



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Shahed Elanssary

Die Integration von Design Thinking in das Scrum-Framework

*Fakultät Technik und Informatik
Department Informatik*

*Faculty of Engineering and Computer Science
Department Computer Science*

Shahed Elanssary

Die Integration von Design Thinking in das Scrum-Framework

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Wirtschaftsinformatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stefan Sarstedt

Zweitgutachterin: Prof. Dr. Ulrike Steffens

Eingereicht am: 18.04.2024

Shahed Elanssary

Thema der Arbeit

Die Integration von Design Thinking in das Scrum-Framework

Stichworte

Scrum, Design Thinking, Agilität, DT@Scrum, Design Thinking + Agilität

Kurzzusammenfassung

Diese Arbeit untersucht die Integration von Design Thinking in das Scrum-Framework und bietet eine Analyse sowie Handlungsempfehlungen für Unternehmen, die ein solches Konzept implementieren möchten. Zunächst werden die Grundlagen von Design Thinking und Scrum vorgestellt, danach wird eine detaillierte Untersuchung anhand ihrer Kernkonzepte durchgeführt. Anhand von zwei bereits konzipierten Integrationsansätzen werden Gemeinsamkeiten, Unterschiede und Handlungsempfehlungen herausgearbeitet. Die Auswahl zwischen den Ansätzen hängt von spezifischen Anforderungen und Zielen eines Projekts sowie den vorhandenen Ressourcen ab. Schließlich werden Herausforderungen der Integration diskutiert.

Shahed Elanssary

Title of Thesis

The Integration of Design Thinking into the Scrum Framework

Keywords

Scrum, Design Thinking, agile, DT@Scrum, Design Thinking + agile

Abstract

This thesis examines the integration of Design Thinking into the Scrum framework and provides an analysis and recommendations for action for companies that want to implement such a concept. First, the basics of Design Thinking and Scrum will be introduced, followed by a detailed investigation based on their core concepts. Using two integration approaches that have already been designed, similarities, differences and recommendations for action are identified. The choice between the approaches depends on the specific requirements and objectives of a project as well as the available resources. Finally, challenges of the integration are discussed.

Inhaltsverzeichnis

Inhaltsverzeichnis	iv
Abbildungsverzeichnis.....	vi
Tabellenverzeichnis.....	vii
Abkürzungsverzeichnis	viii
1 Einleitung.....	1
1.1 Ziel dieser Arbeit	1
1.2 Aufbau dieser Arbeit.....	2
2 Design Thinking	4
2.1 Begrifflichkeit.....	4
2.1.1 Design Thinking als Prozess.....	6
2.1.2 Design Thinking als Mindset.....	9
2.1.3 Design Thinking als Toolbox.....	14
2.2 Kernelemente	15
2.2.1 Multidisziplinäre Teams (People).....	16
2.2.2 Variable Räumlichkeit (Place).....	17
2.2.3 Iteratives Vorgehen (Process).....	17
2.3 Vorgehensmodell	17
2.3.1 Phase 1: Verstehen.....	19
2.3.2 Phase 2: Beobachten	20
2.3.3 Phase 3: Standpunkt definieren.....	22
2.3.4 Phase 4: Ideen generieren	24
2.3.5 Phase 5: Prototypen entwickeln.....	26
2.3.6 Phase 6: Testen	26
3 Scrum	29

3.1	Begrifflichkeit.....	29
3.2	Theorie.....	30
3.3	Werte.....	32
3.4	Prinzipien.....	33
3.5	Bestandteile von Scrum.....	35
3.5.1	Rollen.....	35
3.5.2	Ereignisse.....	36
3.5.3	Artefakte.....	37
3.6	Der Scrum-Prozess.....	38
4	Integration von Design Thinking und Scrum.....	40
4.1	Synergien zwischen Design Thinking und Scrum.....	40
4.2	Konzepte und Ansätze zur Integration von Design Thinking in das Scrum-Framework.....	42
4.2.1	Collective Framework DTScrum.....	42
4.2.2	DT@Scrum.....	49
4.3	Vergleich der Ansätze.....	56
4.4	Handlungsempfehlung.....	58
4.5	Herausforderungen und Hindernisse bei der Integration.....	59
5	Fazit & Ausblick.....	64
5.1	Fazit.....	64
5.2	Ausblick.....	67
	Literaturverzeichnis.....	69

Abbildungsverzeichnis

Abbildung 1: Die Schnittstelle zur Innovation (eigene Darstellung) (Gehm, 2022)	5
Abbildung 2: Die Kernelemente des Design Thinkings (eigene Darstellung) (Blatt & Sauvonet, 2017)	16
Abbildung 3: Der Design Thinking-Prozess (eigene Darstellung)	18
Abbildung 4: Double Diamond (eigene Darstellung) (Hehn et al., 2022).....	19
Abbildung 5: Template einer Customer Journey Map (eigene Darstellung) (Lewrick et. al., 2019)	23
Abbildung 6: 2x2 Matrix zum Clustern von Ideen (eigene Darstellung) (Lewrick et. al., 2019)	25
Abbildung 7: Die empirischen Scrum-Säulen (eigene Darstellung).....	31
Abbildung 8: Scrum-Werte (eigene Darstellung) (Schwaber & Sutherland, 2020)	32
Abbildung 9: Das agile Manifest (eigene Darstellung) (Kleuker, 2018).....	33
Abbildung 10: Der Scrum-Prozess (eigene Darstellung) (Schwaber & Sutherland, 2020).....	38
Abbildung 11: Collective process framework (Hehn et al., 2022)	43
Abbildung 12: Scrum Process Area (Hehn et al., 2022).....	45
Abbildung 13: Design Thinking Process Area (Hehn et al. 2022)	46
Abbildung 14: Aktivitäten des Design Thinking Modus (eigene Darstellung)	50

Tabellenverzeichnis

Tabelle 1: Verschiedene DT-Vorgehensmodelle (eigene Darstellung) (Gehm, 2022).....	7
Tabelle 2: Beispiele aus der DT-Toolbox (Lewrick et al., 2019)	15
Tabelle 3: Agile Prinzipien (Häger et al., 2013; Wieczorrek & Mertens, 2010).....	34
Tabelle 4: Vergleichbare Rollen von DT@Scrum und DTScrum.....	57

Abkürzungsverzeichnis

DT Design Thinking

bspw. beispielsweise

bzw. beziehungsweise

1 Einleitung

Die heutige Geschäftswelt wird vom kontinuierlichen Wandel geprägt, wodurch die Nachfrage nach innovativen Lösungen stetig steigt. Unternehmen müssen sich der Herausforderung stellen, Produkte und Dienstleistungen zu entwickeln, die sowohl den Erwartungen der Kunden entsprechen als auch die Marktvolatilität widerspiegeln. In diesem Kontext sind agile Ansätze für die Produktentwicklung von entscheidender Bedeutung.

Agile Methoden haben sich in den letzten Jahren als ein wirksames Mittel zur Bewältigung dieser Herausforderungen etabliert. Scrum, ein weitverbreitetes agiles Framework für die Projekt- und Produktentwicklung, hat sich als besonders einflussreich erwiesen. Es bietet eine strukturierte Methode zur Durchführung von Projekten in kurzen, iterativen Zyklen und betont die Zusammenarbeit im Team, die Transparenz sowie die kontinuierliche Verbesserung.

Ein weiterer vielversprechender Ansatz, der in den letzten Jahren immer mehr an Bedeutung gewonnen hat, ist Design Thinking. Es ist eine kreative und nutzerzentrierte Methode zur Problemlösung, die darauf abzielt, innovative Lösungen durch die Perspektive und das Feedback der Nutzer zu generieren. Diese Methode betont die Bedeutung von Empathie, Ideenfindung und Prototyping.

Während Scrum auf Effizienz, Planung und Umsetzung abzielt, betont Design Thinking die Kreativität, das Verständnis der Nutzer und die Identifikation von unerfüllten Bedürfnissen. Die Frage, die sich stellt, ist, wie diese beiden Ansätze harmonisch integriert werden können, um den bestmöglichen Nutzen aus beiden zu generieren.

1.1 Ziel dieser Arbeit

Aufgrund des kontinuierlichen Wandels der Welt ist es also eine absolute Notwendigkeit, sich als Unternehmen Änderungen schnell anpassen zu können und sich nicht an bestehende

Methodiken fest zu fahren, sondern eine gewisse Offenheit zu bewahren und diese - wenn möglich - zu erweitern. Da die meisten Unternehmen heutzutage Scrum nutzen (Tiba, 2023), ist es sinnvoll, auf diese agile Methodik weiter aufzubauen.

Dementsprechend sollen in dieser Arbeit zunächst die Grundlagen von Design Thinking und Scrum so vorgestellt werden, dass das nötige Fachwissen als Basis gelegt wird. Daraufhin werden zwei bereits konzipierte Ansätze zur Integration von Design Thinking in das Scrum-Framework vorgestellt und miteinander verglichen. Daraus sollte dann eine adäquate Handlungsempfehlung für Unternehmen ausgesprochen werden können.

Zur Thematik der Integration von Design Thinking in das Scrum-Framework existieren bereits einige wissenschaftliche Artikel, die mögliche Kombinationen beider Themen in verschiedenen Kontexten präsentieren. Jedoch wurden diese noch nicht zuvor miteinander verglichen und eine Empfehlung daraus gezogen, welchen der Ansätze man als Unternehmen versuchen sollte bei sich einzusetzen.

Das Ergebnis dieser Arbeit soll als Orientierungshilfe sowie Grundlage für jegliche Personen dienen, die sich mit der Integration von Design Thinking in Scrum auseinandersetzen und in Überlegung sind ein verbündetes Konzept bei sich im Unternehmen einzusetzen.

1.2 Aufbau dieser Arbeit

Im Folgenden werden kurz die zu erwartenden Inhalte der jeweiligen Kapitel nähergebracht.

Diese Einleitung beschäftigt sich mit der Relevanz der Integration von Design Thinking in das Scrum Framework. Weiterführend hebt dieses Kapitel hervor, dass diese Integration bereits in wissenschaftlichen Beiträgen thematisiert wurde, aber keiner zuvor die bereits bestehenden Modelle verglichen und ausgewertet hat, sodass die Notwendigkeit einer wissenschaftlichen Arbeit, welche mit dieser erreicht werden soll, zu dieser Thematik von großer Relevanz ist.

Als nächstes folgt eine Einführung in Design Thinking, in welcher die Begrifflichkeit sowie die verschiedenen Arten dessen näher erläutert werden. Weiterführend werden in diesem Kapitel die Kernelemente sowie einige Vorgehensmodelle, von welchen eines detaillierter beschrieben wird, veranschaulicht.

Daraufhin folgt ein Kapitel, in welchem Scrum vorgestellt und zusammengefasst erklärt wird. Um den Umfang dieses Kapitels nicht zu sprengen, werden nur relevante Informationen zur Erreichung des Ziels dieser Arbeit thematisiert.

Anschließend wird die Integration von DT und Scrum thematisiert. Dieses Kapitel umfasst sowohl die Synergien von Design Thinking und Scrum als auch Konzepte sowie Ansätze zur Integration dieser. An dieser Stelle werden ebenfalls die Ansätze zur Integration miteinander verglichen sowie Handlungsempfehlungen auf dessen Grundlage ausgesprochen. Zum Schluss werden die Herausforderungen und Hindernisse der Integration thematisiert.

Das letzte Kapitel umfasst sowohl das Fazit als auch den Ausblick. An dieser Stelle werden die Kerninhalte der Arbeit zusammengefasst sowie untersucht, ob das Ziel dieser Arbeit erreicht wurde. Weiterführend wird darauf eingegangen, wie Forschende dieses Gebietes diese Arbeit weiterbehandeln können.

Abschließend wäre darauf hinzuweisen, dass an einigen Stellen auf eine tiefgehende Erläuterung explizit verzichtet wird, um den Rahmen dieser Arbeit nicht zu sprengen.

Zur besseren Lesbarkeit wird in dieser Arbeit das generische Maskulinum verwendet. Die in dieser Arbeit verwendeten Personenbezeichnungen beziehen sich – sofern nicht anders kenntlich gemacht – auf alle Geschlechter.

Außerdem werden relevante Fachbegriffe sowie Namen *kursiv* dargestellt, um diese hervorzuheben.

2 Design Thinking

Aufgrund des steigenden Interesses und der Bereitschaft zur Einbindung von *Design Thinking* sowohl im Betrieb als auch in der IT (Dorst, 2010), ist es von großer Bedeutung, dieses Konzept konkret und verständlich zu definieren.

Dieses Kapitel geht zunächst der Frage nach, weshalb man auf eine Herausforderung trifft, beim Versuch Design Thinking (DT) konkret zu definieren. Daraufhin werden die verschiedenen Arten von DT vorgestellt. Anschließend folgen dessen Kernelemente sowie ein konkretes Vorgehensmodell, um das Gesamtbild von DT zu vervollständigen.

2.1 Begrifflichkeit

Eine eindeutige Definition für DT scheint es nicht zu geben (Brenner & Uebernickel, 2016). Auf der Suche nach dieser thematisierte im Jahr 2017 ein amerikanischer Psychologe – Lawrence W. Barsalou – dies in einem wissenschaftlichen Paper mit dem Titel „Define Design Thinking“. Darin setzte er sich damit auseinander, weshalb es so kompliziert ist, DT zu definieren. Er kam zum Entschluss, dass die Herausforderung teils darin liegen könnte, dass sich die Bedeutung von DT aus Sicht des Betrachters variiert. „ (...) [A]ttempting to define design thinking aims to define a moving target.“ (Barsalou, 2017, S. 105). Dementsprechend müsste man nach seiner Ansicht bei jeder Anwendung DT neu definieren.

Ein weiterer wichtiger Aspekt, der die Erläuterung von DT erschweren könnte, ist der Sprachraum, da es aufgrund dessen gelegentlich zu Missverständnissen kommen kann (Brenner & Uebernickel, 2016). Insbesondere bezüglich des Substantivs *Design*. Dieses nimmt im deutschen Sprachgebrauch eher einen Bezug auf gestalterische Merkmale eines Produktes (Gerstbach & Gerstbach, 2020). Die englische Definition des Begriffs hingegen ist grober erfasst worden, sodass das Verb „to design“ von den Oxford Advanced Learner’s Dictionary

(o. D.) unter anderem folgendermaßen definiert wird: „do or plan (something) with a specific purpose in mind“. Aufgrund dieser verschiedenen Auffassungen kommt es bei DT-Projekten gelegentlich zum (Miss-)Verständnis, dass es sich um klassische Projekte handle, dessen Durchführung zur ästhetischen Verbesserung dienen (Brenner & Uebernicketl, 2016).

Um diese Missverständnisse zu vermeiden, werden oft die verschiedenen, bereits vorhandenen und teils sehr vereinfachten Definitionen bekannter Designagenturen wie bspw. IDEO präsentiert und auf Basis der Gemeinsamkeiten der Definitionen sowie deren Ziele eine neue Erläuterung geschaffen.

Gehm (2022) folgte diesem Beispiel und definierte DT als einen Ansatz, der es ermöglicht, für komplexe Probleme neue Lösungen zu entwickeln, wobei sich diese konsequent sowohl an den zukünftigen Nutzerbedürfnissen orientieren als auch die technische Machbarkeit sowie wirtschaftliche Umsetzbarkeit mit einbeziehen (s. Abb. 1).

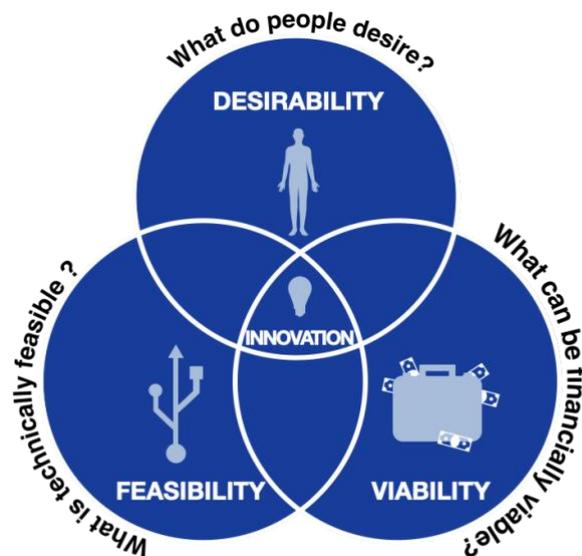


Abbildung 1: Die Schnittstelle zur Innovation (eigene Darstellung)
(Gehm, 2022)

Hier ist hervorzuheben, dass der Fokus von DT auf eine frühe Nutzereinbindung im Verlauf des Projekts liegt; da der Mensch immer den Ausgangspunkt darstellt, sodass die Technologie und das wirtschaftliche Ziel vorerst nach hinten rücken (Mueller-Rotenberg, 2018). Diese Bereiche müssen jedoch nicht immer ausgeglichen sein, da DT kein linearer Prozess ist. Das DT-Team muss im Prozesszyklus die Entscheidung treffen, auf welchem Aspekt der Fokus

liegt, wobei der Mensch kontinuierlich im Mittelpunkt steht (Gerstbach, 2016). Dies wird durch die Verbindung dieser drei Teilbereiche ermöglicht, da der Prozess immer mit der Fragestellung beginnt, „wann eine bestimmte Lösung für eine bestimmte spezifische Personengruppe attraktiv ist“ (Eppler & Hoffmann, 2012, S. 5).

Konkret kann DT vereinfacht als ein Innovationsprozess sowie Denkansatz zur Errichtung von nutzerorientierten Produkten, Dienstleistungen, Lösungen und Erfahrungen definiert werden (Gerstbach & Gerstbach, 2020).

Diese beiden Definitionen dienen als erste Basis der DT-Grundlagen, sodass der weitere Verlauf dieser Arbeit auf diese aufbaut und mit diesem Verständnis voranschreitet.

Laut Brenner und Uebernickel (2016) haben drei Formen des DTs an Bedeutung gewonnen:

- 1) Design Thinking als Prozess bzw. Methodik,
- 2) Design Thinking als Mindset,
- 3) Design Thinking als Toolbox.

Im Folgenden wird konkreter auf diese Arten des DTs eingegangen, da die Kombination aller drei in einer erfolgreichen Anwendung münden kann.

2.1.1 Design Thinking als Prozess

Leonard Bruce Archer (1966) war einer der ersten Designer, der DT als eine systematische Methode beschrieb (Auernhammer & Roth, 2023). Mittlerweile gibt es jedoch mehrere anerkannte systematisch Vorgehensmodelle. Alle haben gemeinsam, dass sie einen iterativen und auf den Nutzer spezialisierten Prozess umfassen. Dabei unterscheiden sie sich meist nur in der Reihenfolge sowie Aufteilung der Phasen und dessen Aufgaben (Gehm, 2022). In der folgenden Tabelle werden einige dieser Vorgehensmodelle zusammengefasst dargestellt.

Tabelle 1: Verschiedene DT-Vorgehensmodelle (eigene Darstellung) (Gehm, 2022)

Autor	Anzahl der Phasen	Namen & Aufgaben der Phasen
Brown (2008)	3	<ul style="list-style-type: none"> (1) Inspiration <ul style="list-style-type: none"> a. Problemdefinierung b. Beobachtung des Nutzers & seiner Umgebung c. Berücksichtigung der Einschränkungen des eigenen Unternehmens (2) Ideation <ul style="list-style-type: none"> a. Erarbeitung der Prototypen b. Testen der Prototypen (3) Implementation <ul style="list-style-type: none"> a. Umsetzung des Prototyps b. Erarbeitung einer Kommunikationsstrategie & eines Business Cases
Plattner et al. (2009)	6	<ul style="list-style-type: none"> (1) Verstehen <ul style="list-style-type: none"> a. Beschreibung der Aufgabenstellung b. Definierung des Problems (2) Beobachten <ul style="list-style-type: none"> a. Aneignung des Expertenwissens (3) Sichtweise definieren <ul style="list-style-type: none"> a. Priorisierung & Zusammenfassung gewonnener Erkenntnisse (4) Ideen finden <ul style="list-style-type: none"> a. Sammlung von Ideen (5) Prototypen entwickeln <ul style="list-style-type: none"> a. Entwicklung von Prototypen (6) Testen <ul style="list-style-type: none"> a. Testen der Prototypen an dem Nutzer
d.school (2010)	5	<ul style="list-style-type: none"> (1) Empathize <ul style="list-style-type: none"> a. Beobachtung der Nutzer & deren Verhalten (2) Define <ul style="list-style-type: none"> a. Definierung der Herausforderung b. Entwicklung eines konkreten Nutzerverständnis (3) Ideate <ul style="list-style-type: none"> a. (s. Plattner et al. (2010) (4)) (4) Prototype <ul style="list-style-type: none"> a. (s. Plattner et al. (2010) (5)) (5) Test <ul style="list-style-type: none"> a. (s. Plattner et al. (2010) (6))

<p>Liedtke und Ogilvie (2011)</p>	<p>4</p>	<ul style="list-style-type: none"> (1) What <i>is</i>? <ul style="list-style-type: none"> a. Analyse des Nutzers & Problems b. Anwendung einer „Value Chain Analysis“ zur Prüfung der Umsetzbarkeit durch das Unternehmen (2) What <i>if</i>? <ul style="list-style-type: none"> a. Gestaltung der Zukunft (3) What <i>wows</i>? <ul style="list-style-type: none"> a. Überprüfung der erfolgsversprechenden Konzepte anhand von Annahmen (4) What <i>works</i>? <ul style="list-style-type: none"> a. Weiterentwicklung der Prototypen mithilfe der Nutzer b. Planung der Markteinführung
-----------------------------------	----------	--

Der derzeit in Europa verbreitete DT-Prozess nach Plattner et al. (2009) wird iterativ durchlaufen; das heißt im Laufe des Projekts werden oftmals Rücksprünge getätigt, die ein kontinuierliches Verfeinern der Ergebnisse oder Korrigieren dieser ermöglicht (Gehm, 2022). Die ersten drei Phasen werden als Problemraum bezeichnet, die letzteren als Lösungsraum (Lindberg et al., 2010) (s. Abb. 3): Ersteres umfasst die Beschreibung sowie Ursache des Problems, wobei zweiteres sowohl die Sammlung als auch Umsetzung der Lösungen thematisiert. Dies sowie die einzelnen Phasen werden im Abschnitt 2.3 vertieft erläutert.

