based, Value-Oriented Approach to Quality Requirements. IEEE Software 25(2):34–41, 2008
- [Glin2020] Glinz, M.: A Glossary of Requirements Engineering Terminology Version 2.0 October 2020. International Requirements Engineering Board (IREB)
- [GOOG2005] Google re:Work, Foster psychological safety: <https://rework.withgoogle.com/guides/understanding-team-effectiveness/steps/foster-psychological-safety/> last accessed 2021/03/11
- [GOOG2020] Google: Google re:Work, <https://rework.withgoogle.com>, last accessed 2021/03/11.
- [Grop1930] Gropius, W.: bauhausbauten dessau. bauhausbücher 12, albert langen verlag, 1930.

- [Grud1992] Grudin, J.: Utility and Usability: Research Issues and Development Contexts. *Interacting with Computers* 4(2):209–217, 1992.
- [Hass2008] Hassenzahl, M.: User Experience (UX): Towards an Experiential Perspective on Product Quality. *International Conference on Association Francophone d'Interaction Homme-Machine*, 11-15, 2008.
- [HaTr2006] Hassenzahl, M., Tractinsky, N.: User Experience – A Research Agenda. *Behavior & Information Technology* 25(2):91–97, 2006.
- [Hewe1992] Hewett, T. et al.: *ACM Curricula for Human-Computer Interaction*. ACM, 1992.
- [Hinm2012] Hinman, R.: *The Mobile Frontier: A Guide for Designing Mobile Experiences*. Rosenfeld, 2012.
- [Hüth2009] Hüther, G.: “Gelassenheit hilft - Anregungen für Gehirnbenutzer“ (in German), <https://www.youtube.com/watch?v=2XIJmew2IK4>, last accessed 2021/03/11.
- [IDSA2020] Industrial Designers Society of America, “How They Do It” <http://www.idsa.org/education/how-they-do-it>, last accessed 2021/03/11.
- [IEEE2017] ISO/IEC/IEEE 24765. *Systems and Software Engineering — Vocabulary*. Second Edition. ISO/IEC/IEEE 24765:2017E). New York: The Institute of Electrical and Electronics Engineering, 2017.
- [ISO2001] ISO/IEC 9126-1: *Software Engineering — Product Quality — Part 1: Quality Model*. International Organization for Standardization, 2001
- [ISO2011] ISO/IEC 25010: *Software Product Quality*. <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>, 2011, last accessed 2021/03/11.
- [ISO2018] ISO 9241-11: *Ergonomics of Human-System Interaction — Part 11: Usability: Definitions and Concepts*. 2018.
- [ISO2019] ISO 9241-210:2019: *Ergonomics of Human-System Interaction — Part 210: Human-Centred Design for Interactive Systems*. ISO, 2019.
- [ISO2020] ISO 9241-110: *Ergonomics of Human-System Interaction — Part 110: Interaction Principles*. 2020.
- [Jera2016] Jerald, J.: *The VR Book: Human-Centered Design for Virtual Reality* (ACM Books). Morgan & Claypool Publishers, 2016.
- [John2010] Johnson, J.: *Designing with the Mind in Mind*. Morgan Kaufmann, 2010.
- [JPMLT2018] Jamshidi, P., Pahl, C., Mendonça, N. C., Lewis, J., Tilkov, S.: *Microservices: The Journey So Far and Challenges Ahead*. *IEEE Software* 35(3):24-35, 2018.
- [Keir1998] Keirse, D.: *Please Understand Me II: Temperament, Character, Intelligence* (1st ed.). Prometheus Nemesis Book Co. ISBN 1-885705-02-6.
- [Kell2016] Kelly, K.: *The Inevitable: Understanding the 12 Technological Forces that will shape our Future*. Viking, 2016.
- [KEMP2017] Kemper, M.: *Teamsetup & interpersonal dynamics* <https://www.smarter-service.com/2017/11/28/persoenliches-im-digitalen-transformationsgetriebe/>
- [Lani2011] Lanier, J.: *You Are Not a Gadget: A Manifesto*. Penguin, 2011.
