



universität
wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

Product-led Growth (PLG): Vom Konzept zur Instanz

„Die Entwicklung moderner, softwarebasierter Angebote auf PLG Basis
unter Anwendung eines metamodellbasierten Vorgehensmodell“

verfasst von / submitted by

Iris Viktoria Edlinger, BSc (WU)

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2023 / Vienna 2023

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

UA 066 915

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Masterstudium Betriebswirtschaft

Betreut von / Supervisor:

o. Univ.- Prof. Dr. techn. Dimitris Karagiannis

Danksagung

Hiermit möchte ich mich bei allen Personen bedanken, die mich im Laufe meines Studiums unterstützt haben.

Ein besonderer Dank gilt Herrn Dr. Christian Lichka, der mich beim Verfassen der vorliegenden Arbeit mit seinen fachlichen Kenntnissen und Betreuung unterstützt hat und mir mit seiner stets motivierenden Art maßgeblich zu dem positiven Ergebnis verholfen hat. Ebenso möchte ich mich herzlich bei o. Univ.- Prof. Prof.h.c. Dr. Dimitris Karagiannis für seine Betreuung bedanken sowie auch für die Möglichkeit, die Masterarbeit am Institut für Wirtschaftsinformatik und in Kooperation mit der BOC Group verfasst haben zu dürfen. Dadurch konnte ich nicht nur in ein äußerst interessantes und zukunftssträchtiges Thema eintauchen, sondern durfte auch tiefe Einblicke in ein spannendes Unternehmen gewinnen. An dieser Stelle möchte ich auch den beiden Produktmanagern des Software-Produkts ADOGRC der BOC Group, Erik Guschlbauer und Michael Höhr meinen Dank aussprechen.

Zuletzt möchte ich mich bei meiner Familie, Freunden und meinem Freund bedanken, die mir während dem Studium immer den Rücken gestärkt haben.

Kurzfassung

Der Einzug fortschreitender Technologie in alle Aspekte des Alltags geht mit einer steigenden digitalen Kompetenz einher und zeigt sich in einem radikalen Wandel der Art und Weise wie Produkte und Dienstleistungen von Menschen gekauft und genutzt werden. So zeigt sich die zunehmende Technikaffinität seitens Käufer:innen und Nutzer:innen in einer Abneigung gegen traditionelle Verkaufsprozesse mit persönlichen Interaktionen und hin in Richtung eines im Kontext von Software digitalen, self-service basierten, kontinuierlichen Überzeugungs- und Nutzungsprozesses.

Um am Markt bestehen zu bleiben, sind Unternehmen mehr denn je gefordert auf das sich verändernde Umfeld rasch zu reagieren und ihre Ausrichtung kontinuierlich zu evaluieren und gegebenenfalls anzupassen.

Mit der Herausforderung diesem aktuellen Wandel Schritt zu halten, sind in der Literatur diverse Konzepte zu finden. Einige davon lassen sich unter dem Konzept Product-led Growth zusammenfassen, welches insbesondere bei Unternehmen aus dem Softwarebereich zunehmend in den Fokus rückt und Schwerpunkt der vorliegenden Arbeit ist. Trotz zahlreicher weiterer bestehender Konzepte, die sich grundsätzlich mit der gleichen Thematik beschäftigen, gibt es bislang kein umfassendes Modell zu einem konkreten Vorgehen.

Ziel dieser Arbeit ist es daher ein Vorgehensmodell auf Basis von Product-led Growth für moderne Software-Angebote zu entwickeln. Ebenso soll es sich dabei um ein metamodellbasiertes Vorgehensmodell handeln, um eine einfachere Instanziierung von Product-led Growth für Software-Angebote zu ermöglichen. Die Schwerpunkte dieser Arbeit liegen auf den drei Schlagwörtern metamodellbasiertes Vorgehensmodell, Product-led Growth und Software-Angebote.

Abstract

The arrival of advancing technology in all aspects of everyday life is accompanied by increasing digital literacy and is reflected in a radical change in the way products and services are bought and used by people. Thus, the increasing affinity of buyers and users for technology is reflected in an aversion to traditional sales processes with face-to-face interactions and moves toward a digital, self-service-based, continuous persuasion and usage process in the context of software.

In order to survive in the market, companies are required more than ever to react quickly to the changing environment and to continuously evaluate and, if necessary, adapt their orientation.

With the challenge to keep up with this current change, various concepts can be found in the literature. Some of these can be summarized under the concept of product-led growth, which is increasingly becoming the focus of attention, particularly among companies in the software sector, and is also the focus of this thesis. Despite numerous other existing concepts that basically deal with the same topic, there is no comprehensive model for a concrete approach so far.