Anhand der Tabelle wird erkenntlich, dass das Vorgehensmodell der d.school (2010) dem des Hasso-Plattner-Instituts (2009) ähnelt. Dies liegt an deren engen Verbindung (Gehm, 2022). Beide unterscheiden sich von den Bestandteilen darin, dass die ersten drei Phasen des sechsteiligen Ablaufs von Plattner in den ersten zwei des fünfteiligen Prozesses von der d.school zusammengefasst werden. Letztlich geschieht im Ganzen jedoch genau dasselbe, sodass kein signifikanter Einfluss auf die Ergebnisse oder den Erfolg des DT-Prozesses besteht.

Der DT-Prozess wechselt zwischen Phasen des *divergenten* und *konvergenten Denkens*, zwei häufig verwendete Konzepte der Kreativitätsforschung (Gebbing et al., 2021; Sternberg, 2019). An dieser Stelle werden diese zwei Phasen des kreativen Denkens genauer erläutert.

Divergentes Denken umfasst die Suche nach zahlreichen Lösungsmöglichkeiten, sodass es dem *Quantität-vor-Qualität*-Prinzip folgt (Linke, 2020). Somit werden während dieser Denkphase diverse Ideen zur kreativen Ideengenerierung ermittelt (Gebbing et al., 2021). Genauso unterstützt es das Einnehmen vielseitiger Sichtweisen und das Identifizieren von unterschiedlichen Möglichkeiten (Gehm, 2022).

Dieser Denkphase wird das **konvergente Denken** gegenübergestellt. Sie umfasst die konkrete Lösungsfindung eines Problems (Gebbing, et al., 2021). Im DT geht man vom divergenten Zustand in die konvergente Phase über, sodass man aus den zahlreich gesammelten Lösungsmöglichkeiten nun einige Vorschläge auswählt, diese priorisiert sowie den Fokus auf eine Teilmenge setzt (Gehm, 2022).

Selbstverständlich wirken diese Denkweisen sich auf das DT-Mindset aus, welches im folgenden Abschnitt thematisiert wird.

2.1.2 Design Thinking als Mindset

Mittlerweile hat es sich etabliert, dass DT nicht ausschließlich als eine Methodik oder ein Prozess angesehen werden sollte, da zum erfolgreichen Einsetzen und Anwenden dessen ein gewisses DT-Mindset adaptiert werden sollte (Gehm, 2022). Dieses wird durch bestimmte Grundprinzipien, die eine Kombination aus divergentem sowie konvergentem Denken darstellen, charakterisiert (Brenner & Uebernicket, 2016).

Somit zeichnet sich DT als Mindset durch die folgenden Grundprinzipien aus, die essenziell für den Erfolg der Methodik sind und somit von allen Teammitgliedern gelebt werden sollten (Brenner & Uebernicket, 2016; Gehm, 2022).

Laut Schweitzer, Groeger & Sobel (2016) gehören die folgenden Prinzipien zum DT Mindset:

(1) Empathetic towards People's Needs and Context

Der Mensch steht beim DT mit seinen Bedürfnissen sowie Erfahrungen an erster Stelle und gilt als Ausgangspunkt aller Überlegungen (Brenner & Uebernicket, 2016; Gehm, 2022; Mueller-Rotenberg, 2018). Daher spielt die Empathie eine wichtige Rolle. Empathie ist die Fähigkeit, etwas aus der Perspektive eines anderen zu sehen und zu erleben, um ein besseres Verständnis darüber zu erhalten, warum Menschen das tun, was sie tun (Kelley & Kelley, 2013). Sie ermöglicht dem DT-Team den sozialen Kontext und die Erkenntnisse der (potenziellen) Nutzer zu verstehen, indem ihr Fokus auf den Menschen gesetzt wird (Michlewski, 2008).

(2) Collaboratively Geared and Embracing Diversity

Die Bildung interdisziplinärer Teams in DT-Projekten ist essenziell. Genauso wichtig ist es, dass einzelne Teilnehmer kollaborativ ausgerichtet sind. Das heißt, sie besitzen die Fähigkeiten, sich leicht in neue Teams zu integrieren, Teamdynamiken zu untersuchen und auch zu konfrontieren. Weiterhin akzeptieren sie die Persönlichkeit, die Expertise und den Arbeitsstil jedes einzelnen Teammitglieds als essenzielle Voraussetzung, um aus der multidisziplinären Zusammenarbeit Vorteile zu ziehen. Die Zusammenarbeit und der Wissensaustausch der Teilnehmer fördern zusätzlich zur Empathie eine schnelle Problemlösung sowie die Entwicklung neuer Ideen (Hassi & Laakso, 2011).

(3) Accepting of Uncertainty and Open to Risk

DT-Praktiker werden im Verlauf der DT-Projekte mit Situationen konfrontiert, in denen sie auf Grundlage einer zukünftigen potenziellen Lösung Entscheidungen treffen müssen, anstelle von einer bereits funktionierenden Lösung. Dies bringt ein gewisses Risiko mit sich, das durch die Angst des Scheiterns noch verstärkt wird (Kelley & Kelley, 2013; Michlewski, 2008). Abgesehen davon, dass das Team ohne garantierte Ergebnisse in einen Designprozess einsteigt, sind die Zeit sowie das Volumen der Iterationen zur erforderlichen Verfeinerung des Designergebnisses umso weniger bekannt oder garantiert. Weswegen DT-Praktiker müssen solche Unsicherheiten über längerem Zeitraum eher akzeptieren und annehmen (Hassi & Laakso, 2011; Kelley & Kelley, 2013). Wenn diese Unsicherheit als selbstverständlich akzeptiert wird, ist das Team in der Lage vielzählige Faktoren gleichzeitig zu berücksichtigen. Dieses können bspw. Kundenbedürfnisse, technologische Machbarkeit, strategische Auswirkungen und Kosten sowie Vorteile verschiedener Lösungsvorschläge sein (Jacoby & Rodriguez, 2007; Johannsson & Woodilla, 2009).

(4) Inquisitive and Open to New Perspectives and Learning

Trotz der Bewältigung der hohen Unsicherheit von DT, verfolgen die Praktiker einen Entdeckungs- sowie Lernprozess, indem sie erforschen, experimentieren, testen und sich Feedback von mehreren Stakeholdern einholen (Plattner et al., 2012). Hinter so einem Ausdauervermögen stecken viel Neugierde sowie Lernbegierde, welche im Verlauf des

Projektes vorhanden sind, um den Kunden und Markt präziser zu verstehen (Kelley & Kelley, 2013).

Dieser Informationssammelungsprozess setzt voraus, dass DT-Praktiker offen für neue Perspektiven sind (Brown, 2009). Doch auch wenn die Ergebnisse nicht ihren Annahmen entsprechen und bspw. die Prototypen nicht überzeugen, erfassen sie wertvolle Daten zum Iterieren der Lösung. Dieser Prozess der Generierung und Entwicklung neuer Annahmen sowie Ideen erfordert einen wissbegierigen, offenen und positiven Geist, sodass die Verarbeitung dieser Misserfolge und dem Feedback einen neuen Ausgangspunkt zur besseren Lösungsfindung darstellt. Dieser Lernprozess stellt einen zentralen Bestandteil der Iteration zwischen den divergenten und konvergenten Phasen des DT-Prozesses dar.

(5) Mindful of Process and Thinking Modes

Die Achtsamkeit gegenüber dem Prozess und der Denkweise zeigt, dass man sich der Arbeit bewusst ist, die man verrichtet, wie man diese durchführt, warum man sie auf eine gewisse Art und Weise ausführt und wie man die angewendeten Methoden verbessern kann. Achtsamkeit umfasst, das Verständnis darüber, in welcher Phase des Designprozesses sich das Team befindet und welche Verhaltensweisen sowie Ziele es zu einem bestimmten Zeitpunkt haben könnte. Vor allem ist dieses Mindset von großer Bedeutung, wenn sich das Team darüber im Klaren sein soll, wann es äußerst generativ sein muss und wann es sich auf einen bestimmten Lösungspfad konzentrieren sollte.

Bei der Durchführung eines Designprozesses wird vorausgesetzt, dass die Teilnehmer zu unterschiedlichen Zeitpunkten divergentes sowie konvergentes Denken anwenden können (Brown & Katz, 2011). Diese Fähigkeit und das Wissen darüber, wann und wie man divergent denkt, bevor man sich auf eine Möglichkeit festlegt, zeichnet DT-Praktiker aus (Brown & Katz, 2011).

(6) Experiential Intelligence

Erfahrungsbildung umfasst die Vorliebe für das Ausprobieren von Ideen, indem man Mock-Ups anfertigt. Dabei werden Modelle gebaut sowie etwas Greifbares zum Experimentieren erzeugt, um Ideen in etwas Erlebbares und Testbares zu verwandeln. Somit wird ermöglicht sowohl immaterielle Ideen als auch Konzepte in greifbare

Ergebnisse umzusetzen (Clark & Smith, 2008). Zusätzlich ist in der DT-Literatur oft die Rede von der Wichtigkeit Daten visuell zu präsentieren, Geschichten zu erzählen sowie Tests anhand von physischen Prototypen durchzuführen. Grund dafür ist das Verständnis zu schärfen, die Bedeutung zu kommunizieren oder Feedback zu testen sowie zu fördern. Dieser Prozess wird oft als *Visual Thinking* bezeichnet (Kelley & Kelley, 2013).

(7) Taking Action Deliberately and Overt

Handlungsorientierung wird dadurch ausgezeichnet, dass man handlungsorientiertes Verhalten einer Diskussion und konzeptuellen sowie analytischem Verhalten vorzieht. Um das Team zum neuen Denken anzuregen, wird es bevorzugt in die reale Welt hinauszugehen und Nutzer einzubinden, Prototypen zu erstellen sowie Ideen zu testen. Das soll zu dem Ergebnis führen, dass Entscheidungen nicht ausschließlich auf Diskussionen oder Denkprozessen basieren, sondern auch auf Erfahrungen sowie Experimenten aus erster Hand. Außerdem bezieht sich die Handlungsorientierung auch auf die Fähigkeit, große Aufgaben zu analysieren und einen Ausgangspunkt zu wählen. Mit dieser Vorgehensweise kann man beispielsweise unverzüglich eine der möglichen Lösungen als Prototypen umzusetzen, um möglichst schnell Feedback vom Nutzer zu dieser einzuholen. Somit dient dieses Mindset als ein Werkzeug um Veränderungen zu beeinflussen, Handlung zwischen Teams und anderen zu initiieren, positives und produktives Verhalten zu demonstrieren sowie das Arbeiten an und lösen komplexer Probleme (Kelley & Kelley, 2013).

(8) Consciously Creative

Professionelle DT-Anwender sind mit kreativen Methoden, Denkweisen sowie Ausdrucksformen vertraut, da sie bewusst einen kreativen Arbeitsansatz verfolgen. Kreativität ermöglicht das Erkundigen und Ausdrücken von weniger greifbaren und eher subjektiven Inhalten, indem es das Abstrakte zum Leben erweckt. Somit ist es besonders wichtig ein gewisses Verständnis von Kreativität zu besitzen und zu verstehen, wie kreatives Denken für andere funktioniert (Kelley & Kelley, 2013). Bestandteile des kreativen Verständnisses sind die Anerkennung von Fehlern, Minimierung von Hierarchien, Förderung der Ideen Anderer sowie der Vorstellungskraft und die Inspiration

sowie Motivation von Menschen (Badke-Schaub et al., 2010; Miller & Moultrie, 2013). Der Kreativitätsprozess erfordert, dass man eine gewisse Zeit Ideen reifen lässt, um Zusammenhänge zu formulieren. Dies wird von Kelley und Kelley (2013) als *engaged relaxed attention* bezeichnet.

(9) Modelling Behaviour

DT-Praktiker bringen unterschiedliche Menschengruppen zusammen. Dies ist wichtig für den DT-Prozess, da vieles für die Beteiligten neu sein kann. Daher dienen sie als „advocates for creativity and design“ (Michlewski, 2008, S.16), indem sie bei der Prozessdurchführung sowohl positives Verhalten als auch eine optimistische Einstellung vorleben. Da DT sehr kollaborativ ausgelegt ist, besitzen dessen Experten die Fähigkeit, Diskussionen des Teams zu leiten. Dabei legen sie ihr eigenes Ego beiseite und beruhigen das Ego Anderer. Dies stellt sicher, dass alle Ideen sowie Erkenntnisse ohne Urteil sowohl eingebracht als auch gleichermaßen bewertet werden (Brown, 2008; Kelley & Kelley, 2013, Hassi & Laakso, 2011). Somit sind die DT-Praktiker Experten darin, Ideen zu kommunizieren und das Kommunizieren der Ideen anderer zu fördern. Modelling Behaviour umfasst den Aufbau der DT-Fähigkeiten sowie des Vertrauens in andere, anhand der allmählichen Auseinandersetzung und Intensivierung der Herausforderungen über einen bestimmten Zeitraum. Ein weiteres wichtiges Merkmal ist, die Fähigkeit, bedingungslosen Optimismus zu zeigen, welcher von Kelley und Kelley (2013) als *urgent optimism* bezeichnet wird. Es ist nämlich der Optimismus, der aus Ideen Projekte realisiert.

(10) Desire and Determination to Make a Difference

Zusätzlich zum Optimismus der DT-Praktiker kommt das Bedürfnis und der Wunsch danach, etwas zu bewirken, Situationen zu verbessern sowie die Entschlossenheit, Ideen zu verwirklichen. Dieses Verlangen ist eine essenzielle Eigenschaft, da sie das Potenzial für Veränderungen erkennt, vor allem insofern wie und wo diese sinnvoll wären. Praktiker streben danach, ein Katalysator für Transformation zu sein, wofür sie zielgerichtete Disziplin sowie eine starke Stimme der Absicht aufweisen müssen (Kelley & Kelley, 2013). Es ist klar, dass DT ein hohes Maß an Entschlossenheit benötigt und Praktiker somit die Fähigkeit besitzen müssen, die vorgestellten Ergebnisse denen zu präsentieren, die

noch nicht überzeugt sind. Dementsprechend müssen sie widerstandsfähig sein und geschickt mit Überzeugung sowie Verhandlung umgehen können (Martin, 2010).

(11) Critically Questioning

DT bezieht den Input vieler Stakeholder ein. Daher ist es von großer Bedeutung, dass der DT-Anwender ein Mindset des kritischen Hinterfragens besitzt und auf Ideen anderer aufbaut (Kelley & Kelley, 2013). Darunter versteht man die Fähigkeit, offen für jeglichen Ideenansatz, besonders in frühen Phasen zu sein. Somit wird sichergestellt, dass gute Ideen überleben und Ergebnisse weiterentwickelt werden können. Diese Denkweise ermöglicht Projektteams sich nicht von alten Denkweisen leiten zu lassen (Bolland & Collopy, 2004; Martin, 2009). Anhand dessen wird der Fokus im Projekt- und Prozessverlauf auf das ursprüngliche Designproblem gelegt, sodass das Team immer vor Augen hat, worauf sie hinarbeiten (Kelley & Kelley, 2013).

DT zeichnet sich als Denkweise, der gleichzeitig spielerischen und lösungsorientierten Kombination dieser Prinzipien aus. Erfahrene Anwender erkennen diese Prinzipien als gegeben und nutzen sie, um innovative und kundenorientierte Lösungen zu schaffen (Brenner & Uebernicket, 2016).

2.1.3 Design Thinking als Toolbox

Nicht zuletzt wird DT auch als Toolbox angesehen, welche diese in eine Vielfalt von Techniken und Werkzeugen unterteilt. Basierend auf die Situation sowie den Kontext können diese im Rahmen des Projekts eingesetzt werden (Hehn et al., 2022).

Aufgrund des steigenden Interesses an DT, entstanden in den vergangenen Jahren zahlreiche Praktikerkataloge, die sich mit DT-Methoden sowie Tools beschäftigen (d.school, 2010; Hehn et al., 2022; IDEO.org, o. D.; Uebernicket et al., 2020). Die breite Werkzeug-Palette kann das Team im Laufe des Prozesses dabei unterstützen Probleme besser zu verstehen, Ideen zu generieren und Prototypen zu entwickeln. Aufgrund der umfangreichen Toolbox wird die Annahme verstärkt, dass DT keinen präskriptiven Prozess mit eindeutigen Phasen darstellt, sondern eher eine Sammlung praktischer Methoden sowie Techniken zur kontextabhängigen Unterstützung umfasst (Hehn et al., 2022).

In Tabelle 2 befinden sich Beispiele aus der Toolbox für DT-Werkzeuge:

Tabelle 2: Beispiele aus der DT-Toolbox (Lewrick et al., 2019)

Werkzeug	Beschreibung
Empathy Mapping	Entwicklung eines tiefen Verständnisses für die Bedürfnisse & Emotionen der Nutzer, indem ihre Perspektiven, Ziele & Herausforderungen visualisiert werden.
Brainstorming	Kreative Technik zur Generierung von zahlreichen Ideen, indem Teammitglieder spontan & ohne Einschränkungen Ideen äußern.
Prototyping	Erstellung von schnellen, einfachen Modellen oder Prototypen zur Visualisierung von Ideen & schnellen Einholung von Nutzerfeedback
User Journey Mapping	Visuelle Darstellung der Interaktion eines Nutzers mit einem Produkt oder einer Dienstleistung über verschiedene Phasen hinweg.
Ideation Sessions	Geplante Sessions zur kreativen Ideengenerierung, die Teilnehmern ermöglicht zusammenzukommen, um gemeinsam zu brainstormen & innovative Lösungen zu entwickeln.
Rapid Prototyping	Schneller Entwicklungsansatz zur Erstellung von Prototypen mit minimalen Ressourcen, um Ideen zu testen & iterativ zu verbessern.

Dies sind nur einige der vielzähligen Beispiele der umfangreichen DT-Toolbox. Je nach Kontext sowie Anforderungen eines Projekts können verschiedene Methoden und Werkzeuge eingesetzt werden, um den Prozess zu unterstützen und so die bestmöglichen Ergebnisse zu erzielen (Lewrick et al., 2019).

2.2 Kernelemente

DT besitzt drei Kernelemente, die für den Erfolg von Projekten entscheidend sind (s. Abb. 2) (Blatt & Sauvonnnet, 2017; Brown, 2016; Plattner et al., 2009; Uebernicket & Brenner, 2015). Diese werden im englischen Sprachgebrauch auch als *three Ps of Design Thinking* bezeichnet

(Curedale, 2013): Sie stehen für *People*, *Place* sowie *Process* und werde in diesem Abschnitt näher erläutert.

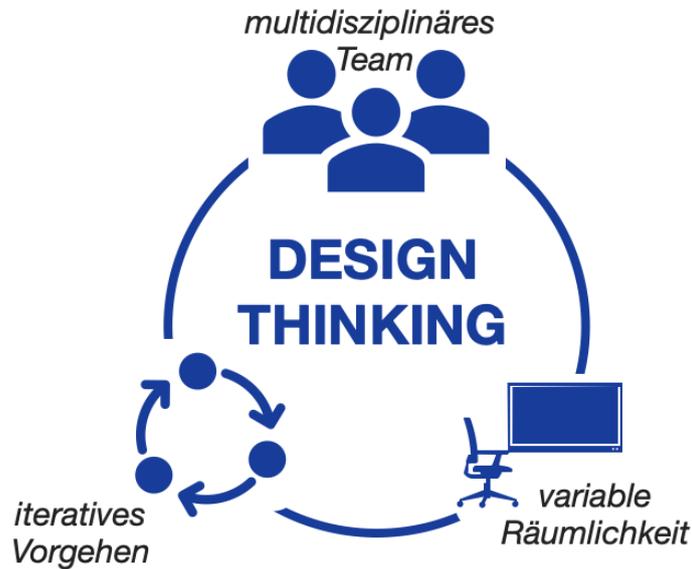


Abbildung 2: Die Kernelemente des Design Thinkings
(eigene Darstellung) (Blatt & Sauvonnnet, 2017)

2.2.1 Multidisziplinäre Teams (People)

Das Projektteam erzielt dann die besten Ergebnisse, wenn es aus mehreren Spezialisten mit unterschiedlichen Fähigkeiten zusammengesetzt ist. Dadurch werden diverse Sichtweisen auf die Problemstellung eingebracht, sodass multidisziplinäre Teams ausgefallenerere Ergebnisse bei deutlich höherer Produktivität erzielen als homogene Teams (Gehm, 2022). Nach Grots und Pratschke (2009) müssen Menschen, die diese multidisziplinären Teams bilden, gewisse Eigenschaften und ein sogenanntes *T-Profile* vorweisen. Der vertikale Balken des Buchstabens T (|) soll fachspezifisches und analytisches Wissen widerspiegeln, wobei der horizontale Balken dessen (-) für entscheidendere Fähigkeiten wie bspw. Neugier, Offenheit und Intuition steht (Blatt & Sauvonnnet, 2017; Gehm, 2022).

2.2.2 Variable Räumlichkeit (Place)

Um der Kreativität freien Lauf zu lassen, ist es von essenzieller Bedeutung alltägliche Strukturen aufzubrechen und somit neuen Raum für Ideen zu schaffen. Eine variabel eingerichtete Räumlichkeit besteht aus Mobiliar, das Mobilität, Flexibilität und Einfachheit fördert und somit die Arbeit des Teams unterstützt. An dieser Stelle kann Gebrauch von bspw. Trennwänden gemacht werden, um den Raum bei Bedarf zu verkleinern oder zu vergrößern. Genauso gut sind mit Rollen ausgestattete Möbel wie Stühle, Tische und Tafeln, die die Teilnehmer beliebig und ihren Wünschen nach ausrichten können. Ein solches Raumkonzept trägt dazu bei, die Teamarbeit zu fördern, demzufolge müssen also Nähe und Gemeinschaft aufeinander abgestimmt werden (Blatt & Sauvonnet, 2017).