- [LBGH2018] Lauenroth, K., Bramsiepe, H., Gilbert, D., Hartwig, R., Lehn, K., Schubert, U., Trapp, M.: *The Digital Design Manifesto*, https://www.digital-design-manifest.de/wp-content/uploads/2021/03/Bitkom_LF_Digital_Design_Manifest_EN.pdf, last accessed 2021/05/02.

- [LRHV2009] Law, E. L. C., Roto, V., Hassenzahl, M., Vermeeren A. P., Kort, J.: Understanding, Scoping and Defining User Experience: A Survey Approach. Proceedings of the Conference on Human Factors in Computing Systems, 719-728, 2009.
- [LWLB2017] Lee, J. D., Wickens, D., Liu, Y., Boyle, L.: Designing for People: An Introduction to Human Factors Engineering. CreateSpace Independent Publishing, 2017.
- [MaLa2015] Margolis, E., Laurence, S. (Eds): The Conceptual Mind: New Directions in the Study of Concepts. MIT Press, 2015.
- [Mcco2004] McConnell, S.: Code Complete: A Practical Handbook of Software Construction. Microsoft Press, 2004.
- [McCo2014] McConnell, S.: Software Estimation: Demystifying the Black Art. Microsoft Press, 2014.
- [McEl2017] McElroy, K.: Prototyping for Designers. O'Reilly, 2017.
- [MCPKV2006] McCurdy, M., Connors, C., Pyrzak, G., Kanefsky, B., Vera, A. H.: Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed-fidelity success. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06). ACM, New York, 2006.
- [McRW1977] McCall, J. A., Richards, P. K., Walters, G. F.: Factors in Software Quality. Rome Air Development Center, 1977.
- [Meye2014] Meyer, B.: Agile: The Good, the Hype and the Ugly. Springer, 2014.
- [MiKi1994] Milgram, P., Kishino, F.: A Taxonomy of Mixed Reality Visual Displays. IEICE Transactions on Information and Systems. Vol. 77, 1994.
- [Ming2020] Minge, M.: meCUE 2.0. <http://mecue.de/english/index.html>, 2020, last accessed 2021/03/11.
- [MoJo2016] Moazed, A., Johnson, N. L.: Modern Monopolies: What It Takes to Dominate the 21st Century Economy. Griffin Publishing, 2016.
- [Newm2020] Newman, D.: The Process of Design Squiggle. thedesignsquiggle.com, last accessed 2021/03/11.
- [Niel1994] Nielsen, D.: 10 Usability Heuristics for User Interface Design: <https://www.nngroup.com/articles/ten-usability-heuristics/>. Last accessed August 2019.
- [NIJS2019] Nijs, D.(Eds.): Advanced Imagineering. Edward Elgar Publishing, 2019.
- [Nobl1996] Noblet, J. de: Industrial Design: Reflection of a Century. Flammarion 1996.
- [OKKP2015] Obbink, H., Kruchten, P., Kozaczynski, W., Postema, H., Ran, A., Dominick, L., Kazman, R., Hilliard, R., Tracz, W., Kahane, E.: Software Architecture Review and Assessment Report, <https://pkruchten.files.wordpress.com/2011/09/sarav1.pdf>, last accessed 2021/03/11.
- [OPBS2014] Osterwalder, A., Pigneur, Y., Bernarda, G., Smith, A.: Value Proposition Design: How to Create Products and Services Customers Want. Wiley, 2014.
- [Orio2018] Oriol, M. et al.: FAME: Supporting Continuous Requirements Elicitation by Combining User Feedback and Monitoring. 26th IEEE International Requirements Engineering Conference (RE), 2018.
- [OsPi2010] Osterwalder, A., Pigneur, Y.: Business Model Generation. Wiley, 2010.
- [Pari2021] Paris-Bicking, Christel "DICP" www.keypeoplecheck.com, last accessed 2021/05/02
- [Patt2014] Patton, J.: User Story Mapping: Discover the Whole Story, Build the Right Product. O'Reilly, 2014.

- [PaYC2010] Paul, D., Yeates, D., Cadle, J. (Eds): Business Analysis. Second Edition, BCS, 2010.
- [PeWo1992] Perry, D. E., Wolf, A. L.: Foundations for the Study of Software Architecture. ACM SIGSOFT Software Engineering Notes 17(4), 1992.