The goal of this work is to develop a procedure model based on product-led growth for modern software offerings. Beyond that it is to concern a metamodel-based procedure model, in order to make a simpler instantiation possible of product-led growth for software offerings. The emphasis is on the three key words metamodel-based procedure model, product-led growth and software offerings.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Motivation.....	1
1.2	Problemstellung und Zielsetzung	2
1.3	Relevanz und Forschungsfrage.....	3
1.4	Systematische Literaturanalyse.....	3
1.4.1	Relevante Ergebnisse der Literaturrecherche.....	8
1.4.2	Schlussfolgerungen aus der Literaturrecherche	9
1.5	Struktur der Arbeit	10
2	Theoretische Grundlagen	12
2.1	Product-led Growth	12
2.2	Modellierung und Metamodellierung	15
2.2.1	Der Modell-Begriff.....	15
2.2.2	Rahmenwerk der Modellierung.....	16
2.2.3	Der Metamodell-Begriff.....	18
3	Methodik.....	21
3.1	Modellerstellung	21
3.1.1	Einführung in CoChaCo.....	21
3.1.2	Modellerstellung mit CoChaCo	24
3.2	Erläuterung der Szenarien-Evaluierung	27
4	Ergebnisse	29
4.1	Das metamodellbasierte Vorgehensmodell.....	29
4.2	Erweiterung des Modells um phasenspezifische Zielsetzungen	34
4.3	Erweiterung des Modells um ausgewählte Konzepte	41
4.3.1	Lean Startup	42
4.3.2	Business Model Canvas	45
4.3.3	Value Proposition Canvas	47
4.3.4	Business Model Freemium	48
4.3.5	Personas.....	49
4.3.6	Customer Journey Mapping.....	49
4.3.7	User Experience Design	51

4.3.8	User Interface Design	53
4.3.9	Minimum Viable Product.....	54
4.3.10	Cloud-Native	54
4.3.11	Microservices	55
4.3.12	Agile Softwareentwicklung	55
4.3.13	DevOps	56
4.3.14	Voice of the Customer.....	56
4.3.15	Übersicht der Konzepte.....	57
5	Evaluierung	60
5.1	Define.....	60
5.2	Design	64
5.3	Build	66
5.4	Deploy and Scale	67
5.5	Engage and Extend.....	68
5.6	Understand and Measure/Learn and Adopt	69
6	Fazit.....	71
7	Literaturverzeichnis	73
8	Abbildungsverzeichnis.....	78
9	Tabellenverzeichnis.....	80
10	Abkürzungsverzeichnis.....	81
11	Anhang	82

1 Einleitung

Im einleitenden Kapitel wird die zugrundeliegende Motivation dieser Arbeit dargelegt, die ermittelte Problemstellung des behandelten Themenbereichs erläutert sowie auch die sich daraus ergebende Zielsetzung beschrieben. Ebenso wird auf die Relevanz des Themas sowie auch auf die Forschungsfragen näher eingegangen und die im Rahmen dieser Arbeit durchgeführte systematische Literaturanalyse wiedergegeben. Abschließend wird die Struktur der Arbeit aufgezeigt, um dem Leser oder der Leserin einen Überblick über die vorkommenden Kapitel zu geben.

1.1 Motivation

Der Begriff „Digitale Transformation“ hat in den letzten Jahren rasch an Bedeutung gewonnen und sich zu einem wichtigen Phänomen in der Forschung sowie auch innerhalb der Unternehmenswelt etabliert. Denn der Einzug fortschreitender Technologie in alle Aspekte des Alltags geht mit einer steigenden digitalen Kompetenz einher und zeigt sich in einem radikalen Wandel der Art und Weise wie Produkte und Dienstleistungen von Menschen gekauft und genutzt werden (Vial, 2019, S. 120–121).

So zeigt sich die zunehmende Technikaffinität seitens Käufer:innen und Nutzer:innen in einer Abneigung gegen traditionelle Verkaufsprozesse mit persönlichen Interaktionen. Anstatt auf Unternehmen angewiesen zu sein, wollen sich Käufer:innen und insbesondere End-Nutzer:innen selber über deren Produkte und Dienstleistungen informieren, diese erforschen und deren Nutzen für sich entdecken. Folglich gelten Einheitslösungen, entwickelt für die breite Masse am Markt, als nicht zufriedenstellend und eine umfangreiche Palette an Funktionen nicht mehr als Alleinstellungsmerkmal. Stattdessen ist die Kundschaft sowie End-Nutzer:innen vielmehr auf der Suche nach mehrwertstiftenden Lösungen, abgestimmt auf ihre speziellen Bedürfnisse und trachten insgesamt nach einzigartigen Produkt- und Nutzungserlebnissen (Yang et al., 2019, S. 173).