2.2.3 Iteratives Vorgehen (Process)

DT hat keinen vorgeschriebenen Lösungsansatz, sondern bietet eine sogenannte Toolbox an, mit vielen Methoden und Werkzeugen, auf die Teilnehmer zurückgreifen können. Somit ist es ihnen überlassen, mit welchen Herangehensweisen, sie zum Ziel gelangen möchten (Blatt & Sauvonnet, 2017).

Nichtsdestotrotz bedarf DT für den Erfolg eine klare Systematik, die in Form eines strukturierten Arbeitsprozesses erkennbar wird. Dieser verbindet sowohl analytische als auch intuitive Arbeitsweisen miteinander. Beispiele für analytische Arbeit wären Informationen sammeln, ordnen und auswerten, wobei die Arbeit im intuitiven eher Lösungen entwickeln und Annahmen formulieren umfasst (Blatt & Sauvonnet, 2017).

Konkret handelt es sich um ein Vorgehensmodell, das aus Projektphasen besteht, welche aufeinander abgestimmt sind sowie „iterativ miteinander verbunden sind und in ihrer Reihenfolge unbedingt eingehalten werden sollten“ (Plattner et. al., 2009, S. 127).

2.3 Vorgehensmodell

Vorweg ist es von essenzieller Bedeutung klarzustellen, dass, wie im ersten Kapitel erwähnt, DT nicht ausschließlich ein Prozess ist, sondern sowohl eine Denkhaltung als auch Toolbox, sodass die Kombination dieser drei notwendig ist zum Erfolg eines Projekts.

Des Weiteren sollte der Prozess nicht in den Mittelpunkt gestellt werden, ansonsten ist man dem Risiko ausgesetzt, dass DT nicht die erwünschte Wirkung erzielt (Gehm, 2022). Es ist also wichtig, dass die existierenden Vorgehensmodelle als grob gefasste Wegweiser angesehen werden (AJ&Smart, 2020). Dies kann man sich so vorstellen, dass die Ziele dieser Phasen zwar feststehen, aber wie das Team dahin gelangt, und welche Tools verwendet werden, um diese zu erreichen, ist ganzheitlich den Teilnehmern überlassen. Sie können und sollen diese Modelle beliebig dem Anwendungsfall anpassen.

Im Laufe dieses Kapitels sowie dieser Arbeit liegt der Fokus auf ein spezifisches Modell des Hasso-Plattner-Instituts, welches mit sechs Phasen (s. Abb. 3) ausgestattet ist. Diese Phasen werden nachfolgend detailliert erläutert.

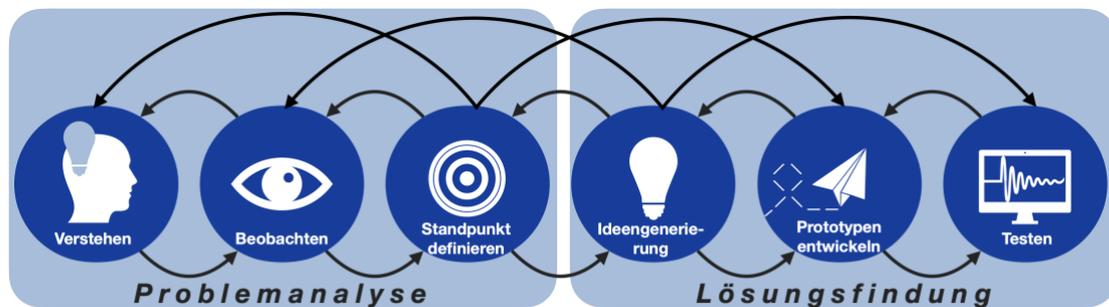


Abbildung 3: Der Design Thinking-Prozess (eigene Darstellung)

Vorerst jedoch wurden im ersten Kapitel der abgebildeten Problem- als auch der Lösungsraum angesprochen, sodass diese an dieser Stelle erneut aufgegriffen werden, da das Vorgehensmodell darauf aufbaut. Diese Unterteilung basiert nämlich auf dem *Double Diamond*-Ansatz (s. Abb. 4).

Im Rahmen des Problembereichs wird der Kontext des Problems sorgfältig untersucht, indem alle relevanten Interessentengruppen einbezogen werden und alle gesammelten Informationen zu einem klaren Standpunkt zusammengeführt werden, wobei dieser auch die Bedürfnisse sowie Erkenntnisse umfasst (Hehn et al., 2022). In der Abbildung vier wird dies durch die Visualisierung der Dreiecke deutlich: die gesammelten Informationen werden durch die Basis des Dreiecks dargestellt, während die rechte Spitze den Standpunkt repräsentiert.

Im Lösungsbereich hingegen wird der Standpunkt aufgegriffen und die Generierung von Ideen sowie die Erstellung von Prototypen gefördert, sodass dies die Basis des Dreiecks abbildet.

Diese Prototypen werden dann von Benutzern bewertet sowie getestet, sodass die daraus resultierende Aussortierung der unbrauchbaren Prototypen das Team näher ans Ziel führt, welches durch die Dreiecksspitze vertreten wird.

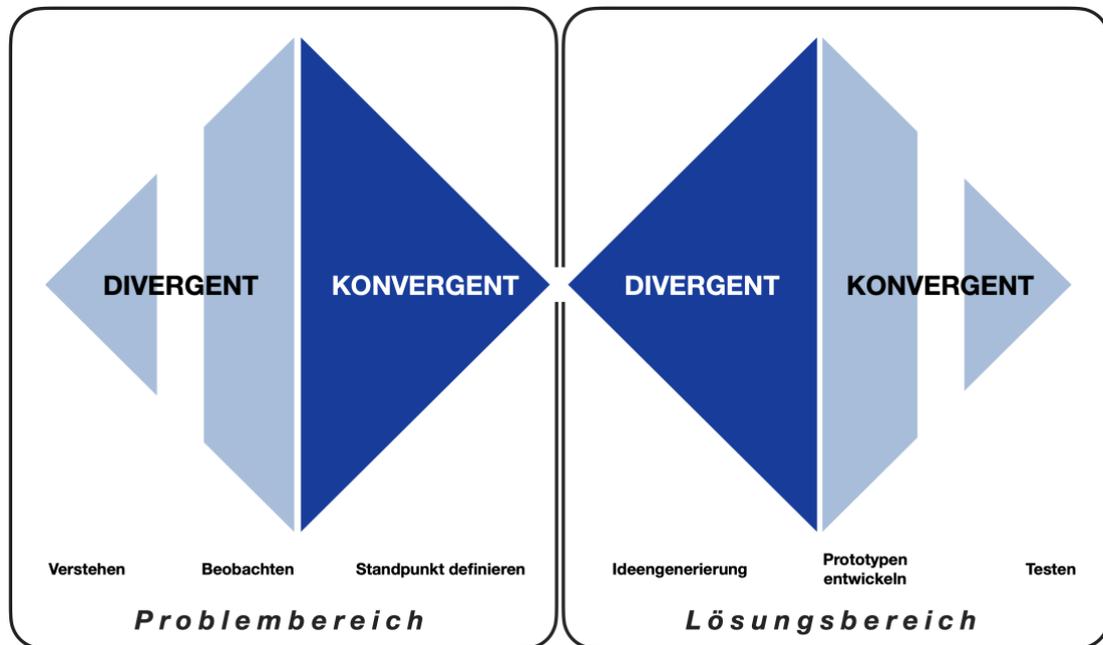


Abbildung 4: Double Diamond (eigene Darstellung) (Hehn et al., 2022)

Dieser gesamte Prozess wird mehrmals durchlaufen, bis eine endgültige Lösung entwickelt wurde. Nun stellt sich die Frage, was in dem Prozess passiert. Dies wird mit dem folgenden Abschnitt anhand der sechs Phasen ausführlich thematisiert.

2.3.1 Phase 1: Verstehen

DT-Projekte starten in der Regel mit der Verstehen-Phase, in welcher sich das Team ein gemeinsames Verständnis über die Ausgangslage erarbeitet. In dieser ist es von großer Bedeutung, dass sich das Team mit dem Themengebiet ausführlich auseinandersetzt und das Problem versteht. Teilnehmer sollte in diesem Schritt nicht an Lösungen denken, denn erst wenn die unbewussten und versteckten Nutzerbedürfnisse verstanden worden sind, ist der Raum für die Entwicklung von innovativen und erfolgreichen Produkten wie Dienstleistungen gegeben (Simschek & Kaiser, 2019). Grundlegend ist somit eine unvoreingenommene sowie offene Sichtweise des möglichen Problemfeldes (Gehm, 2022).

In der Toolbox gibt es mehrere Werkzeuge, die an dieser Stelle zum Einsatz kommen können, wie bspw. Expertengesprächen. Diese bieten dem Team eine sehr effiziente Variante auf den aktuellen Stand zu kommen. Methoden wie Brainstorming helfen dem Team zahlreiche Informationen in kürzester Zeit zu sammeln (Simschek & Kaiser, 2019). Sich anhand von wissenschaftlicher Literatur in das Thema einzulesen, schafft ebenfalls eine fundierte Grundlage. Verallgemeinert kann also gesagt werden, dass jede Recherchemethode, die die Teilnehmer effizient und schnell zu *Sofortexperten* in dem Themenfeld ausbildet, geeignet ist (Gehm, 2022).

Anschließend muss das Team die richtige Frage identifizieren sowie formulieren. Diese wird auch als „Design-Thinking-Challenge“ bezeichnet. Bei der Formulierung dieser steht der Nutzer mit seinen Bedürfnissen im Mittelpunkt. Die Teilnehmer müssen darauf achten, dass die Frage einerseits konkret genug formuliert worden ist, um fokussiert an dem Thema arbeiten zu können, andererseits sollte sie offen genug sein, sodass keine Lösungen aus der Fragestellung zu implizieren sind (Gehm, 2022).

Durch den iterativen Prozess sollte es jedoch klar sein, dass die Ausgangsfrage nicht final formuliert ist, sondern immer wieder verändert werden kann und sollte, bis das Team der Meinung ist, den richtigen Fokus gefunden zu haben (Gehm, 2022).

Ein weiterer und essenzieller Teil dieser Phase ist, wie zu Beginn jedes Projektes, eine sorgfältige Projektplanung. Diese sollte wichtige Rahmenbedingungen wie unter anderem Termine, Kosten und Kapazitäten der Teilnehmer beinhalten (Gehm, 2022).

Ebenso setzen sich die Teilnehmer in dieser Phase mit Teambuilding auseinander, indem sie Werte und Normen festhalten, nach denen im Laufe des Projektes gearbeitet werden soll (Gehm, 2022).

2.3.2 Phase 2: Beobachten

In der Beobachten-Phase liegt der Fokus auf das tiefgründige Verständnis des Nutzers und seinen Bedürfnissen. Präziser ausgedrückt sollte man an dieser Stelle Empathie zum Nutzer aufbauen, sodass man seine Handlungen, Bedürfnisse, Gedanken, Motivationen und Verhaltensweisen versteht. Hierbei kann es hilfreich sein, sich an Ortschaften zu begeben, an

denen sich der Nutzer normalerweise aufhalten würde und sich angemessen in die Rolle hineinzuversetzen (Gehm, 2022). Der Einsatz von Interviews und Beobachtung der potenziellen Nutzer kann an dieser Stelle von großem Nutzen sein.

User Interviews als Recherchemethode ermöglichen ein besseres Verständnis von Nutzerbedürfnissen. Zu Beginn stellt der Interviewer sich und das Projekt vor, sodass eine erste Beziehung zum Nutzer aufgebaut werden kann. Anschließend wird eine sowohl einfache als auch offene Frage, die dazu dient, das Thema einzuleiten, zur Gesprächseröffnung gestellt. Anhand eines Interviewleitfadens kann man gezielte Fragestellungen nach und nach abarbeiten. Wichtig ist jedoch, sich nicht strikt an diesen zu halten, sondern sich auf das Gespräch einzulassen. Im Idealfall kommen an dieser Stelle emotionale Erlebnisse und Geschichten zum Thema, da besonders darin Potenzial steckt, wahre Hintergründe zu erfassen. Zum Schluss bedankt man sich beim Nutzer für seine Offenheit. (Gehm, 2022)

Durch das Beobachten potenzieller Nutzer kann man ebenso zahlreiche Eindrücke erzielen. Wichtig hierbei ist jedoch, dass der Nutzer sich nicht beobachtet fühlen soll, da dies gegebenenfalls sein Handeln verändern könnte (Gehm, 2022).

Holcomb und Segovia (2019) schrieben für die d.school in Stanford einen Artikel über Bill Pacheco, ein DT Bootcamp Alumni und Angestellter des „Cybex“-Unternehmens. Dieses Unternehmen stellt Trainingsgeräte im hochpreisigen Segment her. Hier wurde dieses Werkzeug eingesetzt, indem Pacheco regelmäßig Fitnessstudios besuchte, um Menschen zu beobachten. Sein Fokus lag auf ältere Sportler, die das Laufband verwendeten. Ihn überraschte, wie sie sich zur Unterstützung versuchten überall am Gerät festzuhalten, je schneller sie wurden. Bei Cybex mit seinen neuen Erkenntnissen und Mock-Ups angekommen, wurde er von seinem Vorgesetzten nicht ernst genommen. Doch er führte seine Bemühungen weiter, da er durch seine Recherche mit den Nutzern Empathie aufgebaut hatte und somit das Bedürfnis entwickelt hatte ihnen zu helfen. Somit modellierten er und sein Design-Team Laufbänder mit zusätzlichen Stützgriffen und stellten sie in einem kleinen Hotel in dessen Fitnessstudio auf. Ziel war es, wertvolle Daten für seine Vorgesetzten aufzuzeichnen. Zeitnah nach der Installation tendierten Nutzer zur Nutzung der neuen Geräte und auf Nachfrage, sagten sie, sie fühlen sich sicherer mit diesen Geräten zu trainieren. Nachdem Pacheco dies an sein

Management weitergab, bekam er die Möglichkeit mit den neuen Laufbändern an den Markt zu gehen. Sie erhöhten in den ersten zwei Jahren ihren Umsatz um 20%.

Diese Geschichte unterstreicht die Wichtigkeit dieser Phase. Denn durch die Leidenschaft, die man erst durch die in diesem Schritt generierten Empathie empfindet, erzielt man als Team das Bedürfnis dem Nutzer zu helfen.

2.3.3 Phase 3: Standpunkt definieren

Die letzten beiden Schritte befassen sich ausführlich mit der Fragestellung und dem Nutzer, sodass zu diesem Zeitpunkt zahlreiche Informationen und Beobachtungen vorliegen. In dieser Phase gilt es nun, die gesammelten Erkenntnisse zu verknüpfen, zu sortieren und auf Basis dessen einen Standpunkt zu definieren (Gehm, 2022). Ziel hiervon ist es eine bedeutungsvolle und umsetzbare Problemstellung zu erstellen (d.school, o. D.). Diese Phase ist eines der wichtigsten, da oftmals das Problem nicht vollständig verstanden worden ist, bevor man Lösungen generiert, welches wiederum in suboptimalen Lösungen mündet (unfccc, 2022). Eine klar definierte Problemstellung stellt Orientierung im Verlauf des Innovationsprozesses bereit und ermöglicht, einen breiten Lösungsraum für kreatives Denken zu schaffen (unfccc, 2022). Um dieses Ziel zu erreichen, können mehrere Tools zum Einsatz kommen.

Um die Informationskomplexität zu reduzieren, können sogenannte Personas eingesetzt werden (Gehm, 2022; unfccc, 2022). Sie sind fiktional und repräsentieren eine spezifische Nutzergruppe. Außerdem ermöglichen sie ein tiefes Verständnis der Nutzergedanken, -gefühle und -handlungen (unfccc, 2022), inspirieren das DT-Team und unterstützen es dabei, Entscheidungen zu treffen (Uebernicket et al., 2020).

Eine weitere Methodik sind *Customer Journey Maps*. Anhand dieser kann das Nutzerverhalten durch mehrere aufeinander folgende Berührungspunkte mit dem Unternehmen, Produkt oder einer Dienstleistung dargestellt werden (Gehm, 2022; Lewrick et al., 2019). Ein Vorteil dieser ist somit, dass sie nicht ausschließlich die direkten Berührungspunkte in Betracht ziehen, sondern auch die indirekten. Sie ermöglichen dem Team, mehr über ihre Zielgruppen zu lernen (Interaction Design Foundation, 2016).

Jede Persona besitzt individuelle Customer Journeys, da verschiedene Zielgruppen individuell mit Marken interagieren. Ein Beispiel für eine Customer Journey Map ist in Abbildung 5 zu sehen.

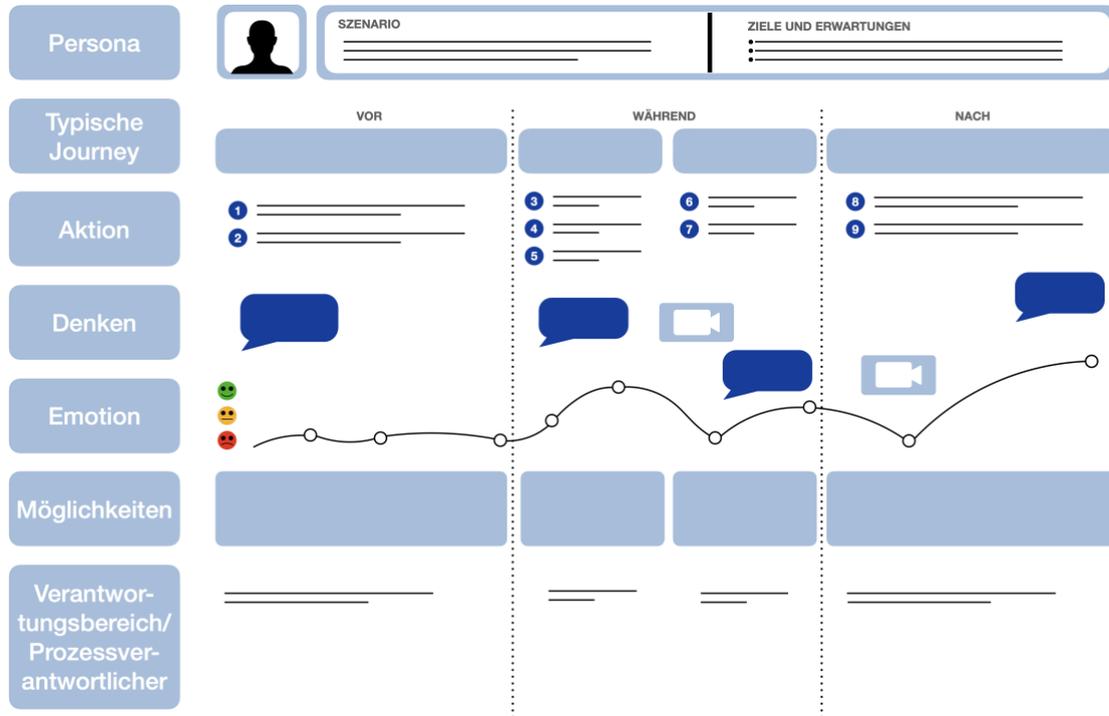


Abbildung 5: Template einer Customer Journey Map (eigene Darstellung) (Lewrick et. al., 2019)

Ein weiteres Tool, das oft in dieser Phase verwendet wird und ein essenzieller Bestandteil der DT-Toolbox ist, nennt sich „How might we...“-Frage. Mit dieser besonderen Ausdrucksweise wird das Team dabei unterstützt, in eine explorierende Denkweise zu kommen (Lewrick et. al., 2019). *How* schließt ein, dass es mehrere Lösungswege zu einer Frage gibt, wobei mit *might* die Sicherheit gegeben ist, dass eine mögliche Idee funktionieren könnte und das *we* als Erinnerung dafür gilt, dass man als Team an dem Problem arbeitet und dies gemeinsam löst (Lewrick et. al., 2019).

Diese Phase sorgt dafür, dass alle gesammelten Erkenntnisse auf das Wichtigste beschränkt worden sind. In ihr wird die Problemanalyse mit einem bestimmten Nutzerbild und einer

Fragestellung abgeschlossen, sodass sie den Beginn der Lösungsfindung abbildet (Gehm, 2022).

2.3.4 Phase 4: Ideen generieren

Die Ideengenerierungsphase ermöglicht anhand verschiedener Kreativitätstechniken viele Ideen zu entwickeln, zu clustern sowie zu priorisieren und anhand dessen Lösungen für das definierte Problem zu finden (Gehm, 2022).

Ein mögliches und oft angewendetes Werkzeug ist *Brainstorming*. Hierbei geht es in erster Reihe darum, zahlreiche Ideen zu generieren, sodass hierbei die Quantität erstmal wichtiger als die Qualität ist (Gehm, 2022). Es gibt es einige Grundregeln, die an dieser Stelle eingehalten werden müssen, um den kreativen Prozess zu stimulieren (Gehm, 2022; Schallmo & Lang, 2020):

- **Visualisieren:** Zum Inspirieren anderer sollten Ideen prägnant und verständlich auf bspw. einem Post-It dargestellt werden.
- **One Speaker:** Beim Vortragen von Ideen spricht nur eine Person und hält sich kurz sowie präzise.
- **Baue auf den Ideen anderer auf:** Als Inspirationsquelle können Ideen andere Teammitglieder gezielt verwendet werden.
- **Quantität vor Qualität**
- **Stelle Kritik zurück:** In dieser Phase liegt der Fokus auf Weiterentwicklung, daher sollten die Ideen anderer weder bewertet noch diskutiert werden.
- **Go for wild ideas:** Zu verrückte Ideen gibt es nicht, im Gegenteil, sie tragen oft durch den Neuigkeitswert am meisten zum Finden von Lösungen und Inspirationen bei.

Es ist wichtig, dass typische Denkmuster durch das Verändern der Rahmenbedingungen durchbrochen werden können. An dieser Stelle können Fragen nach Lösungsmöglichkeiten bei unbegrenztem Geld limitierende Faktoren entfernen, sodass neue Ideen generiert werden können (Gehm, 2022).

Nachdem viele Ideen vom Team vorgestellt worden sind, sollten diese nach festgelegten Kriterien gruppiert und strukturiert werden, um dem Team einen guten Überblick zu verschaffen (Gehm, 2022).