- [PoLR2013] Polaine, A., Løvlie, L., Reason, B.: Service Design: From Insight to Implementation. Rosenfeld Media, 2013.
- [Rams2016] Rams, D.: Less but besser. 6th Edition. Jo Klatt Design+Design Verlag, 2016.
- [Rein1997] Reinertson, D.: Managing the Design Factory. Simon and Schuster, 1997.
- [Ries2011] Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Books, 2011.
- [RiWe1973] Rittel, H. W. J., Webber, M. M.: Dilemmas in a General Theory of Planning. In: Policy Sciences 4(2):155–169, 1973.
- [RoCa2003] Rosson, B. M.; Carroll, J. M. (2003): Scenario-Based design. In: Jacko J. A. (ed.); The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications. L. Erlbaum Associates Inc., USA, 1032–1050.
- [Ross2019] Rossmann, J.: Think Like Amazon: 50 ½ Ideas to Become a Digital Leader. McGraw-Hill, 2019.
- [Royce1970] Royce, W.: Managing the Development of Large Software Systems: Concepts and Techniques. In: Proceedings of IEEE WESCOM. IEEE Computer Society Press, Los Alamitos, 1970.
- [SaLe2016] Sauro, J., Lewis, J. R.: Quantifying the User Experience: Practical Statistics for User Research. Morgan Kaufmann, 2016.
- [ScHo2016] Schmalstieg, D., Höllerer, T.: Augmented Reality: Principles and Practice. Addison-Wesley, 2016.
- [ScKr2010] Schulze, K., Krömker, H.: A framework to measure user eXperience of interactive online products. International Conference on Methods and Techniques in Behavioral Research, 1–5, 2010.
- [ScSu2020] Schwaber, K., Sutherland, J.: The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. November 2020.
- [SeRP2017] Sedano, T., Ralph, P., Péraire, C.: Software Development Waste. 39th International Conference on Software Engineering (ICSE), 2017.
- [ShRP2019] Sharp, H., Rogers, Y., Preece, J.: Interaction Design: Beyond Human-Computer Interaction. Fifth edition. Wiley, 2019.
- [SIWi1997] Slater, M., Wilbur, S.: A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments. PRESENCE: Virtual and Augmented Reality 6(6):603–616, December 1997.
- [Snow2005] Snowden, D.: Multi-Ontology Sense Making: A New Simplicity in Decision Making. Informatics in Primary Health Care 13(1):45-54, 2005.
- [Snyd2003] Snyder, C.: Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces. Morgan Kaufmann, 2003.
- [SSRW2013] Stanton, N. A., Salmon, P. M., Rafferty, L. A., Walker, G. H., Baber, C., Jenkins, D. P.: Human Factors Methods. A Practical Guide for Engineering and Design. Taylor & Francis, 2013.
- [Steu1992] Steuer, J.: Defining Virtual Reality: Dimensions Determining Telepresence. Journal of Communication 42(4):73–93, 1992.

- [StSc2011] Stickdorn, M., Schneider, J.: This is Service Design Thinking: Basics - Tools – Cases. BIS Publishers, 2011.
- [TARV2019] Tarver, E.: Corporate Culture. <https://www.investopedia.com/terms/c/corporate-culture.asp>, 2019, last accessed 2021/03/11.
- [ThMa2007] Thüring, M., Mahlke, S.: Usability, Aesthetics and Emotions in Human–Technology Interaction. International Journal of Psychology 42(4):253–264, 2007.
- [TuAI2013] Tullis, T.; Albert, B.: Measuring the User Experience. Collecting, Analyzing, and Presenting Usability Metrics. Morgan Kaufmann, 2013.
- [User2020] User Interface Design GmbH: AttrakDiff. <http://attrakdiff.de/index-en.html>, 2020, last accessed 2021/03/11.
- [VPGV2008] Van de Ven, A., Polley, D., Garud, R., Venkataraman, S.: The Innovation Journey. Oxford University Press, 2008.
- [Wake2003] Wake, B: INVEST in Good Stories, and SMART Tasks. 2003, <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>, last accessed 2021/03/11.