Um am Markt bestehen zu bleiben, sind Unternehmen mehr denn je gefordert auf das sich verändernde Umfeld rasch zu reagieren und ihre Ausrichtung kontinuierlich zu evaluieren und gegebenenfalls anzupassen (Vial, 2019, S. 118).

Ein Konzept, das insbesondere bei Unternehmen aus dem Softwarebereich immer stärker in den Fokus rückt und zunehmend Beachtung findet, wird als *Product-led Growth* (deutsch: produktgesteuertes Wachstum) bezeichnet. Dabei handelt es sich um eine *Go-To-Market* Strategie (deutsch: Markteinführungsstrategie), bei der das Produkt als das zentrale Steuerungsinstrument von Wachstum angesehen wird und gleichzeitig als Interaktionsmedium für alle Bedürfnisse von Kunden oder Kundinnen und Nutzer:innen dient. Folglich soll die Akquise, die Erweiterung, die Umwandlung

sowie auch die Bindung von Nutzern oder Nutzerinnen vorrangig durch das Produkt selbst erfolgen. Im Vordergrund dieser Markteinführungsstrategie steht die Generierung eines kurzen *Time-to-Value* (Zeit die es benötigt, bis der oder die Nutzer:in den Mehrwert des Produkts realisiert) (Shein, 2022). Product-led Growth findet bereits bei namhaften Softwareunternehmen wie Clickup, IBM oder auch Airtable Anwendung (NewsRX LLC, 2019).

Nachdem nun die zugrundeliegende Motivation dieser Arbeit aufgezeigt wurde, wird in Abschnitt 1.2 auf die Problemstellung, die sich Unternehmen bei dem Entschluss nach Product-led Growth vorzugeben ergibt, eingegangen. Des Weiteren wird die sich daraus ergebende Zielsetzung erläutert.

1.2 Problemstellung und Zielsetzung

Product-led Growth stellt ein sehr junges und insbesondere äußerst facettenreiches Thema dar, welches sowohl fachliche sowie auch softwarebezogene Aspekte aufweist. Dabei ergibt sich die Problemstellung der Arbeit aus der Notwendigkeit eines instanzierbaren Modells, welches diese Aspekte berücksichtigt und gleichzeitig ein konkretes Vorgehen für Unternehmen bei der Entwicklung moderner, softwarebasierter Angebote auf Product-led Growth Basis festlegt.

Als Zielsetzung der Arbeit gilt daher die Entwicklung eines metamodellbasierten Vorgehensmodells auf Basis von Product-led Growth, welches einen *Blueprint* (deutsch: Bauplan) im Sinne eines Konzepts darstellt, aus dem ein konkretes Vorgehen (Ergebnis 1), phasenspezifische Zielvorgaben (Ergebnis 2) sowie auch ausgewählte exemplarische Konzepte (Ergebnis 3) ablesbar sind. Die Anwendung des Konzepts der Metamodellierung (siehe Abschnitt 2.2) würde eine einfachere Instanziierung von Product-led Growth für (metamodellbasierte) Software-Angebote ermöglichen.

Die zuvor genannten angestrebten Ergebnisse sind in der folgenden Tabelle 1 nochmals aufgeschlüsselt.

Ergebnis 1	Die Entwicklung eines metamodellbasierten Vorgehensmodells auf Basis von Product-led Growth.
Ergebnis 2	Die Erweiterung des metamodellbasierten Vorgehensmodells um spezifische Zielsetzungen der einzelnen Phasen.
Ergebnis 3	Die Erweiterung von Ergebnis 2 um ausgewählte exemplarische Konzepte.

Tabelle 1: Im Rahmen dieser Arbeit angestrebte Ergebnisse

Der folgende Abschnitt 1.3 behandelt die Relevanz des Themas und gibt die konkreten Forschungsfragen dieser Arbeit wieder.

1.3 Relevanz und Forschungsfrage

Die Relevanz des Themas ergibt sich aus der Notwendigkeit für Unternehmen dem aktuellen Trend hinsichtlich der Änderung im Nutzer:innenverhalten zu folgen und eine Anpassung ihrer Ausrichtung in Richtung eines produktbezogenen Wachstums vorzunehmen.

Im Rahmen der vorliegenden Arbeit wurden drei Forschungsfragen aufgestellt (Tabelle 2), die es zu beantworten gilt.