Hierbei kann eine 2x2-Matrix wie in Abbildung 6 illustriert hilfreich sein, da diese einen guten Überblick darüber schafft, welche Ideen weiterverfolgt werden sollten, und welche verworfen werden sollten (Lewrick et. al., 2019). Ideen, die wenig Mühe aufwenden und einen hohen Wert aufweisen, werden in den entsprechenden Balken gesetzt; hier grün. Genauso führt man das mit den weiteren Ideen fort; viel Mühe und geringer Wert in rot und so weiter.

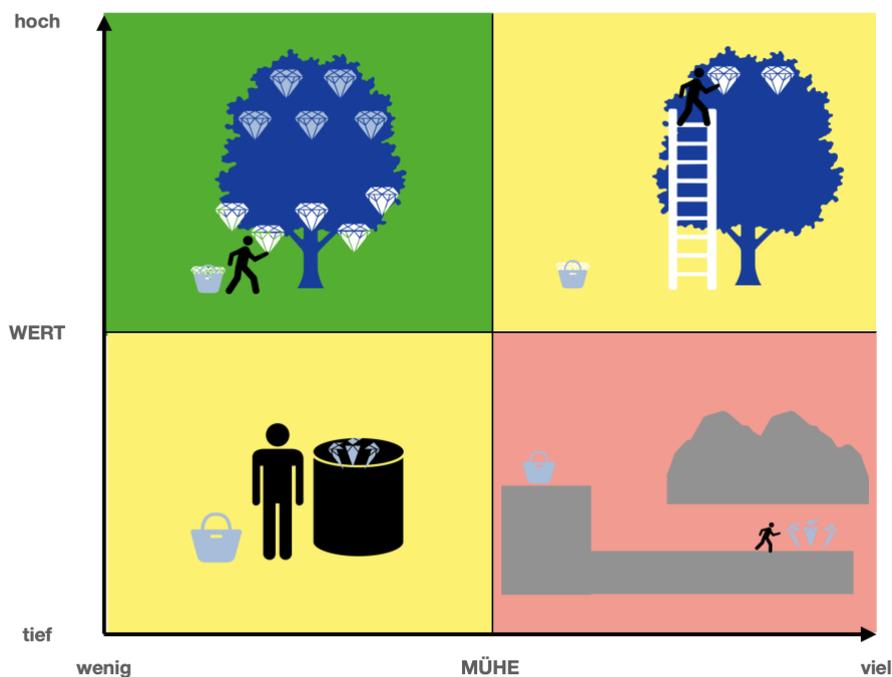


Abbildung 6: 2x2 Matrix zum Clustern von Ideen (eigene Darstellung) (Lewrick et. al., 2019)

Sobald das Team eine Priorisierung der Ideen durchgeführt hat, können diese weiter detailliert werden (Gehm, 2022). Um zu vermeiden zu stark ins Detail zu gehen, empfiehlt sich hier ein Template zu verwenden. Zusätzlich unterstützt hierbei, die Ideen visuell zu präsentieren und Fragen gerichtet auf die Zielgruppe und Nutzerbedürfnisse zu beantworten (Gehm, 2022). Dies fördert ein gemeinsames Verständnis und stellt eine gute Grundlage für die anschließende Phase dar.

2.3.5 Phase 5: Prototypen entwickeln

Die zuvor generierten Ideen werden in der Prototypenbau-Phase in greifbare, erlebbare und sichtbare Lösungen realisiert. Ziel hiervon ist, die auserwählten Ideen für potenzielle Nutzer mit möglichst wenig Aufwand bereitzustellen, sodass schnellstmöglich wertvolles Kunden-Feedback generiert werden kann (Gehm, 2022). Damit können zeitnah Optimierungspotenziale aufgedeckt werden. Man orientiert sich an den Ansatz des *Minimum Viable Product* (MVP), welches nur die wahrlich erforderlichen Informationen abbildet, die zum Testen des Produkts benötigt werden (Gehm, 2022). Somit werden anhand des Weglassens unnötiger Features Zeit, Kosten und Entwicklungsressourcen gespart (Gehm, 2022).

Es gibt eine Vielzahl an möglichen Prototyp-Arten (Gehm, 2022):

- a) Eine digitale oder analoge Skizze kann das Produkt, die Dienstleistung oder den Prozess in einer einfachen Form illustrieren.
- b) Rollenspiele ermöglichen Teammitgliedern gut Erlebnisse des Kunden mit dem jeweiligen Produkt oder der jeweiligen Dienstleistung dazustellen.
- c) Genauso ist es sinnvoll ein physisches 3D-Modell der Idee darzustellen mit einfachen Materialien wie Karton, Knete oder Legos.

Die unterschiedlichen Prototyp-Arten können selbstverständlich auch kombiniert werden, um eine bessere Interaktion zu ermöglichen. Aufbauend auf dem daraus generierten Nutzerfeedback kann iterativ die Anfertigung sowie Weiterentwicklung der Prototypen angestoßen werden (Gehm, 2022).

2.3.6 Phase 6: Testen

Ein essenzieller Aspekt des DTs ist das Testen der aus der letzten Phase generierten Prototypen an potenziellen Nutzern. Somit wird nicht ausschließlich Feedback zu den jeweiligen Prototypen generiert, sondern auch die Sichtweise des Teams auf das Problem und den Nutzerbedürfnissen geschärft (Lewick et. al., 2019). Durch die neuen Erkenntnisse können Ideen sowohl bewertet als auch verworfen oder weiterentwickelt werden (Gehm, 2022). Außerdem wird somit eine Rückkopplung zu den ersten beiden Phasen (Verstehen und Beobachten) generiert, welches wiederum relevant für den dritten Schritt (Standpunkt

definieren) wäre (Lewrick et. al., 2019). Aufgrund dessen spricht man beim DT von einem iterativen Vorgehen, denn diese Mikrozyklen können beliebig oft wiederholt werden, bis das Team eine endgültige Lösung bereitstellen kann.

Laut Gehm (2022) sollte man in dieser Phase folgende Bestandteile beachten:

- a) **Planung des Tests:** Zum Anfang sollten verständliche Lernziele definiert werden, da für den Erfolg eines Prototypentests eine gute Vorbereitung essenziell ist.
- b) **Rollenverteilung:** Vor Testbeginn werden unterschiedliche Rollen definiert:
 - a. **Moderator:**
 - unterstützt die Testperson dabei, sich in die Prototyp-Situation zu versetzen
 - erklärt den notwendigen Kontext
 - stellt Fragen, um mehr Erkenntnisse über die Erfahrung bei der Nutzung zu sammeln
 - b. **Schauspieler:**
 - helfen dabei, die Situation realistischer zu gestalten
 - c. **Bobachter:**
 - dokumentieren alles, was der Nutzer tut
- c) **Initialer Testlauf:** Ein initialer Testverlauf mit einer Testperson sollte durchlaufen werden, um Fehler beim Testszenario mit dem Nutzer auszuschließen.
- d) **Dokumentieren:** Die Testresultate sollten strukturiert dokumentiert werden. Hier sollten Fragen sowie Ideen festgehalten werden, was gut lief und wo Probleme während der Nutzung aufgetaucht sind, sollte ebenfalls durchgeführt werden.

Da die ersten Prototypen improvisiert und simpel aussehen, scheut sich keiner davor seine ehrliche Meinung sowie Kritik an die mit wenig Aufwand verrichtete Arbeit zu äußern. Außerdem fällt es dem DT-Team leichter die Ideen wieder zu verwerfen und von vorne anzufangen (Gehm, 2022).

Abschließend wird durch diese Phasen des DTs sichergestellt, dass auf iterativer Art und Weise ein nutzerorientiertes Produkt oder Dienstleistung erzeugt werden kann.

Nachdem dieses Kapitel sich mit den Grundlagen und Konzepten von DT ausführlich befasst hat, widmet sich das folgende Kapitel einem anderen wichtigen Framework für die Produktentwicklung: Scrum.

3 Scrum

Scrum ist eine agile Projektmanagement-Methode, die in den vergangenen Jahren erheblich an Bedeutung gewonnen (AlleAktien GmbH, 2024). Es fokussiert sich auf die agile Projektsteuerung und die kontinuierliche Wertschöpfung (Lenz, 2021). Die Anwendung dessen findet gewöhnlich im Software-Entwicklungsbereich statt, inzwischen wird es jedoch auch in anderen Branchen eingesetzt (AlleAktien GmbH, 2024).

In diesem Kapitel werden wird der Begriff sowie seine Herkunft erläutert. Daraufhin folgen die Theorie und Werte der agilen Methodik. Anschließend werden die Kernprinzipien sowie Bestandteile wie Rollen, Ereignisse und Artefakte von Scrum genauer beleuchtet.

3.1 Begrifflichkeit

Wenn man den Begriff *Scrum* zum ersten Mal hört und nicht zufällig ein Fan von Rugby ist, kann man sich mit großer Wahrscheinlichkeit nichts Konkretes darunter vorstellen. Das Wort lässt sich ins Deutsche zu *Gedränge* übersetzen. Diese spezielle Rugby-Spielsituation umfasst 2 Teams, dessen Spieler eng zusammenstehen und als kleine, selbstorganisierte Einheiten arbeiten. Von außen wird lediglich eine Richtung verordnet und dem Team ist überlassen, wie sie ihr Ziel erreichen (Simschek & Kaiser, 2019b). Dieser sogenannte *Rugby-Approach* lässt sich auf zwei japanische Wirtschaftswissenschaftler - Ikujiro Nonaka und Hirotaka Takeuchi - zurückführen. Diesen thematisieren sie in ihrem im Jahr 1986 veröffentlichten Artikel „The New New Product Development Game“. Darin sind sie der Ansicht, dass einer der wichtigsten Faktoren für sehr erfolgreiche Produktentwicklungsteams die räumliche Nähe während der Entwicklungsarbeit ist (Simschek & Kaiser, 2019b).

Im Jahr 1995 präsentierten Jeff Sutherland und Ken Schwaber aus ihrer Arbeit „The Scrum Development Process“ bei der OOPSLA-Konferenz (Object-Oriented Programming, Systems, Languages & Applications) sowohl das Scrum-Konzept als auch dessen Anwendungsmöglichkeit auf die Softwareentwicklung (Satpathy, 2016). Beide hatten Scrum einige Jahre zuvor schon in Projekten eingesetzt und stetig optimiert (Kaiser & Van Bennekum, 2022).

„Scrum ist ein leichtgewichtiges Rahmenwerk, welches Menschen, Teams und Organisationen hilft, Wert durch adaptive Lösungen für komplexe Probleme zu generieren.“ (Schwaber & Sutherland, 2020, S. 3)

3.2 Theorie

Scrum beruht einerseits auf *Empirie*, welche besagt, dass Wissen aus Erfahrung erworben wird und Beobachtungen als Basis genommen werden, um Entscheidungen zu treffen. Andererseits basiert es auf *Lean Thinking*, welches den Fokus auf das Wichtige setzt und Verschwendung verringert (Schwaber & Sutherland, 2020).

Der iterative und inkrementelle Ansatz von Scrum bietet die Gelegenheit zur Inspektion und Adaption in stetigen, kurzen Abständen (Simschek & Kaiser, 2019b). Es beruht auf der Zusammenarbeit von Teams, dessen Mitglieder über eine Vielzahl von Fähigkeiten und Fachkenntnissen verfügen, um die Arbeit zu bewältigen.

Das Scrum-Framework integriert vier formale Ereignisse (s. Abs. 3.5.2) in einem Sprint, wobei diese daraus ausgelegt sind, die grundlegenden Prinzipien von Scrum zu unterstützen (s. Abb. 7). Diese werden nachfolgend nach Schwaber und Sutherland (2020) erläutert.

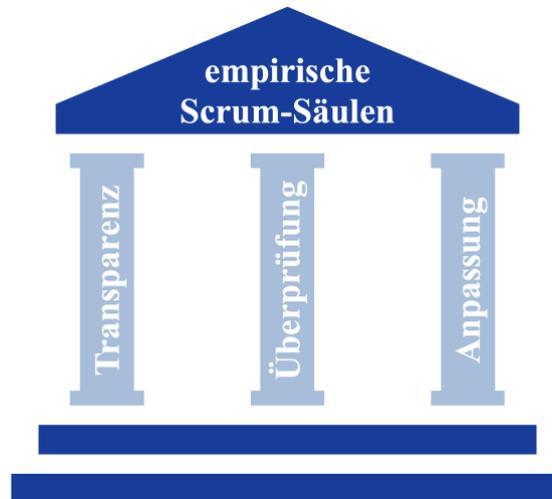


Abbildung 7: Die empirischen Scrum-Säulen
(eigene Darstellung)

Transparenz

Sowohl der fortlaufende Prozess als auch die sich entwickelnde Arbeit müssen für diejenigen, die sie durchführen, und diejenigen, die sie empfangen, deutlich sichtbar sein. In Scrum bildet der aktuelle Stand seiner drei formalen Artefakte (s. Abs. 3.5.3) die Grundlage für wichtige Entscheidungen, sodass wenig transparente Artefakte zu Entscheidungen führen könnten, die den Wert verringern sowie das Risiko steigern.

Überprüfung

Auf Basis dieser Transparenz müssen die Artefakte und der Fortschritt der gesetzten Ziele regelmäßig und gründlich überprüft werden, sodass mögliche ungünstige Abschweifungen sowie Probleme frühzeitig aufgedeckt werden können. Um diese Überprüfung zu gewährleisten, stellt Scrum seine fünf Events bereit.

Anpassung

Falls eine Diskrepanz einzelner Aspekte des Prozesses vorliegen sollte oder das Produktergebnis abweicht, muss der eingesetzte Prozess oder die hergestellten Ergebnisse entsprechend verbessert und angepasst werden. Um weitere Abweichungen zu vermeiden, muss die Anpassung zeitnah vorgenommen werden. Außerdem muss sich das Scrum Team selbständig verwalten dürfen, sonst wird die Anpassung schwieriger, denn es wird vom Scrum

Team erwartet, dass es sich direkt anpasst, sobald es durch die Überprüfung etwas Neues gelernt hat, ohne auf die Genehmigung Dritter zu warten.

3.3 Werte



Abbildung 8: Scrum-Werte (eigene Darstellung) (Schwaber & Sutherland, 2020)

Scrum kann erfolgreich angewendet werden, wenn die in Abbildung 8 dargestellten Werte durch das Scrum Team und der Beteiligten des Scrum-Prozesses verkörpert werden. Daraus werden auch die im letzten Abschnitt thematisierten Scrum-Säulen erst lebendig und stellen Vertrauen her.

Das Scrum Team **verpflichtet** sich, die vorgenommenen Ziele zu erreichen sowie sich gegenseitig zu unterstützen. Sein hauptsächlicher **Fokus** liegt auf der zu verrichtenden Arbeit des Sprints, sodass der ideale Fortschritt zum Ziel geschaffen werden kann. Genauso muss auf beiden Seiten des Scrum Teams und dessen Stakeholder eine gewisse **Offenheit** bezüglich der Arbeit und der Herausforderungen sein. Die Mitglieder des Scrum Teams sollten sich nicht davor scheuen und **mutig** sein, das Richtige zu tun. Dies umfasst an herausfordernden Problemen zu arbeiten. Selbstverständlich sollten sich dessen Beteiligten gegenseitig als qualifizierte, unabhängige Personen **respektieren** und bekommen ebenfalls diesen Respekt von Menschen, mit denen sie zusammenarbeiten verliehen.

Das Scrum Team kann sich an diesen Werten bezüglich seiner Arbeit, seiner Aktionen sowie seiner Verhaltensweise orientieren. Die Werte bilden das Fundament einer erfolgreichen

Anwendung des Scrum-Frameworks. Genauso bilden sie die Grundlage, für die im folgenden Abschnitt thematisierten 12 agilen Prinzipien.

3.4 Prinzipien

Scrum ist ein agiles Vorgehensmodell und basiert auf den Grundsätzen des agilen Manifests (s. Abb. 9) (Kuster et al., 2011).



Abbildung 9: Das agile Manifest (eigene Darstellung) (Kleuker, 2018)

Aus diesem agilen Manifest lassen sich agilen Prinzipien (s. Tabelle 3) herleiten. Aus diesen Grundgedanken sowie Prinzipien wird erkenntlich, dass die Kommunikation des

Entwicklerteams untereinander sowie mit dem Kunden im Mittelpunkt steht (Wieczorrek & Mertens, 2010).

Tabelle 3: Agile Prinzipien (Häger et al., 2013; Wieczorrek & Mertens, 2010)

(1) Höchste Priorität hat die Zufriedenstellung des Kunden durch frühe & kontinuierliche Lieferung brauchbarer Software
(2) Anforderungen sind auch in fortgeschrittenen Entwicklungsstadien möglich
(3) Die Software wird inkrementell & in kurzen Iterationen erstellt
(4) Fachexperten & Entwickler arbeiten möglichst direkt & täglich zusammen.
(5) Die effizienteste & effektivste Art, Informationen zu verbreiten, ist die direkte Kommunikation von Angesicht zu Angesicht.
(6) Funktionierende Software ist die primäre & wichtigste Kenngröße für den Projektfortschritt.
(7) Konzentration auf das Wesentliche heißt explizit & regelmäßig zu entscheiden, was wegzulassen ist.
(8) Das Entwicklungsteam reflektiert in regelmäßigen Abständen darüber, wie es die gemeinsame Arbeit verbessern kann.
(9) Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen, und vertraue darauf, dass sie die Aufgaben erledigen.
(10) Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten kontinuierlich ein konstantes Entwicklungstempo beibehalten können.
(11) Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
(12) Die besten Architekturen, Anforderungen und Designs entstehen durch selbstorganisierte Teams.

Diese zwölf agilen Prinzipien bieten Leitlinien für die agile Produktentwicklung und dienen als Grundlage für die Umsetzung von Scrum. Die Umsetzung dieser Prinzipien erfolgt in Scrum durch verschiedene Bestandteile des Frameworks, die dafür entwickelt wurden, die agilen Werte und Prinzipien in die Praxis umzusetzen. Dazu gehören die Rollen, Ereignisse und Artefakte, die im kommenden Abschnitt vorgestellt werden.

3.5 Bestandteile von Scrum

Weiterhin wird Scrum in drei Rollen, fünf Ereignissen sowie drei Artefakte aufgeteilt (Schwaber & Sutherland, 2020; Simschek & Kaiser, 2019b), die im Folgenden nach Simschek und Kaiser (2019b) weiter erläutert werden.

3.5.1 Rollen

Die Aufgaben von jedem Teammitglied werden von den nach Scrum definierten Rollen geregelt, sodass jede Rolle bestimmte Aufgaben, Rechte sowie Pflichten besitzt. Im Folgenden werden die spezifischen Rollen erläutert.

Entwicklungsteam/Developer

Das Entwicklungsteams repräsentiert das Herz von Scrum, da es sich sowohl um die Implementierung des Produktes kümmert als auch dieses entwickelt. Es muss sich selbst organisieren, um die gesetzten Sprint-Ziele zu erreichen.

Product Owner

Der Product Owner setzt sich für die Interessen des Kunden (der Auftraggeber) ein. Er trägt die Verantwortung für die Planung sowie Lenkung der Produktentwicklung (Gloger, 2016) und somit auch für den geschäftlichen Erfolg des Produkts .

Scrum Master

Der Scrum Master ist für die Implementierung der Scrum-Regeln verantwortlich und vertritt die Rolle des Moderators und Coachs.

3.5.2 Ereignisse

Anhand der Ereignisse, auch als Events bezeichnet, werden Form, Frequenz und Inhalt der Kommunikation zwischen den Rollen sowie Mitgliedern des Teams geregelt.

Sprint

Scrum wird durch sein iteratives sowie zyklisches Vorgehen ausgezeichnet. In einem Sprint werden einzelne Elemente des Produktes entwickelt, welche als Inkrement bezeichnet werden. Dementsprechend kann man einen Sprint als eine Iteration des Scrum-Prozesses betrachten. Innerhalb dessen arbeitet das Entwicklungsteam gewisse zuvor definierte Produkteigenschaften ab und setzt diese um. Die Erreichung der vom Development-Team gesetzten Ziele, wird in Sprints angestrebt. Innerhalb eines Sprints finden die folgenden Events statt: Sprint Planning, Daily Scrum, die Entwicklungsarbeit, das Sprint Review sowie die Sprint Retrospektive. Ein Sprint kann von einigen Tagen bis maximal 4 Wochen andauern und sollte immer dieselbe Dauer betragen. Grund dafür ist, dass das Entwicklungsteam eine höhere Leistungsfähigkeit nachweisen kann. Es existiert keine feste Anzahl an Sprints, denn diese werden so lange durchgeführt, wie das Entwicklungsteam an dem Produkt oder der Dienstleistung arbeitet und ist somit immer projektgebunden. Nichtsdestotrotz besteht die Möglichkeit, dass der Product Owner einen Sprint abbricht, wobei als einzige Bedingung hierfür ist, dass das Sprintziel irrelevant geworden ist.

Sprint Planning

Im Sprint Planning wird angestrebt, den bevorstehenden Sprint zu planen, sodass jeder Sprint mit diesem Event beginnt und dieser somit nur einmal pro Sprint stattfindet. Es dauert bei einem vier-wöchigen Sprint maximal acht Stunden und passt sich somit der Sprintdauer an.

Es wird in zwei Teile untergliedert; der erste Teil umfasst die Planung des anstehenden Sprints und thematisiert die Umsetzung der Produkteigenschaften dessen, wobei der zweite Teil die Umsetzung der anstehenden Arbeit bespricht.

Daily Scrum

Nach Abschluss des Sprint Plannings, setzt sich das Team an die Arbeit und bearbeitet nacheinander die definierten Aufgaben ab. Das Team trifft sich während dieser Entwicklungsarbeiten pro Werktag einmal zu einem Daily Scrum Meeting, das immer zum

selben Zeitpunkt stattfindet. Das Meeting dauert maximal 15 Minuten und darin stimmt sich das Entwicklungsteam ab. In jedem Daily Scrum werden zunächst transparent die Entwicklungsarbeiten der letzten 24 Stunden offenbart und die der anstehenden 24 Stunden geplant. Parallel wird der Fortschritt des Sprint-Ziels dokumentiert sowie der Fortschritt der Backlog Items analysiert. Anhand des Daily Scrums wird die Wahrscheinlichkeit der Erreichung des Sprint-Ziels erhöht.