- [WaMe1986] Ward, P. T., Mellor, S. J.: Structured Development for Real-Time Systems. Pearson, 1986.
- [WEF2016] World Economic Forum: The 10 skills you need to thrive in the Fourth Industrial Revolution, 2016, <https://www.weforum.org/agenda/2016/01/the-10-skills-you-need-to-thrive-in-the-fourth-industrial-revolution/>, last accessed 2021/03/11.
- [WEF2018] World Economic Forum: 5 things to know about the future of jobs, 2018, <https://www.weforum.org/agenda/2018/09/future-of-jobs-2018-things-to-know/>, last accessed 2021/03/11.
- [Wegn2005] Wegner, I.: Complexity Theory. Exploring the Limits of Efficient Algorithms. Springer, 2005.
- [Wei2018] Wei, E.: Remains of the Day: Invisible Asymptotes. 2018 <https://www.eugenewei.com/blog/2018/5/21/invisible-asymptotes>, last accessed 2021/03/11.
- [Wein1971] Weinberg, G. M.: The Psychology of Computer Programming. Van Nostrand Reinhold, 1971.
- [Wein2011] Weinschenk, S. M.: 100 Things Every Designer Needs to Know about People. New Riders, 2011.
- [WHBP2016] Wickens, C. D., Holland, J. G., Banbury, S., Parasuraman, R.: Engineering Psychology and Human Performance. Taylor and Francis, 2016.
- [Wik2020a] Wikipedia: Databases. <https://en.wikipedia.org/wiki/Database>, last accessed 2021/03/11.
- [Wik2020b] Wikipedia: Outline of databases. https://en.wikipedia.org/wiki/Outline_of_databases, last accessed 2021/03/11.
- [Wiki2020c] Wikipedia: Keirsey Temperament Sorter. https://en.wikipedia.org/wiki/Keirsey_Temperament_Sorter, last accessed 2021/03/11.
- [WiSh2011] Williamson, D. P., Shmoys, D. B.: The Design of Approximation Algorithms. Cambridge University Press, 2001.
- [XDF2020] Xue, H., Desmet, P. M. A., Fokkinga, S. F.: Mood Granularity for Design: Introducing a Holistic Typology of 20 Mood States. International Journal of Design, 14(1):1–18, 2020.

II. Abbildungsverzeichnis

Abbildung 1 - Vereinfachte Darstellung der digitalen Lösung YPRC	21
Abbildung 2 - Vereinfachte Form und Funktion der digitalen Lösung YPRC	21
Abbildung 3 - Überschneidungen zwischen den Kerntätigkeitsbereichen des Bauprozesses	29
Abbildung 4 - Überblick über die Abstraktionsebenen Lösung, System und Element	37
Abbildung 5 - Aktivitäten des Bauprozesses laufen parallel auf System- und Elementebene.....	41
Abbildung 6 - Das π -förmige Kompetenzprofil eines DDP auf Foundation Level Niveau	42
Abbildung 7 - Der Design-Squiggle [Newm2020].....	49
Abbildung 8 - Das Dual-Mode-Modell des Designs [Dors1997].....	51
Abbildung 9 - Die drei wesentlichen Schritte und Arbeitsprodukte des Bauprozesses.....	52
Abbildung 10 - Die Kontext-Scoping-Perspektiven des Digital Design	53
Abbildung 11 - Angepasste Qualitätsmerkmale nach ISO/IEC 25010 [ISO2011].....	66
Abbildung 12 - Ein idealisiertes Modell des Bauprozesses.....	69
Abbildung 13 - Das Ende des Bauprozesses ist ein neuer Anfang.....	71
Abbildung 14 - Eine Pressemitteilung aus der Zukunft	84
Abbildung 15 - Eine Persona-Vorlage	86
Abbildung 16 - Eine Vorlage für ein Wertversprechen (Value Proposition Canvas)	87
Abbildung 17 - Vorlage für eine Customer Journey Map Canvas	88
Abbildung 18 - Eine Vorlage für ein Business Model Canvas	89
Abbildung 19 - Eine Zielvorlage/Beispiel aus der YPRC-Fallstudie	96
Abbildung 20 - Eine Randbedingungs Vorlage	98