Sprint Review

Am Ende der Entwicklungsarbeiten findet der Sprint Review statt und dient zur Präsentation, Überprüfung sowie Anpassung der wichtigsten Sprintereignissen. Dies ermöglicht eine Transparenz des neuesten Stands des Produktinkrements, sodass das Product Backlog aktualisiert werden kann. Die Dauer des Sprint Reviews beträgt bei einem vierwöchigen Sprint maximal vier Stunden.

Sprint Retrospektive

Die Sprint-Retrospektive findet am Ende jedes Sprints statt und dient dazu Feedback zur organisatorischen sowie strukturellen Verbesserung des Entwicklungsprozesses einzuholen. Somit liegt der Fokus nicht auf die erarbeitete Leistung, sondern auf die Arbeitsweise. Somit soll das Verbesserungspotenzial in Bezug auf Menschen, Interaktion, Prozess und Werkzeuge aufgedeckt werden. Die Sprint Retrospektive dauert maximal drei Stunden, wenn an dem Sprint vier Wochen lang gearbeitet wird.

3.5.3 Artefakte

Als Artefakte werden bestimmte Tools und Techniken bezeichnet, die essenziell zur erfolgreichen Umsetzung von Scrum sind sowie notwendig zur effizienten Gestaltung des Projektablaufs sind.

Product Backlog

Im Product Backlog befindet sich eine Sammlung aller Produktfeatures, die das Produkt nach Abschluss der Entwicklungsarbeiten besitzen sollte. Die einzelnen Features, die sich in diesem Backlog befinden, werden auch als Product Backlog Items bezeichnet und werden nach Priorität geordnet. Es bleibt bestehen, solange das Produkt vorhanden ist. Das Product Backlog

ist nie vollständig, was unter anderem daran liegt, dass es im Laufe des Projekts und der Produktentwicklung kontinuierlich überprüft sowie angepasst werden kann. Somit besitzt das Product Backlog dynamische Eigenschaften und verändert sich je nach Produkthanforderungen.

Sprint Backlog

Das Sprint Backlog dient zur Veranschaulichung der Backlog Items, die während des Sprint Plannings aus dem Product Backlog entnommen worden sind., um in dem jeweiligen Sprint vom Entwicklungsteam umgesetzt zu werden. Genauso stellt es eine Übersicht des aktuellen Standes dar, sodass transparent wird, welche Aufgaben schon erledigt worden sind und welche noch zur Erreichung des Sprint-Ziels anstehen.

Inkrement

Das Produkt in seinem aktuellsten Auslieferungszustand (inklusive das, was im aktuellen Sprint umgesetzt wurde) trägt die Bezeichnung Inkrement. Es beinhaltet alle Backlog Items, die in den vergangenen Sprints umgesetzt worden sind.

3.6 Der Scrum-Prozess

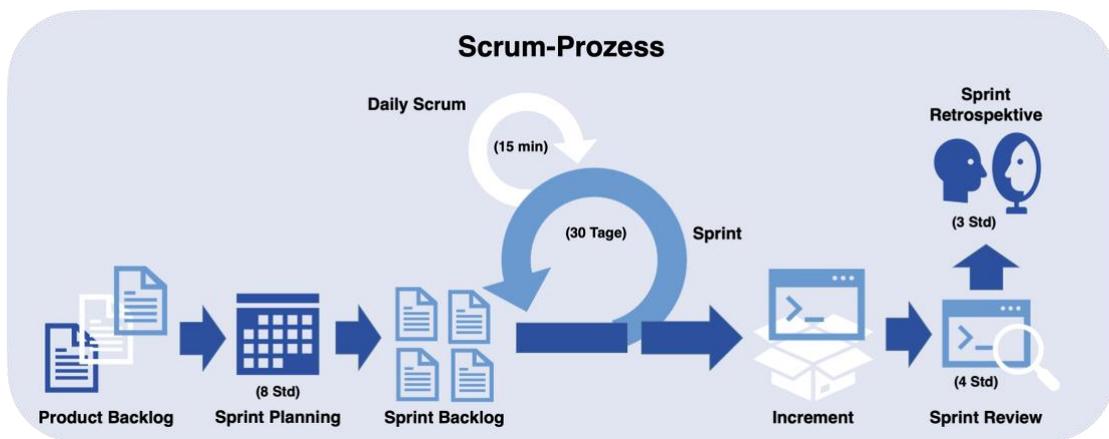


Abbildung 10: Der Scrum-Prozess (eigene Darstellung) (Schwaber & Sutherland, 2020)

Der Scrum-Prozess startet erst, wenn ein Kunde ein Produkt benötigt. Die Produkthanforderungen werden im Product Backlog festgehalten und gesammelt. Sobald dies vollständig ist, fängt man mit dem Sprint Planning an, welches die Planung zur Umsetzung der Produktfeatures im anstehenden Sprint umfasst. Im Anschluss wird diese Teilmenge in das

Sprint Backlog überführt. Nun kann der Sprint beginnen. Im Laufe des Sprints erfolgt im Rahmen des Daily Scrums ein täglicher Austausch des Scrum Teams. Sobald der Sprint abgeschlossen ist, sollten als Resultat neue Produktfeatures für das Produktinkrement präsentiert werden. Dem anschließend besteht die Möglichkeit im Sprint Review Anpassungen und Überprüfungen vorzunehmen. Es ermöglicht dem restlichen Team, das an den Entwicklungsarbeiten nicht teilgenommen hatte, offene Fragen zu klären oder sich Informationen über den aktuellen Entwicklungsstand einzuholen.

Nachdem die wesentlichen Bestandteile und Prinzipien von Scrum betrachtet wurden, stellt sich nun die Frage, wie sich dieses agile Framework mit anderen Methoden sowie Ansätzen integrieren lässt. Eine vielversprechende Möglichkeit besteht darin, DT mit Scrum zu kombinieren, um die Vorteile beider Ansätze zu nutzen.

4 Integration von Design Thinking und Scrum

In den vorangegangenen Kapiteln befasste sich diese Arbeit intensiv mit den Grundlagen des DTs. Genauso wurde Scrum zusammenfassend aufgegriffen. Während beide Ansätze vereinzelt bereits einen erheblichen Mehrwert bieten, wird in der Praxis zunehmend anerkannt, dass die Kombination die Potenziale beider optimal nutzbar sind (Hehn et al., 2022).

In den folgenden Abschnitten befasst sich dieses Kapitel daher mit der Frage, wie DT und Scrum erfolgreich integriert werden können. Dazu werden zunächst die Synergien zwischen beiden Ansätzen beleuchtet, gefolgt von der Vorstellung von zwei konzipierten Integrationsansätzen: dem Collective Framework DTScrum von Hehn et al. (2022) und dem DT@Scrum-Ansatz von Häger et al. (2014). Anschließend werden beide Ansätze verglichen und davon Handlungsempfehlungen abgeleitet, um Unternehmen dabei zu unterstützen, das für ihre spezifischen Anforderungen am besten geeignete Integrationskonzept auszuwählen. Im Anschluss werden Hindernisse, auf die man beim Integrationsversuch stößt, thematisiert.

4.1 Synergien zwischen Design Thinking und Scrum

In diesem Abschnitt werden die zahlreichen Synergien vorgestellt, die sich durch die Integration von DT und Scrum anbieten.

Iteratives Vorgehen

Sowohl DT als auch Scrum sind iterative Ansätze. Der iterative DT-Prozess betont das kontinuierliche Testen und Überarbeiten von Ideen. Genauso basiert Scrum auf einem iterativen Ansatz, bei dem Arbeit in kurzen Sprints durchgeführt wird, gefolgt von Feedback und Anpassungen.

Nutzerzentrierung

Im Kapitel Design Thinking wurde die Bedeutung des Einbezugs des Menschen betont, sodass hervorgehoben wurde, dass Nutzerzentrierung eine große Rolle spielt. DT zielt dementsprechend auf die Entwicklung eines tiefen Verständnisses für die Bedürfnisse, Wünsche sowie Probleme der Nutzer ab. Dadurch, dass Scrum Teams ermöglicht schnell und iterativ zu arbeiten, wird auch hier eine enge Interaktion mit den Nutzern im Verlauf des gesamten Entwicklungsprozesses ermöglicht und somit die Nutzerzentrierung gewährt.

Experimentation und Prototyping

DT ermutigt frühzeitig Prototypen zu erstellen und anhand dessen Ideen schnell zu testen, um zeitnah wertvolles Feedback zu erhalten. Genauso dient die Förderung einer regelmäßigen Lieferung des Inkrements, die es den Nutzern ermöglicht das Produkt frühzeitig zu testen sowie Feedback zu geben.

Kollaboration und Multidisziplinarität

Sowohl DT als auch Scrum fördern die Zusammenarbeit des Teams. Eines der Kernelemente von DT ist das Bilden von interdisziplinären Teams, sodass möglichst viele Perspektiven sowie Fähigkeiten eingebunden werden können. Scrum basiert auf die enge Zusammenarbeit innerhalb eines Teams und betont die Bedeutung von Transparenz, Kommunikation sowie gemeinsamer Verantwortung.

Flexibilität und Anpassungsfähigkeit

Aufgrund des allgemein gehaltenen Prozesses ist DT flexibel und passt sich somit den Anforderungen sowie Herausforderungen des jeweiligen Produktes an. Genauso bietet Scrum eine flexible Struktur, die es dem Team ermöglicht, Entscheidungen basierend auf sich ändernde Anforderungen anzupassen und kontinuierlich zu verbessern.

Ergebnisfokussierung

Ziel von DT ist es, innovative Lösungen zu entwickeln, die einen wahren Mehrwert für die Nutzer bieten. Ähnlich liegt der Fokus auf die Bereitstellung von Ergebnissen und Mehrwert für den Nutzer bei Scrum, indem es die Lieferung von funktionsfähiger Software in kurzen Iterationen fördert.

4.2 Konzepte und Ansätze zur Integration von Design Thinking in das Scrum-Framework

Die Integration von DT in das Scrum-Framework wurde bereits mehrmals untersucht. Hehn et al. (2022) stellen ein kollektives Framework vor, in dem Scrum und DT wie gewohnt allein arbeiten, jedoch auf mehreren Weisen im Laufe des Prozesses verbunden sind, sodass von den Stärken beider profitiert werden kann. Dieses Konzept wird in Abschnitt 4.2.1 detaillierter beschrieben. Häger et al. (2014) betonen die Notwendigkeit einer ordnungsgemäßen Integration von DT-Aktivitäten in Softwareentwicklungsprozessen, um die Umsetzung innovativer Ideen sicherzustellen. Dieses Integrationsmodell wird in Abschnitt 4.2.2 vorgestellt.

Diese beiden Modelle werden dann im Anschluss miteinander verglichen, sodass im nächsten Kapitel eine adäquate Handlungsempfehlung ausgesprochen werden kann.

4.2.1 Collective Framework DTScrum

Dieses Konzept nach Hehn et al. (2022) zielt darauf ab, die Vorteile von DT und Scrum in der Softwareentwicklung zu vereinen, um menschenzentrierte Ansätze zur Lösung komplexer Probleme zu nutzen. Es basiert auf der Idee des kollektiven Arbeitens von Malone et al. (2009) und betont sowohl die Zusammenarbeit als auch den Austausch zwischen beiden Ansätzen.

Das Framework umfasst verschiedene Komponenten, darunter Artefakte, Rollen und Verantwortlichkeiten sowie Aktivitäten. Zusätzlich fördert es Feedback-Zyklen und Koordinationsmeeting, um eine kontinuierliche Verbindung zwischen DT und Scrum zu gewährleisten.

Der Begriff *kollektiv* beschreibt sowohl die multidisziplinäre Natur des Frameworks als auch das parallele verlaufen beider Prozesse (s. Abb. 11). Sie ergänzen sich somit im gesamten Lösungsfindungs- und -entwicklungsprozess auf verschiedene Weise, sodass von den jeweiligen Stärken des jeweils anderen profitiert werden kann. Dieses Konzept umfasst folgende Prozessbereiche, auf die im Anschluss gründlicher eingegangen wird:

Multidisciplinary Knowledge Café, Design Thinking Process Area, Scrum Process Area sowie Product Backlog Design Matching.

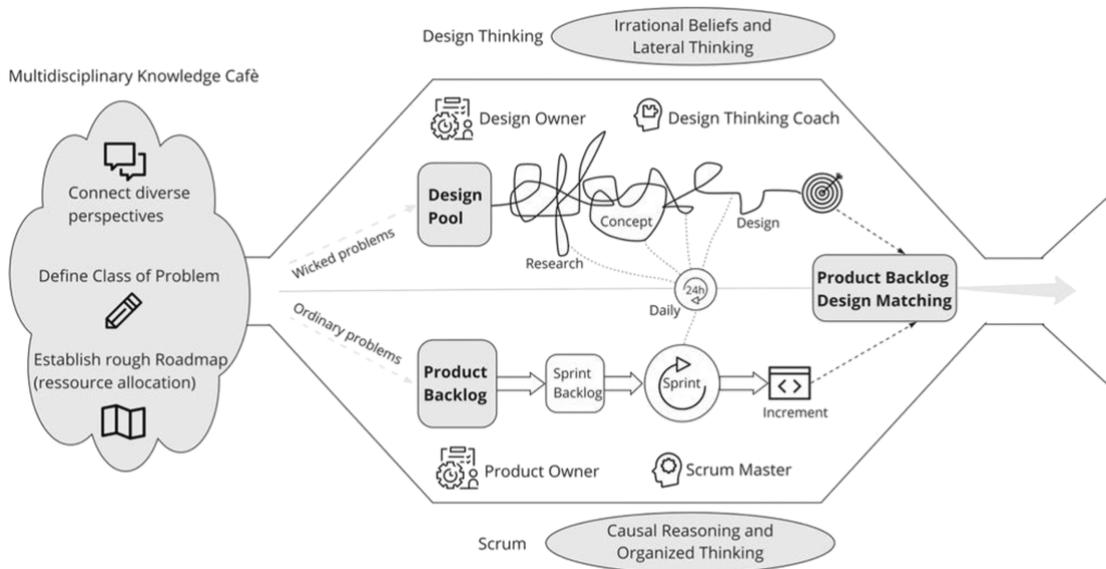


Abbildung 11: Collective process framework (Hehn et al., 2022)

Multidisciplinary Knowledge Café

Der erste Prozessbereich bildet das Grundgerüst eines erfolgreichen Ansatzes zur Designimplementierung und umfasst verschiedene Aspekte, darunter das Zusammenbringen von Menschen mit unterschiedlichen Hintergründen, das Verbinden verschiedener Standpunkte, das vorzeitige Erstellen einer groben Roadmap zur angemessenen Planung und Ressourcenzuweisung sowie das Identifizieren und Klassifizieren von Problemen. *Multidisziplinär* bezieht sich hier auf das DT-Mindset, das auf die Zusammenführung verschiedener Standpunkte abzielt.

DT ermöglicht das Lösen komplexer Probleme und die Generierung neuer Ideen. Eine kreative und offenere Herangehensweise führt entweder zur Identifizierung konkreter Probleme oder zu der Erkenntnis, dass kein echtes Problem besteht, sodass Zeit sowie Ressourcen eingespart werden können. Somit zielt das Multidisciplinary Knowledge Café darauf ab, Subprobleme sowohl zu identifizieren als auch zu klassifizieren. Bei diesen wird zwischen *gewöhnlichen* und *wicked* Problemen unterschiede wird. Während sich ersteres gut mit Scrum definieren lässt,

behandelt DT eher komplexe Probleme und Innovationen. Dementsprechend ist eine Klassifizierung der Problemart von wichtiger Bedeutung, um einen effektiven Umgang mit ihnen zu gewährleisten.

Das Collective Process Framework integriert nicht nur Scrum und DT, sondern strebt eine parallele Planung und Zusammenarbeit der beiden an. Dementsprechend sind eine kollaboratives Arbeits- sowie Denkweise entscheidend für eine erfolgreiche Implementierung eines Produktdesigns. Dies wird im Konzeptmodell durch die folgenden zwei Prozessbereiche dargestellt.

Scrum Process Area

Der zweite Prozessbereich beginnt mit dem Backlog-Management, in welchem die Probleme strukturiert, kategorisiert und priorisiert werden. Hier ist es von großer Bedeutung, einen besseren Überblick über die anstehende Arbeit zu bekommen. Die verschiedenen Klassen von Problemen, die im vorherigen Prozessschritt diskutiert wurden, werden in diesem Schritt überprüft.

Auf dieser Seite des Prozesses befindet sich das Scrum-Framework, das sich auf organisiertes und kausales Denken konzentriert sowie eine klare Verbindung zwischen Ursache und Wirkung herstellt. Es umfasst ein klares Verständnis, welches Ereignis ein anderes verursacht. Insbesondere sind hierfür sowohl die Kommunikation in verteilten Systemen als auch die Festlegung der Reihenfolge, in der verschiedene Anweisungen erfolgen, wichtig. Somit führt diese Denkweise zu einer kausalen Reihenfolge, in welcher der Output einer Aktion zur nächsten führt, beziehungsweise notwendig für dessen Ausführung ist. Dadurch können Prozesse vollständiger geplant werden. Darüber hinaus betont es die Analyse sowie Kategorisierung von konzeptuellem Wissen durch Vermittlung eines systematischen und logischen Denkprozesses.

Um Scrum korrekt anwenden zu können, muss der Anwender die wesentlichen Prinzipien der Agilen Softwareentwicklung (s. Abs. 3.4) beachten. Zunächst sollten die Funktionen sofort implementiert werden, sodass funktionsfähige Features dem Kunden schnellstmöglich in Form von Produktinkrementen geliefert werden können.

Darüber hinaus ist ein entscheidender Faktor die Eigenschaft Veränderungen anzunehmen, weshalb Scrum-Praktiker sich bewusst sein sollten, dass auf Veränderungen zu reagieren besser ist, als sie zu planen.

Außerdem ist der Umfang nicht festgelegt, sondern spiegelt sich anfangs des Projektplans in Form des Product Backlogs wider, welcher nach jeder abgeschlossenen Iteration aktualisiert werden kann. Genauso sind die üblichen Scrum-Ereignisse wie das Sprint Backlog, der Sprint selbst und der damit verbundene Daily Scrum, das Sprint Review sowie die Retrospektive enthalten (s. Abb. 12).

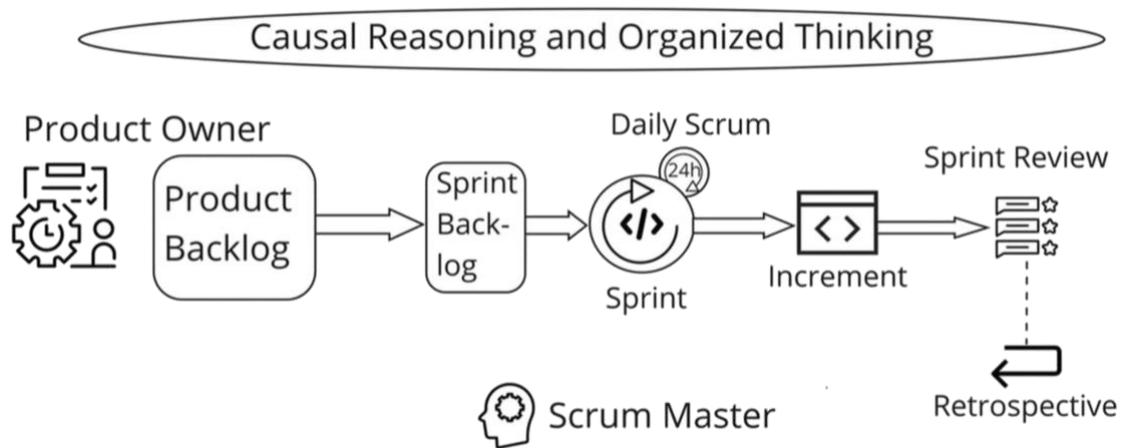


Abbildung 12: Scrum Process Area (Hehn et al., 2022)

Das Problem bei Scrum ist, dass die meisten Organisationen heutzutage die direkte Zusammenarbeit von Benutzern und Entwicklungsteam nicht zulassen. Die Prinzipien des DT könnten den Product Owner oder jeder anderen beteiligten Rolle, dabei unterstützen, ein besseres Verständnis dafür zu entwickeln, wie komplexe Probleme beherrscht, innovatives Denken gefördert sowie eine kundenorientierte Denkweise entwickelt werden können.

Design Thinking Process Area

Der dritte Prozessbereich beginnt ebenfalls mit dem Backlog-Management, da die Probleme hier ebenfalls strukturiert, kategorisiert und priorisiert werden müssen. Auf dieser Seite hingegen befindet sich der eher *laterale* DT-Ansatz.

Laterales Denken wird von Designern verwendet, um Probleme anhand einer Argumentation anzugehen, die nicht sofort ersichtlich ist. Es wendet kreative und indirekte Methoden an, sodass Praktiker offen für neue Perspektiven sind, Probleme aus anderen Blickwinkeln betrachten und neue Erkenntnisse gewinnen, mithilfe dessen neue innovative Lösungen gefunden werden können (The Interaction Design Foundation, 2016).

Der DT-Ansatz setzt laterales Denken ein, sodass er für die Analyse durch Intuition steht. Mentale Sprünge werden nicht als ungewöhnlich angesehen und jede Zwischenerkenntnis muss nicht zu Beginn sinnvoll sein. Vor allem sind ungewöhnliche Ideen nicht nur willkommen, sondern werden gefördert. Außerdem sollte man sich von gesetzten Ausgangssituationen und Rahmenbedingungen nicht einschränken lassen, sondern sie als Chance ansehen, um neue Perspektiven zu entwickeln. Es ist wichtig sich selbst herauszufordern und an etwas zu glauben, das den bisherigen Überzeugungen nicht entspricht. Indem man sich zwingt zunächst unkonventionell erscheinende Ideen zu akzeptieren, kann man schneller im DT-Prozess zu innovativen Lösungen gelangen. Somit endet DT mit einem guten Verständnis und Vertrauen in den generierten Ideen.

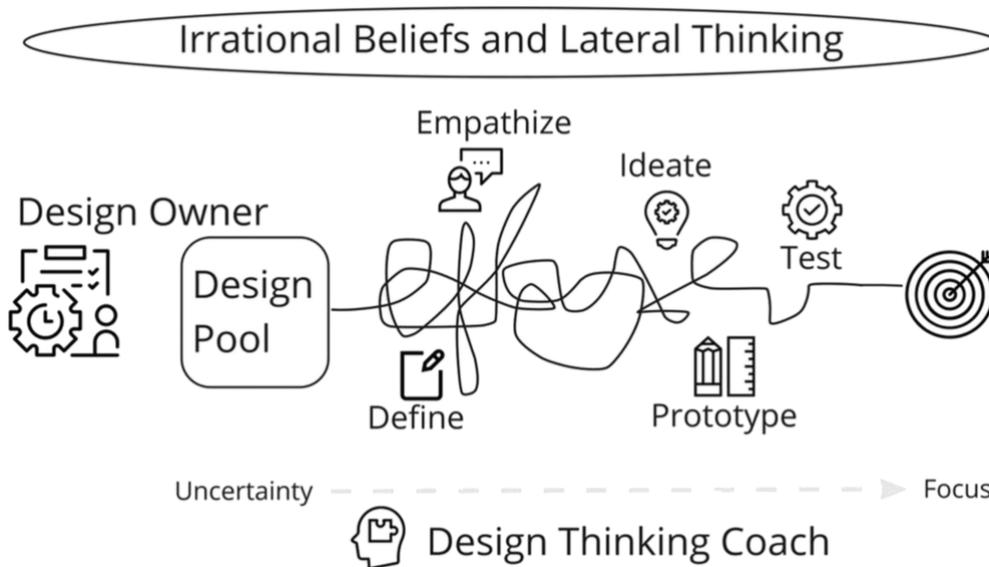


Abbildung 13: Design Thinking Process Area (Hehn et al. 2022)

Die Abbildung 13 zeigt die Illustration von Damien Newmann den Design Squiggle, die sich auf den Designprozess bezieht und den Prozessverlauf von der Recherche über die Entwicklung von Prototypen bis hin zur endgültigen Festlegung des klaren Designs beschreibt (Naiman, 2019).

Durch die kontinuierliche Einbindung von Kunden in mehreren Prototypen wird das Risiko, eine Lösung für eine nicht vorhandenes Problem zu entwickeln sowie die Unsicherheit über das richtige Problem, minimiert.

An dieser Stelle werden die verteilten Rollen erläutert. Die Rolle des Design Owners ist das Pendant zum Product Owner und trägt die Verantwortung für das Designergebnis, zum Beispiel das Mock-Up. Wichtig ist hier zu erwähnen, dass der Design Owner in Absprache mit dem DT-Team entscheidet, welche Elemente des Design Pools Teil des nächsten Designschritts sein sollten und zu klären, ob es sinnvoll wäre zu einem anderen Schritt zurück- oder vorwärtszugehen. Der Design Owner löst ähnlich wie bei Scrum den Design-Sprint aus, der jedoch iterativ vom DT-Team durchgeführt wird. Dies schließt sowohl in einigen Fällen Fachexperten als auch den DT-Coach ein. Er dient als Gegenstück des Scrum-Masters.

Product Backlog Design Matching

Die gemeinsame Vision stellt einen wichtigen Erfolgsfaktor da, welche wiederum entscheidend für die Vereinigung aller Aufgaben ist. Daher reicht es nicht aus, ausschließlich die beiden Prozesse parallel laufen zu lassen, um vom kollektiven Prozessansatz zu profitieren. Es muss gewährleistet sein, dass beide Frameworks einerseits von ihren Gegenstücken profitieren und andererseits gleichzeitig ihre eigenen Stärken nutzen.

Ein solchen kooperatives Vorgehen benötigt ständige Kommunikation und Abstimmung untereinander. Daher wurde der vierte Prozessbereich eingeführt, in welchem die Ereignisse des DTs, beispielweise nicht-funktionale Prototypen, als Eingabe für weitere Entwicklungszyklen verwendet werden sollten. Das bedeutet, dass die Erkenntnisse über Kundenbedürfnisse sowie beherrschbare Probleme in die Entwicklungssprints des Scrum-Teams eingebunden werden müssen.

An dieser Stelle geht es darum, die Arbeit zurückzubringen und voneinander zu lernen. Das Multidisciplinary Knowledge Café, das beim ersten Durchlauf verschiedene Menschen mit

unterschiedlichen Hintergründen und Persönlichkeiten zusammengebrachte, kann an diesem Punkt verwendet werden, um die getrennten Arbeitsgruppen wieder zusammenzuführen.

Für diesen konvergierenden Schritt kann ein Artefakt, ähnlich wie ein Backlog erneut als Werkzeug zur Zusammenführung dienen. Hiermit werden sowohl die Personen wieder zusammengebracht als auch die verschiedenen Leistungen, Ideen und Probleme in einem gemeinsamen Bereich namens *Product Backlog Design Matching* zusammengelegt. Es bietet den Teilnehmern die Möglichkeit, sich mit neu entwickelten Funktionen, verbesserten Mock-Ups sowie klar definierten Innovationsmöglichkeiten auseinanderzusetzen. Teil dieses Pools sind ebenfalls zuvor aufgetretene komplexe Probleme, die aufgrund fehlender klarer Definition nicht implementiert werden konnten.

Wenn man sich die parallel funktionierenden Prozesse auf ganzheitlicher Ebene anschaut, besteht während des Prozessverlaufs immer noch eine Verbindung zum Endbenutzer. Von ihm erhalten die Teilnehmer das Wissen darüber, welche neuen Funktionen in die nächsten Sprints integriert werden sollten. Darüber hinaus können Probleme diskutiert werden, um Missverständnisse auszuschließen sowie gegebenenfalls mit einem anderen Standpunkt eine alternative Lösungsmöglichkeit aufzuzeigen.

Die im Multidisciplinary Knowledge Café entwickelte Vision kann sich im Laufe der Zeit weiterentwickeln, indem das Kernproblem sowie das Ziel im Auge behalten werden. Ein ebenfalls unverzichtbarer und kritischer Meilenstein während des Lösungsentwicklungsprozesses ist die Wissensweitergabe. In einem DT-Team ist viel spezifisches Wissen vorhanden. Diese Wissensweitergabe wird dann notwendig, wenn aufgrund von Kapazitäts- oder Ressourcenknappheit ein neues Team aufgebaut wird oder auch Wissen an ein anderes Team weitergegeben werden soll. Insbesondere ist es kritisch, da DT kaum dokumentiert. Als unverzichtbar gilt es, weil Ideen existieren, die umgesetzt werden müssen, um überprüfen zu können, ob die Idee funktioniert.

Zusammengefasst kann man also festhalten, dass das Team DT-Stories durchführt. Bereits vorhandene User Stories aus dem Scrum Product Backlog können angepasst oder entfernt werden und neue erstellt werden.

Sobald das Product Backlog Design Matching abgeschlossen ist, beginnt die *tägliche* Arbeit erneut.

4.2.2 DT@Scrum

In der Herangehensweise von Häger et al. (2014) bietet Scrum den umfassenden Rahmen für alle Aktivitäten, wodurch die beteiligten Teams Sprints nutzen, nicht nur zur Verwaltung ihrer Aktivitäten während der Softwareentwicklung, sondern auch für das Management von Designaktivitäten. DT@Scrum führt das Design Planning ein, um Designteams dabei zu unterstützen, ein Gefühl für die Dauer und den Wert von DT-Aktivitäten zu entwickeln und ihnen zu ermöglichen, ihre kreative Arbeit besser zu strukturieren. Design Planning verwendet bereits aus Scrum bekannte Planungsmethoden wie Planning Poker für Designaktivitäten und umfasst die Erstellung eines Backlogs für Designaktivitäten, die Sprintplanung im Voraus sowie eine retrospektive Evaluation.

Zusätzlich schlägt dieser Ansatz drei verschiedene Operationsmodi vor: den *Design Thinking Mode*, den *Initial Development Mode* und den *Fully Integrated Mode*. Diese werden im Anschluss näher erläutert und unterscheiden sich lediglich im Verhältnis von DT- und Entwicklungsaktivitäten. Während der Design Thinking Mode den Fokus auf DT-Aktivitäten legt und sich der Initial Development Mode auf die Softwareentwicklung konzentriert, strebt der Initial Development Mode danach, die beiden Arten von Aktivitäten auszubalancieren. Dadurch ermöglicht er dem Team einen allmählichen Übergang von DT zur Softwareentwicklung.

Mit zunehmendem Verständnis für das Problem und die Anforderungen an eine Lösung verringert das Team stetig den Anteil an DT-Aktivitäten und steigert den Anteil an Softwareentwicklung. Der Design Thinking Mode dient der Erkundung des Problems sowie des Lösungsraums. Sobald das Team eine Produktvision entwickelt hat, die zur Problemlösung dient, kann es im Initial Development Mode beginnen, das Konzept durch die Implementierung erster Funktionen zu verfeinern.

Nachdem die Produktvision hinsichtlich der in Abschnitt 2.4 bereits erläuterten Grenzen – Feasibility, Desirability und Viability – verfeinert und getestet wurde, kann das Team zum

Fully Integrated Mode voran streiten. Dabei wird die Produktvision schrittweise entwickelt, bis das Softwaresystem vollständig implementiert ist.

Design Thinking Mode

In diesem Modus werden DT-Techniken zur Erkundung der Problemstellung und des Lösungsraums angewendet. Im Verlaufe dessen verfeinert das Projektteam das Problem und entwickelt eine Produktvision. Hauptereignisse an dieser Stelle sind die Entwicklung von einfachen Prototypen, mehrere User Stories und eine klare Produktvision.

Vor Sprint-Beginn sollten die Unternehmensstrategie und eine Problemstellung vorliegen. Genauso sollte Zugang zu potenziellen Benutzern und anderen Stakeholdern gewährleistet werden und den Teammitglieder eine DT-Schulung bereitgestellt worden sein.

Die **Aktivitäten** (s. Abb. 14) während dieses Modus folgen dem grundlegenden DT-Prozess, verwenden jedoch Scrum als Prozessrahmen.

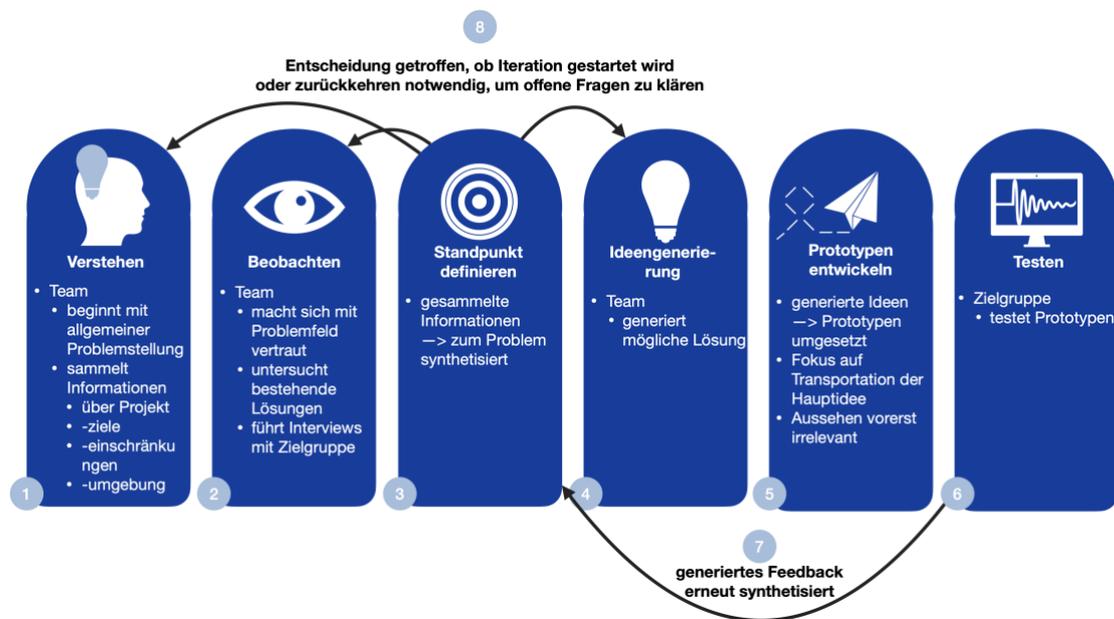


Abbildung 14: Aktivitäten des Design Thinking Modus (eigene Darstellung)

Außerdem nennen Häger et al. (2014) ebenfalls **Techniken**, die das Team in diesem Modus unterstützen können. Diese werden an dieser Stelle jedoch nicht näher thematisiert. Einige sind in der Tabelle 2 vorhanden.

Der DT-Modus umfasst vier **Rollen**; das Design Thinking Team, der (potenzielle) Nutzer, der Projektsponsor und ähnlich wie beim letzten Ansatz, der Design Thinking Coach.

Das *Design Thinking Team* trägt die Verantwortung für das Planen und Durchführen der Design Sprints. Es besteht meist aus drei bis sechs Personen aus unterschiedlichen Fachbereichen, dessen Expertise zur Durchführung des Projekts notwendig ist.

Der *potenzielle Nutzer* dient als Informationsquelle und soll zum Thema seine Meinung, Probleme und Feedback für Ideen, Prototypen oder andere Wegweiser für das Projekt äußern. Potenzielle Nutzergruppen werden vom Designteam interviewt sowie beobachtet, sodass es die Zielgruppe auf bestimmte Nutzer eingrenzen kann, mit denen es kontinuierliche Tests und Feedbackzyklen durchführen kann.

Der *Projektsponsor* hat ein großes Interesse an dem Projekt, da er meist die Gruppe repräsentiert, die die Ausgangsherausforderung definiert. Er dient für das Team als Kontaktperson und trägt die Verantwortung Partner zu interviewen und Einführungsmaterial für die Herausforderung bereitzustellen. Zu seinen Aufgaben gehören ebenfalls das Vereinfachen der Kommunikation mit anderen Bereichen des Unternehmens, Maßnahmen zu ergreifen, sodass verschiedene Teams besser zusammenarbeiten können und die Wiederverwendung von bereits vorhandener Software zu ermöglichen. Außerdem äußert er ähnlich wie der Nutzer Feedback sowohl an Ideen als auch Prototypen aus. In einer Unternehmensumgebung wird diese Rolle von einer Führungskraft übernommen.

Design Thinking Coaches tragen die Verantwortung für die Einführung nützlicher Techniken, die Moderierung von Diskussionen sowie Teamdynamikhindernissen. Zudem stellen sie sicher, dass das Team fokussiert an den Aufgaben arbeiten kann. Im Scrum-Prozess kümmert sich der Scrum Master darum, dass das Team der Prozessstruktur folgt und moderiert während der Projektplanung und Reflektionsmeetings. Häger et al. (2015) sagen, dass diese Rollen in diesem Ansatz zu einem verschmelzen können, und zwar dem *Process Master (PM)*.

Als **Ergebnis** dieses Modus sind mehrere verschiedene, vereinfachte Prototypen und ein eher ausgearbeiteter Lösungsprototyp. Letzteres sollte zusammen mit den gesammelten Erkenntnissen eine klare Lösungsvision darstellen und erläutern, weshalb alle Aspekte des Prototypens so entworfen worden sind. Zusätzlich müssen nicht-funktionale Anforderungen für die Produktimplementierung und User Stories, die die Kernfunktionalität des zu implementierenden Produkts beinhalten, definiert werden. Die Erkenntnisse, die zu den Prototypen führten, sollten in Form von Dokumentationen festgehalten werden, damit die Entscheidungen, die in diesem Modus getroffen wurden, zurückverfolgt werden können.

Initial Development Mode

Nun wird die aus dem vorherigen Modus resultierende Produktvision in dem Initial Development Mode weiter erforscht. Das Team zielt darauf ab, in dieser Phase mit der Implementierung, Testung und Verfeinern verschiedener Aspekte der Lösung zu beginnen. Ergebnisse dieses Modus sind unter anderem verfeinerte und ausgearbeitete Prototypen und User Stories, aber auch nicht-funktionale und technische Anforderungen.

Zusätzlich zu den im vorigen Modus erwähnten **Voraussetzungen** sollte nun folglich eine klare Produktvision und eine Reihe an ausgearbeiteten User Stories vorhanden sein. Genauso sollten aus der DT-Phase sowohl funktionale als auch vereinfachte Prototypen vorliegen. Anschließend sollte eine initiale Liste von nicht funktionalen Anforderungen zur Verfügung gestellt werden.

Bei den **Aktivitäten** dieses Modus angekommen, steht in dessen die Verfeinerung des Lösungsprototyps und der Produktvision im Fokus. Hierbei werden Funktionen oder Designaspekte des Lösungsprototyps identifiziert, für die weitere Klarstellungen sowie Tests hinsichtlich der Machbarkeit erforderlich sind. Diese können dann in Form von *User Experience* (UX) Prototypen, einer Proof-of-Concept-Funktionsimplementierung oder einer Implementierung, die mögliche Technologien erkundet, prototypisiert werden. Alle Prototypen werden innerhalb (Scrum) Sprints entwickelt. Prototypen, die eine Benutzeroberfläche bieten sollte mit Zielbenutzern getestet werden, sodass eine maximale Benutzerzufriedenheit gewährleistet werden kann. Diese aus der Prototypentwicklung und -testung generierten Informationen können wiederum verwendet werden, um sowohl die Implementierung als auch die User Stories in den nächsten Sprints weiter zu verfeinern sowie zusätzliche nicht-

funktionale und technische Anforderungen hinzuzufügen. Darüber hinaus sollte die Systemarchitektur sowie das Integrationskonzept prototypisiert werden.

In diesem Modus kann das Team grundlegende **Techniken** zur Unterstützung einsetzen; *Low-Fidelity UX-Prototyping* ist das Entwerfen von simplen Skizzen auf Papier, die eine schnelle Erstellung von UI-Prototypen und Tests verschiedener Interaktionskonzepte ermöglichen. *Mid-Fidelity UX-Prototyping* wie Wireframing vereinfacht die Erstellung eines Benutzeroberflächenprototyps. *User Story Mapping* ist eine weitere Methode, die angewendet werden kann, um die Funktionalität eines Systems zu verstehen und Release-Pläne zu erstellen, indem die Hauptaktivitäten in einer sinnvollen Reihenfolge angeordnet werden. Zuletzt wäre es ebenfalls sinnvoll täglich die aktuellen Prototypen anzuschauen, da dies sicherstellt, dass das gesamte Team über die Entwicklung sowie Ergebnisse informiert ist.

Während des Initial Development Mode sind verschiedene **Rollen** beteiligt, die jeweils spezifische Aufgaben besitzen:

Das *Scrum Team* trägt ähnlich wie das Design Team die Verantwortung für die Planung und Durchführung der Entwicklungssprints. Es besteht in der Regel jedoch aus acht bis zehn Entwicklern, die aus dem Design Team des vorrangigen Modus stammen sowie zusätzlichen Entwicklern aus verschiedenen Fachbereichen, die je nach Bedarf für die Softwareimplementierung erforderlich sind, beispielweise Backend-, Frontend-Entwickler oder Datenbankexperten.

Vom *potenziellen Nutzer* ist, das Testen der verschiedenen Prototypen und dazu Feedback zu äußern, die Hauptaufgabe.

Die des *Projektsponsors* ist es ebenfalls Feedback zu den entwickelten Prototypen zu geben, hinzu kommt jedoch die Kundgabe der allgemeinen Ausrichtung des Projekts. Zusätzliche Verantwortungen umfassen die Kommunikationsförderung mit anderen Unternehmensabteilungen sowie die Werbung für den Projektfortschritt.

Der *Product Owner* vertritt den Kunden und ist verantwortlich für die Befüllung des Backlogs mit User Stories sowie deren Priorisierung. Er kann im kombinierten Prozessmodell ein Mitglied des Design-Teams sein.

Die Aufgaben des *Prozess Masters* sind denen aus dem DT-Modus gleichzustellen.

In diesem Modus stehen hauptsächlich die erstellten Prototypen und dessen Testergebnisse im Vordergrund. Zu den *Ergebnissen* gehören UX-Prototypen, die vom Benutzer getestet worden sind und zu weiteren sowohl funktionalen als auch nicht-funktionalen Anforderungen geführt haben. Die Erkenntnisse aus der Entwicklung und der Prototypentests ermöglichen es, die User Stories weiter zu verfeinern und sowie neue hinzuzufügen. Abschließend muss eine klare Spezifikation der Integration innerhalb des Unternehmenskontexts erstellt werden. Dazu gehört die Identifizierung von Wechselwirkungen mit anderen bereits vorhandenen Systemen oder das Potenzial zur Wiederverwendung vorhandener Softwarekomponenten in der endgültigen Implementierung.

Fully Integrated Mode

Der Fully Integrated Mode folgt hauptsächlich einem Scrum-Entwicklungsprozess, der es dem Team ermöglicht, in inkrementellen Schritten auf ein endgültiges Produkt hinzuarbeiten. Sollten währenddessen Hindernisse auftreten, werden DT-Tools ad hoc eingesetzt. Um schnell auf diese Hindernisse mit den richtigen Werkzeugen reagieren zu können, muss der Entwicklungszyklus genau beobachtet werden.

Zu den bereits im Initial Development Mode genannten **Voraussetzungen** kommen weitere hinzu, die vor der Durchführung der Sprints vorhanden sein sollten. Dazu gehören eine nach Priorisierung geordnete Liste der detaillierten technischen Anforderungen und eine der technischen Anforderungen. Dem hinzuzufügen sind eine Menge an UX-Prototypen sowie eine Menge an Proof-of-Concept Implementationen.