Abbildung 21 - Eine Benutzertyp-Vorlage	99
Abbildung 22 - Vorlage für ein bestehendes Objekt.....	100
Abbildung 23 - Eine Vorlage für ein bestehendes System	101
Abbildung 24 - Eine Gerätevorlage	102
Abbildung 25 - Eine Software-Vorlage	103
Abbildung 26 - Eine Szenario-Vorlage	104
Abbildung 27 - Eine Hardware-Schnittstellenvorlage	106
Abbildung 28 – Eine User Interface-Vorlage	108
Abbildung 29 – Eine Software-Schnittstellenvorlage	110
Abbildung 30 - Eine Entitätsvorlage	112
Abbildung 31 - Eine Vorlage für physische Teile.....	113
Abbildung 32 - Eine Funktionsvorlage.....	114
Abbildung 33 - Eine Vorlage für einen Use Case.....	115
Abbildung 34 - Eine Vorlage für Qualitätsanforderungen.....	117
Abbildung 35 - Überblick über das digitale YPRC-System mit seinen Elementen.....	118

Abbildung 36 - Bausteine eines Software-/Gerätedesignkonzepts	119
Abbildung 37 - Beispiel für einen Use Case, der andere Elemente einschließt.....	121
Abbildung 38 - Bausteine eines Systemdesignkonzepts	122
Abbildung 39 - Überblick über die Arten von Designkonzepten in den verschiedenen Phasen des Bauprozesses	128
Abbildung 40 – Interaktionsumfang.....	137
Abbildung 41 - Ein Beispiel eines Bildschirms aus den Prototypen der YPRC-Fallstudie, das verschiedene visuelle Qualitätsstufen zeigt, die zu unterschiedlichen Realitätsgraden der Dimension der sensorischen Vollkommenheit führen. Karte im Bild ganz rechts: © OpenStreetMap-Mitwirkende, Lizenz: www.openstreetmap.org/copyright	141
Abbildung 42 - Typische Verwendung von Prototypen (Aktivitäten) im Rahmen eines Digital Design Projekts	148
Abbildung 43 - Beispielhafte Industriedesign-Prototypen (Copyright: Neuland/GENERATIONDESIGN).....	152
Abbildung 44 - Beispielbildschirme eines Papierprototyps für die YPRC-Fallstudie.....	154
Abbildung 45 - Beispiel eines Papierprototyps für die Fallstudie YPRC mit mehreren skizzierten Bildschirmseiten.....	155
Abbildung 46 - Beispiel eines Kartonprototyps der Smartwatch der YPRC-Fallstudie	156
Abbildung 47 - Beispiel für ein Storyboard der YPRC-Fallstudie	157
Abbildung 48 - Beispiele für Technologie, unterteilt in (1) wahrnehmbare Form und Funktion für Hardware und Software (oberer Teil) und (2) zugrundeliegende Form und Funktion für Hardware und Software (unterer Teil)	161
Abbildung 49 - Das Virtuality Continuum nach Milgram und Kishino [MiKi1994].....	171
Abbildung 50 - Vereinfachtes Modell von Empfindung, Aufmerksamkeit und Wahrnehmung.....	192
Abbildung 51 - Das Komponentenmodell der User Experience (CUE) (angepasst von [ThMa2007][Ming2020]).....	197
Abbildung 52 - Das Three-Horizon-Model.....	202
Abbildung 53 - Effektive Verteilung der Temperamente für die interdisziplinäre Zusammenarbeit bei der Erstellung einer digitalen Lösung [KEMP2017]	215
Abbildung 54 - Three-Horizon-Model und unterstützende Teamprofile	217
Abbildung 55 - Beispiele für Teamprofile	217
Abbildung 56 - Ein Instrument zur Messung des Verständnisses für eine digitale Lösung	227
Abbildung 57 - Ein Instrument zur Messung des Grades des Engagements	227
Abbildung 58 - Beziehungen, die die Entwicklung eines guten Business Model Canvas unterstützen	242
Abbildung 59 - System-Backlog und System-Board als Kanban-Board	251
Abbildung 60 - Beispiel für eine Story Map aus der YPRC-Fallstudie	252