Sobald die Voraussetzungen erfüllt sind, kann man mit den für diesen Modus spezifizierten **Aktivitäten** starten. Diese folgen anhand von Scrum als Prozessframework einem gewöhnlichen Softwareentwicklungsansatz. Der Fokus des Teams liegt auf der Implementierung der Inkremente sowie Bereitstellung und Wartung der Software. Es kann die Entscheidung treffen, falls nicht-ausreichende Funktionsdefinierung oder Probleme aufkommen, kurzzeitige DT-Impulse in die Aktivitäten einzuleiten, um die Funktionsidee zu verfeinern oder das Problem zu lösen. Im Vergleich zu den anderen beiden Modi konzentriert sich DT in diesem Operationsmodus nicht darauf, Erkenntnisse vor dem

Softwareentwicklungsprozess zu generieren, sondern schafft spontane Erkenntnisse und unterschiedliche Lösungen zum Bewältigen einiger unüberwindbarer Hindernisse.

Außerdem umfasst dieser Modus ausschließlich die Umsetzung der Produktvision in eine voll funktionsfähige Software und ermöglicht daher die Nutzung der gesamten **Softwaretechniken**. Als Beispiel an dieser Stelle kann man die Praktiken des Extreme Programming (Beck, 2000) nennen, das sehr geeignet für Scrum-Projekte ist. Dazu gehören selbstverständlich auch Test Driven Development (TTD), Continuous Integration (CI) und verschiedene Überprüfungstechniken zur Erhaltung der Code-Qualität. Zusätzlich kommen kontinuierliches Kunden-Testing und kollektives Code-Eigentum hinzu.

Die **Rollen** in diesem Modus ähneln denen des vorherigen Modus; das *Scrum Team* kümmert sich auch hier um die Planung sowie Durchführung der Sprints zur Implementierung von funktionalen Software-Inkrementen. Es kann bei Bedarf weitere Scrum-Teams involvieren, sodass eine parallele Entwicklung gewährleistet werden kann.

Ebenso ist die Hauptaufgabe der *potenziellen Benutzer* während dieses Modus, die Inkremente zu testen und Feedback zu geben.

Genauso wird vom *Projektsponsor* Feedback zu den entwickelten Inkrementen erwartet. Darüber hinaus muss er ähnlich wie beim Initial Integration Mode die Kommunikation mit den verschiedenen Unternehmensabteilungen erleichtern sowie den Projektfortschritt fördern.

Der *Product Owner* besitzt übereinstimmenden Verantwortlichkeiten mit dem vorherigen Modus.

Auch der *Process Master* muss dieselben Pflichten wie in den anderen Modi erfüllen. Hinzu kommt jedoch eine wichtige weitere Aufgabe, und zwar muss er schnell reagieren können, wenn Blockaden den Entwicklungsprozess stoppen, und die richtigen DT-Tools bereitstellen, um dem Team zu helfen.

Da sich der Fully Integrated Mode auf die Erstellung von getesteter und funktionierender Software konzentriert, sollte als Ergebnisse dieses Modus alle Entwicklungen potenziell bereitstellbar sein. Das bedeutet konkret, dass die Software bestimmten Produktstandards entspricht und bereitgestellt werden kann. Somit sollte die Teams bereits Strategien zur

Lieferung der Software an den Endnutzer entwickeln. In manchen Fällen ist dies einfacher als in anderen, beispielsweise bei mobilen Apps. Bei der Entwicklung von On-Premise-Software, dessen Integration in bestehende Systemlandschaften notwendig ist, muss das Team explizit Zeit zur Erstellung einer Deployment-Strategie einplanen. Dies sollte möglich sein, da die Entwickler nicht nur unreflektiert User Stories umsetzen, sondern anpassungsfähige Software entwickeln und offen für potenzielle Verbesserungen sind. Dementsprechend sollten sie auch ihre eigenen Ideen oder Vorschläge erfassen und in User Stories für zukünftige Sprints umwandeln.

4.3 Vergleich der Ansätze

In diesem Abschnitt werden nun die präsentierten Integrationsansätze anhand der nachfolgenden Punkte einander gegenübergestellt.

Schwerpunkt

Der Ansatz nach Häger et al. (2014) kombiniert DT mit dem Scrum-Framework, um einen ganzheitlichen Ansatz für die Softwareentwicklung zu bieten, während das Konzept nach Hehn et al. (2022) die Integration von DT in den Scrum-Prozess betont, um die Benutzerzufriedenheit und Entwicklung innovativer Lösungen zu fördern. Dementsprechend besitzen sie leicht unterschiedliche Schwerpunkte.

Einerseits legen Häger et al. (2014) einen besonderen Wert auf die schrittweise Integration von DT-Aktivitäten in den gesamten Entwicklungsprozess. Hehn et al. (2022) konzentriert sich stark auf die Rolle des Design-Owners und die Verwendung von DT-Techniken.

Während Häger et al. (2014) den gesamten Entwicklungsprozess von der Problemerkundung bis zur Implementierung abdecken, liegt der Schwerpunkt von Hehn et al. stärker auf der Benutzerzentriertheit und der kontinuierlichen Validierung von Ideen durch Prototyping und Feedback.

Frameworks

Beide Frameworks basieren auf der Idee der Integration von Design Thinking und Scrum, weisen jedoch einige Unterschiede in ihrer Struktur sowie Schwerpunktsetzung auf. Sie betonen die Bedeutung einer benutzerzentrierten Entwicklung von Softwarelösungen,

unterschieden sich jedoch leicht in deren Herangehensweise sowie Schwerpunktsetzung. Während bei DT@Scrum der Entwicklungsprozess in klar definierte Phasen gegliedert ist, die nacheinander ablaufen, werden beim kollektiven DTScrum-Ansatz der Scrum- und DT-Prozessbereich parallel durchgeführt. Beide Konzepte betonen die Bedeutung einer benutzerzentrierten Entwicklung von Softwarelösungen, unterscheiden sich jedoch leicht in ihrer Herangehensweise und Schwerpunktsetzung.

Rollen

In der Tabelle vier werden die ähnlichen und vergleichbaren Rollen der beiden Konzepte gegenübergestellt.

Tabelle 4: Vergleichbare Rollen von DT@Scrum und DTScrum

Collective Framework DTScrum	DT@Scrum
Product Owner	Product Owner / Projektsponsor
Scrum Master	Process Master
Entwicklungsteam	Scrum-Team
DT-Team	DT-Team
Design Owner	nicht vorhanden
DT-Coach	DT-Coach
nicht explizit genannt	Nutzer

Sowohl die Product Owner beider Ansätze als auch der Projektsponsor nach Häger et al. (2014) haben ähnliche Verantwortlichkeiten, die die Vertretung der Interessen der Kunden und Stakeholder umfassen. Genauso so sind sie verpflichtet die Anforderungen zu priorisieren. Zusätzlich trägt der Projektsponsor jedoch auch die Verantwortung der Kommunikation mit anderen Unternehmensbereichen sowie die Förderung des Projektfortschritts.

Dem Scrum Master kann der Process Master gleichgesetzt werden, da von beiden erfordert wird, dass sie die Teams bei der Umsetzung des jeweiligen Prozessrahmens unterstützen, Hindernisse beseitigen und die Selbstorganisation fördern.

Das Entwicklungsteam und das Scrum-Team ähneln sich in dem Sinne, dass sie für die Umsetzung der Backlog-Elemente zuständig sind. Außerdem sind deren Rollen sowie Verantwortlichkeiten in beiden Ansätzen ähnlich.

Die DT-Teams der beiden Konzepte bestehen aus Teammitgliedern, die unterschiedliche Fachkompetenzen aufweisen. Bei DT@Scrum übernimmt das Team zusätzlich die Planung, da kein Design Owner parat steht. Beim Collective Framework DTScrum hingegen kümmert sich das Team ausschließlich um die Durchführung der Design Aktivitäten.

Genauso ähneln sich die DT-Coaches, da sie das Team bei der Anwendung von DT unterstützen, Workshops durchführen sowie kreatives Denken fördern.

Auch wenn beim kollektiven Prozessrahmen die Rolle des Nutzers nicht explizit genannt wurde, kann man davon ausgehen, dass dieser dennoch genauso involviert im Prozess ist wie der des DT@Scrum-Ansatzes.

4.4 Handlungsempfehlung

Basierend auf der Vorstellung und dem Vergleich der Ansätze von Hehn et al. (2022) und Häger et al. (2014) lassen sich verschiedene Erkenntnisse ableiten, die bei der Auswahl des geeigneten Konzeptes für ein bestimmtes Projekt berücksichtigt werden sollten. Nachfolgend werden Handlungsempfehlungen auf Grundlage dieser Erkenntnisse präsentiert, um Unternehmen bzw. Personengruppen, die ein geeignetes Konzept in einem Projekt anwenden möchten, dabei zu unterstützen, den für ihre spezifischen Anforderungen am besten geeigneten Ansatz zu wählen.

Beide Ansätze bieten strukturierte Methoden zur Integration von DT und Scrum, weisen jedoch unterschiedliche Schwerpunkte sowie Frameworks auf. Die Wahl zwischen beider hängt von den spezifischen Anforderungen und Zielen eines Projekts ab.

Der Ansatz von Hehn et al. (2022), sollte verwendet werden, wenn der Projektschwerpunkt auf dem Innovationsprozess liegt und ein digitales Produkt oder eine digitale Dienstleistung entwickelt werden soll. Genauso sollte man diesen Ansatz wählen, wenn noch eine eher konservative Unternehmenskultur herrscht und ein klar strukturiertes Phasenmodell bevorzugt wird, das dennoch DT und Scrum integriert. Dieses Konzept ermöglicht, aufgrund der

individuellen Ausführung von jeweils DT und Scrum, ein klares Verständnis für die verschiedenen Phasen, sodass die Einführung innerhalb eines Projekts eventuell leichter auszuführen ist.

Die Anwendung des Ansatzes nach Häger et al. (2014) sollte in Erwägung gezogen werden, wenn man bereits mit dem Scrum-Framework vertraut ist und die Suche nach einem Entwicklungsprozess ist, der DT integriert. Dieses Konzept sollte man ebenfalls auswählen, wenn man die Flexibilität schätzt, auf Basis des Projektschwerpunkts zwischen verschiedenen Operationsmodi zu wählen. Ob der Fokus eher auf DT liegen soll oder vielmehr auf Scrum, kann das Projektteam beliebig zwischen den Modi variieren. Darüber hinaus kann dieser Ansatz ebenfalls angewendet werden, wenn ein schrittweiser Übergang von DT zur Softwareentwicklung gewünscht ist, sodass sich das Team allmählich auf die Implementierung konzentrieren kann.

Außerdem hängt die Auswahl zwischen den beiden Ansätzen von den spezifischen Anforderungen und Zielen eines Projektes sowie der vorhandenen Ressourcen ab. Wenn bereits ein etabliertes Scrum-Team vorhanden ist und die Integration von Designaktivitäten in den Sprint-Planungsprozess erfolgen sollen, könnte der Ansatz von Häger et al. die richtige Wahl sein. Wenn jedoch ein eigenes DT-Team verfügbar ist, wäre die Wahl des Ansatzes von Hehn et al. sinnvoller.

Nichtdestotrotz ist es in beiden Fällen wichtig, dass vor der Auswahl die spezifischen Anforderungen und Ziele eines Projekts sorgfältig geprüft worden sind, um die richtige Methode auszuwählen, die am besten zu den Bedürfnissen des Projekts passt.

4.5 Herausforderungen und Hindernisse bei der Integration

Trotz der vielversprechenden Potenziale, die die Integration von Design Thinking und Scrum bietet, gibt es auch Herausforderungen, die es bei der Integration zu überwinden gilt. Das liegt daran, dass noch wenig Erfahrung und Wissen darüber existiert (Hehn et al., 2022). „Without appropriate reflection to help clarify the basis of communication among all the participants, there is little hope of understanding the foundations and value of DT in an increasingly complex technological culture.“(Buchanan, 1992, S. 8). Auf Grundlage dessen wurden von Hehn et al.

(2022) mehrere Experteninterviews durchgeführt, um eine solche Kommunikationsgrundlage zu definieren sowie weitere Herausforderungen zu identifizieren. Die Erkenntnisse dieser Forschung werden in diesem Abschnitt präsentiert.

Ressource Allocation

Ein Hauptproblem, das bei der Integration auftreten kann, ist die Zuteilung von Ressourcen. Die meisten Unternehmen sehen Ressourcenbeschränkung als eine kritische Thematik an, da es selten vorkommt, dass sich ein Angestellter vollkommen auf ein bestimmtes Problem fokussieren kann, ohne abgelenkt zu werden. Insbesondere ist es schwierig, sich auf die Ressourcenkapazitäten zu einigen, da DT die Einbeziehung des Nutzers sowie das Fachwissen von Personen aus unterschiedlichen Bereichen erfordert.

Aufgrund dessen wird betont, dass die Ressourcen vor Projektbeginn zugewiesen werden müssen. Hierbei ist die genaue Abstimmung der Belegung sowie des Zeitpunkts, zu dem sie voraussichtlich verfügbar sind, von Bedeutung. Die Aufmerksamkeit und Priorisierung der Arbeit sollte auf der effizienten Verwendung der verfügbaren Ressourcen liegen, sodass die Ressourcennutzung so gesteuert werden kann, dass sowohl komplexe als auch gewöhnliche Probleme entdeckt werden können. Weiterhin wird darauf hingewiesen, dass zu viele Ressourcen mit Nutzerforschungsthemen und Ideengenerierungssitzungen vor der eigentlichen Entwicklungsphase belegt werden könnten, wenn DT als vorrangiger Prozess vor den Scrum Schritten stattfindet.

Dies bringt eine weitere Herausforderung mit sich; ein begrenztes Budget für ein nutzerzentriertes Designprojekt sowie der Druck des Managements zeitnah etwas Lieferbares zu entwickeln. Es ist von großer Bedeutung, dass das Management Verständnis dafür aufweist, dass die Integration von DT in Softwareentwicklungsprojekten eine zeitaufwendigere Verpflichtung ist, ohne eine Garantie für eine erfolgreiche Produktidee als Ergebnis. Jedoch bringt jede Erkenntnis, die in den Phasen der Problemerkennung sowie -definierung gewonnen wurde, das Team näher an ein besseres Verständnis der Kundenanforderungen. Daher scheint sich die Ressourceninvestition zu rechtfertigen.

Competing Views and Different Kinds of Problems

Obwohl DT und Scrum vielen Gemeinsamkeiten aufweisen, die in Abschnitt 4.1 thematisiert wurden, sind sie in ihrer Natur dennoch verschieden (Hehn et al., 2022). Während Scrum auf traditionellen ingenieurstechnischen Paradigmen und analytischem Denken beruht, basiert DT auf heuristischem Denken und situativem Verständnis (Plattner et al., 2012).

Hehn et al. (2022) betonen, dass die unterschiedlichen Zwecke sowie Arten von Problemen, die von den Frameworks behandelt werden, bei der Integration zu Konflikten führen können, sodass sich die Frage stellt, wie man beide Frameworks trotz dessen zusammenführen kann. Nach Plattner et al. (2012) wird eine effektive technische Realisierung von IT-Ingenieuren unterstützt, wobei DT hingegen den Umgang mit der Mehrdeutigkeit von Designproblemen ermöglicht, indem es sie als *Wicked Problems* betrachtet. Das ist die Bezeichnung für komplexe Probleme, die aufgrund ihrer komplizierten Natur und den Einfluss vielzähliger Faktoren oft schwer oder gar nicht zu lösen sind. Es mangelt an Klarheit in ihren Zielen sowie Lösungen. Dadurch, dass sie an Konstanten der realen Welt gebunden sind, gibt es keine risikofreien Versuche eine Lösung zu finden (Interaction Design Foundation, 2016).

Diese Spannung zwischen den gegensätzlichen Ansätzen ist Hehn et al. (2022) zufolge in der Praxis zu spüren, da auch die unterschiedlichen Denkweisen der Teams eine Herausforderung darstellen können. Eine harmonische Zusammenarbeit zu erreichen, ist schwierig, vor allem wenn die offeneren Ansätze der Designer auf die analytischen der Scrum-Praktiker treffen. Während Scrum-Anwender empfinden, dass sie von den Design-Praktikern zurückgehalten werden, sind DTs verärgert über die Zweifel der Entwickler gegenüber der Wertschöpfung der Prozessschritte. Daher ist von allen Teammitgliedern eine Sensibilisierung für die gegensätzliche Denkweise gefragt.

Nichtsdestotrotz muss unterstrichen werden, dass die Fähigkeit des Scrum Frameworks, das Problem vollständig zu verstehen und daraufhin Anforderungen zu definieren, begrenzt ist, denn „[a]pproaches, like Scrum, are not yet able to cope with a class of problems demanding techniques reaching from further exploring problems to fully understand them.“ (Hehn et al., 2022, S. 91). Menschenzentrierte Designansätze sind insbesondere in schnelllebigen technologischen Umgebungen erforderlich, um diese wicked problems anzugehen. DT ermöglicht das Herausfordern von Annahmen, Neubewerten komplexer Probleme sowie

Erkunden neuer Lösungsansätze. Im Gegensatz dazu ist Scrum auf die schnelle Entwicklung von Lösungen für klar definierte Probleme ausgerichtet, sodass es für User Stories in Frage kommt, die verständlich definiert worden sind und bei denen kein Interpretationsraum gegeben ist (Hehn et al., 2022). Exemplarisch hierfür kann ein einfacher Button sein, dessen Umsetzung in der Informatik als Basiswissen gilt. Einen ganzen Design-Prozess durchzuführen, wäre an dieser Stelle überflüssig, sodass direkt mit der Entwicklung des Knopfs gestartet werden kann.

Coordination and Communication

Wenn zwei Welten – in diesem Fall Design und IT – aufeinandertreffen, muss es eine konstante Kommunikation zwischen allen Teilnehmern geben, damit überhaupt eine Basis zum Erfolg geschaffen werden kann. Da dieser Prozess Menschen mit diversen Erlebnissen sowie Hintergründen inkludiert, kann es oftmals schnell zu Konflikten führen. Dies kann nur mit effektiver Kommunikation verhindert werden. Das bedeutet nicht, dass man die Kalender der Teilnehmer mit zahlreichen Meetings füllt. Dennoch sollte eine adäquate Lösung gefunden werden zur Vermeidung von zu weniger Kommunikation.

Ein weiterer wichtiger Punkt ist, dass man die Teilnehmer von Anfang an involviert. Hier ist insbesondere die Rede von denjenigen, die die Verantwortung für den Output tragen. Wenn diese, nachdem alles besprochen wurde, eine Liste mit den Anforderungen zur Umsetzung erhalten, könnte dies zu mangelnder Motivation und Verringerung der Verpflichtung gegenüber dem Projekt führen.

Ebenso spielt die organisatorische Struktur des Unternehmens eine Rolle. Wenn diese beispielsweise eine hierarchische Struktur aufweisen und Designanforderungen seitens des Managements abgegeben werden, kann die Effektivität von DT reduziert werden. Daher muss an dieser Stelle ebenfalls eine gute Balance zwischen Unternehmensanforderungen und kreativer Freiheit bestehen (Häger et al., 2014).

Außerdem sollten die Rollen sowie aufgeteilten Aufgaben innerhalb des Projekts von Anfang an klar sein. Dies mindert das Wettbewerbsgefühl unter den Teilnehmern, sodass die Arbeit anderer weniger die eigene bedroht (Hehn et al., 2022).

Trotz der vielversprechenden Potenziale, die die Integration von Design Thinking und Scrum bietet, gibt es auch Herausforderungen, die es zu überwinden gilt. Insbesondere die

unterschiedlichen Anforderungen und Herangehensweisen beider Ansätze können zu Konflikten führen und die reibungslose Umsetzung erschweren. Die Schwierigkeiten bei der Ressourcenzuweisung und die Notwendigkeit einer konstanten Kommunikation innerhalb des Teams sind nur einige der Hindernisse, die auftreten können.

Nichtsdestotrotz ist es wichtig, diese Hindernisse nicht als unüberwindbare Barrieren anzusehen, sondern als Herausforderungen, die es zu bewältigen gilt, um die Vorteile der Integration voll auszuschöpfen. Durch eine offene Kommunikation, eine klare Rollenverteilung und die Bereitschaft, sich anzupassen, können diese Hindernisse überwunden werden.

5 Fazit & Ausblick

Dieses finale Kapitel umfasst zusammengefasst die wichtigsten Erkenntnisse dieser Arbeit. Dabei wird untersucht, ob das Ziel dieser Arbeit erreicht werden konnte. Des Weiteren erfolgt ein Ausblick, der thematisiert, wie Forschende oder andere Personen, die sich mit der Integration von DT in Scrum auseinandersetzen und in Überlegung sind ein verbündetes Konzept bei sich im Unternehmen einzusetzen, diese Arbeit weiterbehandeln könnten.

5.1 Fazit

Das erste Ziel dieser Arbeit war es, die Grundlagen von DT und Scrum vorzustellen, um das nötige Fachwissen als Basis darzulegen, sodass anhand dessen der weitere Verlauf der Arbeit verstanden werden kann.

Das Kapitel über DT thematisiert zunächst die Herausforderung, eine klare Definition zu finden, aufgrund der Vielseitigkeit des Konzepts und der unterschiedlichen Auffassungen des Begriffs *Design*. Es werden verschiedene Definitionen präsentiert, die zu einem besseren Verständnis von DT führen. Weiterhin werden drei Formen des DTs identifiziert: als Prozess, als Mindset und als Toolbox. Diese drei Aspekte wurden näher erläutert, wobei betont wurde, dass erst ihre Kombination zu einer erfolgreichen Anwendung beiträgt.

Darüber hinaus wurden verschiedene DT-Vorgehensmodelle vorgestellt, die einen iterativen und nutzerorientierten Prozess beschreiben, sowie die Phasen des divergenten und konvergenten Denkens erklärt. Dies ist wichtig zum grundlegenden Verständnis des weiteren Verlaufs der Arbeit.

Weiterhin wurde das DT-Mindset tiefgründig erläutert, welches grundlegende Prinzipien wie Empathie, Zusammenarbeit, Risikobereitschaft, Neugierde, Achtsamkeit, Kreativität, Handlungsorientierung, den Wunsch nach Veränderung und die Fähigkeit, kritisch zu hinterfragen, umfasst. Erfahrene Anwender nutzen diese Prinzipien, um innovative sowie kundenorientierte Lösungen zu entwickeln.

Als nächstes wurde DT als Toolbox betrachtet, die eine Vielzahl von Methoden und Werkzeugen umfasst. Diese können je nach Bedarf im Projekt eingesetzt werden, um Probleme zu verstehen, Ideen zu generieren und Prototypen zu entwickeln. Verschiedene Tools können verwendet werden, um optimale Ergebnisse zu erzielen. Dies wurde erläutert, da im weiteren Verlauf der Arbeit einige Tools sowie Techniken als Beispiele für ein besseres Verständnis herangezogen wurden.

Weiterführend wurden die drei Kernkonzepte vorgestellt: People, Place und Process. Erfolgreiche Projekte können durch Teams mit vielfältigen Fähigkeiten und Perspektiven erreicht werden. Eine flexible Raumgestaltung fördert die Zusammenarbeit des Teams. DT soll nicht als starrer Lösungsansatz gesehen werden, sondern als Toolbox mit verschiedenen Methoden, die in einem strukturierten iterativen Prozess angewendet werden.

Dieses Kapitel wurde mit dem DT-Vorgehensmodell bestehend aus sechs Phasen abgeschlossen: Verstehen, Beobachten, Standpunkt definieren, Ideen generieren, Prototypen entwickeln und Testen. In der Verstehen-Phase wird das Problem gemeinsam erarbeitet, während in der Beobachten-Phase ein tiefes Verständnis der Nutzer entwickelt wird. Anschließend wird in der Standpunkt-definieren-Phase eine klare Problemstellung festgelegt. Basierend darauf werden dann Ideen in der vierten Phase generiert, gefolgt von der Entwicklung von Prototypen in Phase fünf. In der letzten Phase, dem Testen, werden die Prototypen an potenziellen Nutzern getestet, um Feedback von diesen Nutzern zu erhalten und die Ideen weiter zu verbessern.

Insgesamt bietet das Kapitel einen umfassenden Einblick in die Kernkonzepte des DTs, sodass eine solide Grundlage für das Verständnis und die Anwendung dieses Konzepts geschaffen wurde und der erste Teil des ersten Ziels erreicht wurde.

Weiterhin bot das Kapitel über Scrum einen zusammengefassten Einblick in das agile Framework. Es betont die empirischen Scrum-Säulen, die Iteration und das Inkrementieren sowie die Zusammenarbeit in multidisziplinären Teams. Scrum basiert auf klaren Werten wie Transparenz, Überprüfung und Anpassung, die durch dessen formale Ereignisse unterstützt werden. Die Rollen innerhalb des Scrum-Teams sowie die Artefakte wurden detailliert erläutert. Zusammen bilden sie den Scrum-Prozess, der darauf abzielt, kontinuierlich Wert durch adaptive Lösungen für Probleme zu generieren.

Zusammenfassend bietet dieses Kapitel also einen eingehenden Einblick in die essenziellen Prinzipien und Bestandteile von Scrum, wodurch eine solide Basis für das Verständnis und die praktische Anwendung dieses agilen Frameworks geschaffen wurde und somit der zweite Teil des ersten Ziels dieser Thesis erfüllt wurde.

Mit Kapitel vier sollte das zweite Ziel diese Arbeit, die Vorstellung von zwei bereits konzipierten Ansätzen zur Integration, die miteinander verglichen werden sollten, erfüllt werden. Basierend darauf wurde eine adäquate Handlungsempfehlung für Unternehmen ausgesprochen.

In diesem Kapitel wurde diskutiert, wie DT und Scrum erfolgreich integriert werden könnten. Zunächst wurden die Synergien zwischen den beiden Ansätzen beleuchtet. Dies umfasst, dass beide Methoden ein iteratives Vorgehen, eine Nutzerzentrierung, Experimentation und Prototyping, Kollaboration und Multidisziplinarität, Flexibilität und Anpassungsfähigkeit sowie Ergebnisfokussierung betonen.

Weiterhin wurden zwei verschiedene Ansätze zur Integration von DT in das Scrum-Framework detailliert vorgestellt. Das erste Konzept von Hehn et al. (2022) ist das Collective Framework DTScrum. Hierbei werden DT und Scrum parallel genutzt, sodass aus den Stärken beider ein vollständiger Nutzen gezogen werden kann. Dieses Konzept umfasst Bereiche wie das Multidisciplinary Knowledge Café, die Design Thinking Process Area, die Scrum Process Area und das Product Backlog Design Matching.

Der zweite Ansatz von Häger et al. (2014) nennt sich DT@Scrum. Hier wird Scrum als Rahmen für alle Aktivitäten genutzt, wobei dies auch Designaktivitäten einschließt. Es besteht aus drei verschiedenen Operationsmodi: Design Thinking Mode, Initial Development Mode und Fully

Integrated Mode. Diese Modi zielen darauf ab, den Übergang von DT zur Softwareentwicklung schrittweise zu gestalten.

Weiterführend wurde diese zwei Integrationsansätze verglichen. Während Häger et al. (2014) auf eine umfassende Integration beider Ansätze setzen, betonen Hehn et al. (2022) die Benutzerzentriertheit und kontinuierliche Validierung von Ideen. Genauso wurden Unterschiede der Struktur sowie Schwerpunktsetzung der Frameworks herausgestellt, wobei zusätzlich eine Tabelle die vergleichbaren Rollen der Ansätze präsentiert.

Anschließend wurden die Handlungsempfehlungen basierend auf den Vergleichen zwischen den Ansätzen von Hehn et al. und Häger et al. ausgesprochen. Während Hehn et al. für Projekte mit Schwerpunkt auf Innovation und klaren Strukturen geeignet ist, bietet Häger et al. Flexibilität und eignet sich für Projekte mit bereits etablierten Scrum-Teams. Die Auswahl hängt von den spezifischen Anforderungen und Zielen eines Projekts sowie den vorhandenen Ressourcen ab. Es ist wichtig, die Bedürfnisse des Projekts sorgfältig zu prüfen, bevor eine Methode ausgewählt wird. Mit diesem Abschnitt kann auch das zweite Ziel als erfüllt angesehen werden.

Abschließend wurden noch die Herausforderungen der Integration thematisiert, um das Thema abzurunden. Besonders die Ressourcenzuweisung gestaltet sich schwierig, da DT und Scrum unterschiedliche Anforderungen haben. Außerdem können sich Konflikte auch aus den unterschiedlichen Herangehensweisen und Arten von Problemen ergeben. Eine konstante Kommunikation zwischen Teammitgliedern ist entscheidend, um diese Hindernisse zu überwinden.

5.2 Ausblick

Aufgrund des immer weiter steigenden Trends von DT und agilen Methoden liegt es nahe, dass die Integration von DT in das Scrum-Framework auch in der Zukunft eine präsenste Thematik sein wird, sodass diese wissenschaftliche Ausarbeitung ein hochaktuelles Thema behandelt.

Diese Arbeit bietet eine umfassende Grundlage, um das Thema DT sowie Scrum nachzuvollziehen. Wie im ersten Kapitel der Einleitung bereits erwähnt, wurde an einigen

Stellen auf eine tiefgehende Erläuterung verzichtet, um den Rahmen dieser Arbeit nicht zu sprengen.

Folglich könnten weitere Forschungsarbeiten innerhalb dieses Themenspektrums diese Arbeit als Basis verwenden, sodass bestimmte Teilaspekte dieser vertiefter thematisiert werden können, wie beispielweise die beiden vorgestellten und verglichenen Konzepte zu einem Ansatz zusammenzuführen.

Abschließend ermöglicht die Arbeit ebenfalls von weiteren Forschenden eventuell in Kooperation mit Unternehmen in seinen jeweiligen Kapiteln erweitert zu werden, sodass eine gemeinsam erarbeitete wissenschaftliche Ausarbeitung mit Schwerpunkt auf der Praxis vorliegen könnte. Diese könnte wiederum als Grundlage für die Thematik von verschiedensten Interessenten verwendet werden.

Literaturverzeichnis

AJ&Smart. (2020, 4. Februar). *What is design thinking? An overview* [Video]. YouTube. <https://www.youtube.com/watch?v=gHGN6hs2gZY>

AlleAktien GmbH. (2024, 16. Januar). *Scrum Definition | AlleAktien Investment Research*. AlleAktien Investment Research. Abgerufen am 16. April 2024, von <https://www.alleaktien.de/lexikon/scrum>

Archer, L. B. (1966). *Systematic Method for Designers*.

Auernhammer, J. & Roth, B. (2023). *What is Design Thinking?* <https://doi.org/10.13140/RG.2.2.17861.22243>

Badke-Schaub, P., Roozenbrug, N. & Cardoso, C. (2010). Design thinking: A paradigm on its way from dilution to meaninglessness? *Proceedings Of The 8th Design Thinking Research Symposium (DTRS8)*. https://www.researchgate.net/publication/265403729_Design_thinking_A_paradigm_on_its_way_from_dilution_to_meaninglessness

Barsalou, L. W. (2017). Define design thinking. *She Ji: The Journal Of Design, Economics, And Innovation*, 3(2), 102–105. <https://doi.org/10.1016/j.sheji.2017.10.007>

Blatt, M. & Sauvonnet, E. (2017). *Wo ist das Problem?: Mit Design Thinking Innovationen entwickeln und umsetzen*. Vahlen.

Boland, R. J. & Collopy, F. (2021). 1 Design matters for management. In *Stanford University Press eBooks* (S. 3–18). <https://doi.org/10.1515/9780804767439-003>

Brenner, W. & Uebernickel, F. (2016). Design thinking for innovation. In *Springer eBooks*.
<https://doi.org/10.1007/978-3-319-26100-3>

Brown, T. (2008). Design thinking. *Harvard Business Review*.
<https://readings.design/PDF/Tim%20Brown,%20Design%20Thinking.pdf>

Brown, T. (2009). *Change by design: How Design Thinking Transforms Organizations and Inspires Innovation*. Harper Collins.

Brown, T. (2016). *Change by Design: Wie Design Thinking Organisationen verändert und zu mehr Innovation führt*. Franz Vahlen.

Brown, T. & Katz, B. (2009). Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation. *Journal Of Product Innovation Management*, 28, 381–383. <https://doi.org/10.1111/j.1540-5885.2011.00806.x>

Buchanan, R. (1992). Wicked problems in design thinking. *Design Issues*, 8(2), 5.
<https://doi.org/10.2307/1511637>

Clark, K. & Smith, R. A. (2008). Unleashing the power of design thinking. *Design Management Review*, 19(3), 8–15. <https://doi.org/10.1111/j.1948-7169.2008.tb00123.x>

Curedale, R. (2013). *Design thinking: Process and Methods Manual*. Design Community College.

Dorst, K. (2010). The Nature of Design Thinking. *Proceedings Of The 8th Design Thinking Research Symposium*.
<https://opus.lib.uts.edu.au/bitstream/10453/16590/1/2010000367OK.pdf>

d.school. (o. D.). An Introduction to Design Thinking: PROCESS GUIDE. *Web Stanford*.
Abgerufen am 3. Dezember 2023, von
<https://web.stanford.edu/~mshanks/MichaelShanks/files/509554.pdf>

d.school. (2010). Bootcamp bootleg. *Hasso Plattner Institute Of Design At Stanford*.
https://hpi.de/fileadmin/user_upload/fachgebiete/d-school/documents/01_GDTW-Files/bootcampbootleg2010.pdf

- Eppler, M. & Hoffmann, F. (2012). *Organisations Entwicklung 2: Design Thinking im Management*. https://www.zu-daily.de/daily-wAssets/pdf/2012-09-30_DT_OrganisationsEntwicklung.pdf
- Gebbing, P., Yang, X., Michalke, S. & Lattemann, C. (2021). Kreativitätsförderung in der virtuellen Gruppenarbeit. *HMD Praxis der Wirtschaftsinformatik*, 58(6), 1364–1377. <https://doi.org/10.1365/s40702-021-00800-5>
- Gehm, J. (2022). *Design thinking etablieren: So verankern Sie Design Thinking im Unternehmen – bewährtes Vorgehensmodell, Erfahrungen, Fallbeispiele*. Springer Gabler.
- Gerstbach, I. (2016). *Design Thinking im Unternehmen: Ein Workbook für die Einführung von Design Thinking*. GABAL Verlag GmbH.
- Gerstbach, I. & Gerstbach, P. (2020). Was ist Design Thinking? In *Design Thinking in IT-Projekten*. <https://doi.org/10.3139/9783446460744.002>
- Gloger, B. (2016). *Scrum: Produkte zuverlässig und schnell entwickeln*. Carl Hanser Verlag GmbH Co KG.
- Grots, A. & Pratschke, M. (2009). Design Thinking — Kreativität als Methode. *Marketing Review St. Gallen*, 26(2), 18–23. <https://doi.org/10.1007/s11621-009-0027-4>
- Häger, F., Kowark, T., Krüger, J., Vetterli, C. & Uflacker, M. (2014). DT@Scrum: Integrating Design Thinking with Software Development Processes. In *Design thinking research: Building Innovators*. Springer.
- Häger, F., Vetterli, C., Uebernicker, F., Brenner, W., Kowark, T., Krueger, J., Mueller, J., Plattner, H., Stortz, B. & Sikkha, V. (2013). *Jumpstarting Scrum with Design Thinking*. https://www.researchgate.net/profile/Christophe-Vetterli/publication/255710860_Jumpstarting_Scrum_with_Design_Thinking/links/02e7e520529e022364000000/Jumpstarting-Scrum-with-Design-Thinking.pdf
- Hassi, L. & Laakso, M. (2011). *Conceptions of Design Thinking in the Management Discourse*. https://www.researchgate.net/publication/274070930_Conceptions_of_Design_Thinking_in_the_Management_Discourse

Hehn, J., Mendez, D., Brenner, W. & Broy, M. (2022). *Design Thinking for Software Engineering: Creating Human-oriented Software-intensive Products and Services*. Springer Nature.

Holcomb, S. & Segovia, K. (2019, 10. Juni). How to Inspire Upper Management with Empathetic Data. *Stanford d.school*. Abgerufen am 3. Dezember 2023, von <https://dschool.stanford.edu/news-events/empathetic-data>

IDEO.org. (o. D.). *Design Kit*. Design Kit. Abgerufen am 7. März 2024, von <https://www.designkit.org/methods.html>

Interaction Design Foundation. (2016, 4. Juni). *What are Wicked Problems?* The Interaction Design Foundation. Abgerufen am 15. März 2024, von <https://www.interaction-design.org/literature/topics/wicked-problems>

Jacoby, R. J. & Rodríguez, D. V. (2007). Innovation, growth, and getting to where you want to go. *Design Management Review*, 18(1), 10–15. <https://doi.org/10.1111/j.1948-7169.2007.tb00067.x>

Johansson, U. & Woodilla, J. (2009). TOWARDS AN EPISTEMOLOGICAL MERGER OF DESIGN THINKING, STRATEGY AND INNOVATION. *8th European Academy Of Design Conference 2009 Aberdeen, 1–5*. http://www.designfakulteten.kth.se/sites/default/files/1.44194TOWARDS_AN_EPISTEMOLOGICAL_MERGER_OF_DESIGN_THINKING_STRATEGY_AND_INNOVATION_JW.pdf

Kaiser, F. & Van Bennekum, A. (2022). *Scrum? Frag doch einfach!: Klare Antworten aus erster Hand*. UVK Verlag.

Kaufman, J. C. & Sternberg, R. J. (2019). Enhancing people's creativity. In *The Cambridge Handbook of Creativity* (2. Aufl., S. 88–104). Cambridge University Press.

Kelley, D. & Kelley, T. (2013). *Creative Confidence: Unleashing the creative potential within us all*. HarperCollins UK.

Kleuker, S. (2018). *Grundkurs Software-Engineering mit UML: Der pragmatische Weg zu erfolgreichen Softwareprojekten*. Springer-Verlag.

Kuster, J., Huber, E., Lippmann, R., Schmid, A., Schneider, E., Witschi, U. & Wüst, R. (2011). *Handbuch Projektmanagement* (3. Aufl.). Springer-Verlag.

Lenz, M. (2021, 5. Februar). *Agiles Projektmanagement mit Scrum - Guide*. EXPERTE.de. Abgerufen am 16. April 2024, von <https://www.experte.de/projektmanagement/scrum>

Lewrick, M., Link, P. & Leifer, L. (2019). *Das Design Thinking Toolkit: Die besten Werkzeuge & Methoden*. Vahlen.

Liedtka, J. & Ogilvie, T. (2011). *Designing for Growth: A Design Thinking Tool Kit for Managers*. Columbia University Press.

Lindberg, T., Gumienny, R., Jobst, B. & Meinel, C. (2010). Is There a Need for a Design Thinking Process? *Proceedings Of Design Thinking Research Symposium*, 8(Design 2010), 243–254.

https://hpi.de/fileadmin/user_upload/fachgebiete/meinel/papers/Design_Thinking/2010_Lindberg_Design.pdf

Linke, L. (2020, 15. Oktober). Kreatives Denken im Design Thinking: Divergieren und Konvergieren. *WEKA - E-Learning mit Profis*. Abgerufen am 10. Januar 2024, von <https://magazin.weka-elearning.de/design-thinking/divergieren-und-konvergieren/#:~:text=Konvergierendes%20Denken,->

[Wenn%20wir%20konvergierend&text=Wir%20verwenden%20Regeln%20und%20Gesetz%20m%C3%A4%C3%9Figkeiten,Blick%20auf%20die%20Realit%C3%A4t%20angeordnet](https://magazin.weka-elearning.de/design-thinking/divergieren-und-konvergieren/#:~:text=Konvergierendes%20Denken,-Wenn%20wir%20konvergierend&text=Wir%20verwenden%20Regeln%20und%20Gesetz%20m%C3%A4%C3%9Figkeiten,Blick%20auf%20die%20Realit%C3%A4t%20angeordnet).

Malone, T. W., Laubacher, R. & Dellarocas, C. (2009). Harnessing Crowds: Mapping the Genome of Collective Intelligence. *Social Science Research Network*. <https://doi.org/10.2139/ssrn.1381502>

Martin, R. L. (2009). *The Design of Business: Why design thinking is the next competitive advantage*. https://en.wikipedia.org/wiki/The_Design_of_Business

- Martin, R. L. (2010). Design thinking: achieving insights via the “knowledge funnel”. *Strategy & Leadership*, 38(2), 37–41. <https://doi.org/10.1108/10878571011029046>
- Michlewski, K. (2008). Uncovering design attitude: inside the culture of designers. *Organization Studies*, 29(3), 373–392. <https://doi.org/10.1177/0170840607088019>
- Miller, K. & Moultrie, J. (2013). Understanding the skills of design leaders. *Design Management Journal*, 8(1), 35–51. <https://doi.org/10.1111/dmj.12002>
- Mueller-Roterberg, C. (2018). *Handbook of Design Thinking: Tips & Tools for How to Design Thinking*. Independently Published.
- Naiman, L. (2019, 20. Mai). *Design Thinking as a Strategy for Innovation*. The European Business Review. Abgerufen am 15. April 2024, von <https://www.europeanbusinessreview.com/design-thinking-as-a-strategy-for-innovation/>
- Oxford Advanced Learner’s Dictionary. (o. D.). *design noun - Definition, pictures, pronunciation and usage notes*. Abgerufen am 16. April 2024, von https://www.oxfordlearnersdictionaries.com/definition/english/design_1?q=design
- Plattner, H., Meinel, C. & Leifer, L. (2012). *Design thinking research: Studying Co-Creation in Practice*. Springer Science & Business Media.
- Plattner, H., Meinel, C. & Weinberg, U. (2009). *Design thinking: Innovation lernen, Ideenwelten öffnen*.
- Satpathy, T. (2016). *Scrum Body of Knowledge (SBOK guide)* (3. Aufl., Nummer 3). <https://servicecontrolling.org/wp-content/uploads/2020/02/SCRUM-Study-SBOK-Guide-German.pdf>
- Schallmo, D. R. & Lang, K. (2020). *Design Thinking erfolgreich anwenden: So entwickeln Sie in 7 Phasen kundenorientierte Produkte und Dienstleistungen*. Springer-Verlag.
- Schwaber, K. & Sutherland, J. (2020). *Der Scrum Guide: Der gültige Leitfaden für Scrum: Die Spielregeln*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-German.pdf>
- Schweitzer, J., Groeger, L. & Sobel, L. (2016). The design thinking mindset: an assessment of what we know and what we see in practice. *ResearchGate*. <https://doi.org/10.1386/dbs.2.1.71>

Simschek, R. & Kaiser, F. (2019a). *Design thinking: Innovation erfolgreich umsetzen*. UVK Verlag.

Simschek, R. & Kaiser, F. (2019b). *SCRUM: Das Erfolgsphänomen einfach erklärt*. UVK Verlag.

The Interaction Design Foundation. (2016, 26. November). *What is Lateral Thinking?* Abgerufen am 24. März 2024, von [https://www.interaction-design.org/literature/topics/lateral-thinking#:~:text=Lateral%20thinking%20\(horizontal%20thinking\)%20is,to%20help%20find%20innovative%20solutions.](https://www.interaction-design.org/literature/topics/lateral-thinking#:~:text=Lateral%20thinking%20(horizontal%20thinking)%20is,to%20help%20find%20innovative%20solutions.)

Tiba. (2023, 15. Januar). *Die 5 beliebtesten agilen Methoden, Ansätze & Prozesse*. Tiba Transformation Group. Abgerufen am 6. April 2024, von <https://www.tiba.de/die-5-beliebtesten-agilen-methoden-tbs/>

Uebernicket, F., Jiang, L., Brenner, W., Pukall, B., Naef, T. & Schindlholzer, B. (2020). *Design Thinking: the Handbook*. World Scientific.

unfcc. (2022). *Design Thinking Methodology* [Datensatz]. <https://unfcc.int/sites/default/files/resource/Design%20Thinking%20Methodology.pdf>

Wieczorrek, H. W. & Mertens, P. (2010). *Management von IT-Projekten: Von der Planung zur Realisierung* (4. Aufl.). Springer-Verlag.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original