Abbildung 61 - Verwendung einer Story Map zur Priorisierung von User Storys	253
Abbildung 62 - Story Map und Epic Board	254
Abbildung 63 - Beziehungen zwischen den Arbeitsprodukten auf den drei Ebenen	257

Abbildung 64 - Beziehungen zwischen Workitem-Management und Designkonzepten.....	258
Abbildung 65 - Beispiel für eine Lösungsaufgabe aus der YPRC-Fallstudie	261
Abbildung 66 - Beispiel für ein Epic aus der YPRC-Fallstudie	262
Abbildung 67 - Beziehung zwischen epischen Konzepten und Designkonzepten.....	263
Abbildung 68 - Beziehung zwischen User Story und Designkonzepten	265
Abbildung 69 - Beispiel für eine User Story aus der YPRC-Fallstudie.....	266
Abbildung 70 - Beziehung zwischen Konzept-Workitems und Designkonzepten.....	269
Abbildung 71 - Beispiel für eine Konzept-Workitem aus der YPRC-Fallstudie	269
Abbildung 72 - Beispiel eines Prototyp-Workitems aus der YPRC-Fallstudie	271
Abbildung 73 - Beispiel für ein Evaluations-Workitem aus der YPRC-Fallstudie.....	272
Abbildung 74 - Beziehung zwischen technischen Workitems und Designkonzepten	273
Abbildung 75 - Beispiel für ein technisches Workitem aus der YPRC-Fallstudie	274
Abbildung 76 - Beispiel für ein Defekt-Workitem aus der YPRC-Fallstudie	275
Abbildung 77 - Eine einfache Struktur für ein Element-Design-Canvas	277
Abbildung 78 - Ein beispielhaftes Element-Design-Canvas aus der YPRC-Fallstudie	278
Abbildung 79 - Auszug aus dem System-Backlog der YPRC-Fallstudie	281

III. Tabellenverzeichnis

Tabelle 1 - Terminologie in der Gebäudearchitektur und im Digital Design.....	14
Tabelle 2 - Beispielhafte Beziehungen zwischen einem Designkonzept und einem Realisierungskonzept	30
Tabelle 3 - Dokumentvorlage für einen Digital Design Brief	76
Tabelle 4 - Dokumentvorlage für ein Lösungsdesignkonzept	78
Tabelle 5 - Dokumentvorlage für ein Systemdesignkonzept.....	79
Tabelle 6 - Dokumentvorlage für ein Softwaredesignkonzept.....	80
Tabelle 7 - Dokumentvorlage für ein Gerätedesignkonzept.....	82
Table 8 – Überblick über Dokumentationstechniken und Arbeitsprodukte für die Lösungsebene	83
Tabelle 9 - Beziehungen zwischen Systemebene und Elementebene	124
Tabelle 10 - Beispiele für Beziehungen zwischen der Lösungsebene und der System-/Elementebene.....	125
Tabelle 11 - Konzepttypen im Digital Design	127
Tabelle 12 - Kriterien für die Kategorisierung von Prototypen	136
Tabelle 13 - Fidelity-Profil des YPRC-Prototyps für die Perspektive des Coaches in Bezug auf die fünf Dimensionen eines Prototyps	140
Tabelle 14 - Fidelity-Profilbereich typischer Prototypen im Digital Design.....	146
Tabelle 15 - Beziehung zwischen den Dimensionen des Prototyps und den Immersionselementen. „X“ steht für eine starke und „~“ für eine schwache Beziehung.	150
Tabelle 16 - User Interface Paradigmen	167
Tabelle 17 - Merkmale der verschiedenen Ansätze zur Anwendungsentwicklung	178
Tabelle 18 - Die drei Fehlerarten.....	195
Tabelle 19 - Charakterisierung der vier Grundtemperamente.....	209
Tabelle 20 - Wirksamkeit eines Temperaments während des Bauprozesses	218
Tabelle 21 - Temperamente und Fähigkeiten für das digitale Zeitalter.....	220
Tabelle 22 - Rollen, Temperamente und Aufgaben	221
Tabelle 23 - Struktur einer Pitch-Präsentation mit Beispielinhalten von YPRC	239