Nr.	Forschungsfrage
1	Wie lässt sich Product-led Growth als ein metamodellbasiertes Konzept umsetzen?
2	Welche bestehenden fachlichen und technischen Konzepte gibt es in der Literatur und Praxis, die mit Product-led Growth zusammenspielen?
3	Inwiefern lässt sich das Metamodell eines Software-Produkts durch Product-led Growth als metamodellbasiertes Konzept erweitern?

Tabelle 2: Aufgestellte Forschungsfragen

Um einen Einblick in den aktuellen Forschungsstand zu dem behandelten Thema zu erhalten sowie etwaige weitere Forschungslücken aufzudecken, wurde eine systematische Literaturanalyse durchgeführt. Diese ist im folgenden Abschnitt 1.4 einzusehen.

1.4 Systematische Literaturanalyse

Um einen Einblick in den aktuellen Forschungsstand zu dem Thema Product-led Growth im Softwarebereich zu erhalten, etwaige Forschungslücken aufzuzeigen sowie das Alleinstellungsmerkmal dieser Arbeit hervorzuheben, wurde auf die Methodik der systematischen Literaturanalyse zurückgegriffen.

Bei der systematischen Literaturrecherche wird themenrelevante Literatur zunächst identifiziert, bewertet und aufgearbeitet, um anschließend die aufgestellten forschungsrelevanten Fragestellungen beantworten zu können. Die systematische Literaturanalyse dient in erster Linie dazu, einen Überblick über den aktuellen Forschungsstand zu erhalten sowie etwaige Forschungslücken herauszufinden. Sie gilt als Grundlage guten wissenschaftlichen Arbeitens, da durch die transparente, replizierbare und systematische Vorgehensweise eine verlässliche Wissensbasis geschaffen wird (Sinzig, 2017, S. 72).

Die gewählte Vorgehensweise hat sich an dem Literaturanalyseprozess nach Fink (2005, S. 3) orientiert und ist in Abbildung 1 dargestellt.

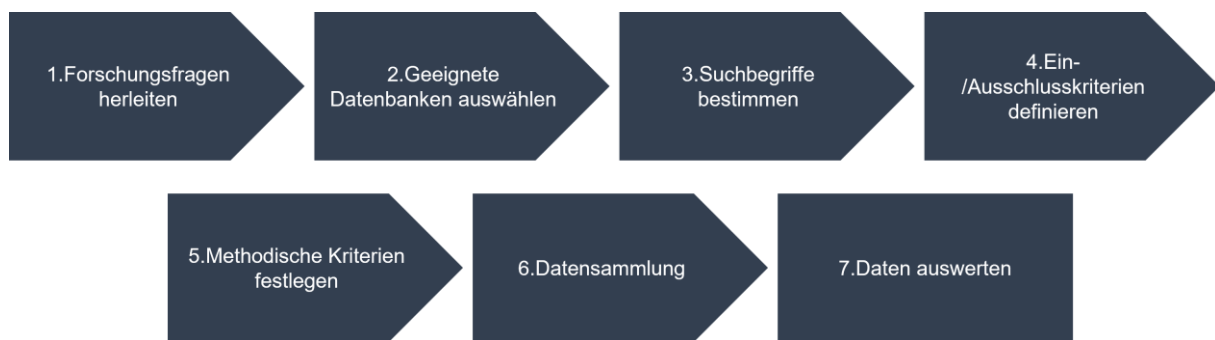


Abbildung 1: Vorgehen bei einer systematischen Literaturanalyse (in Anlehnung an (Fink, 2005, S. 3))

Wie in dem Prozess ersichtlich, wurden auch in der vorliegenden Arbeit in einem ersten Schritt (siehe Abbildung 1/Schritt 1) forschungsleitende Fragen formuliert, welche eine erste Strukturierung der Suche vorgeben sowie auch bei der Auswertung der Literatur unterstützend wirken (Sinzig, 2017, S. 69). Diese forschungsleitenden Fragen wurden bereits im vorherigen Abschnitt 1.3 vorgestellt und sind in der Tabelle 2 zusammengefasst.

Nachdem die Forschungsfragen definiert wurden, wurden in einem nächsten Schritt geeignete Datenbanken für die Literatursuche ausgewählt, Suchbegriffe bestimmt sowie Ein- und Ausschlusskriterien definiert (siehe Abbildung 1/ Schritte 2-4) (Sinzig, 2017, S. 72).

In der vorliegenden Arbeit wurde neben den beiden bekannten Datenbanken ACM Digital Library und Emerald Insight, die Online-Bibliothek U:search der Universität Wien herangezogen, da diese freien Zugang zu einer großen Sammlung veröffentlichter Literatur bietet. Darüber hinaus wurde die Open-Access-Datenbank Google Scholar genutzt, da in dieser ebenfalls eine Menge an Literatur zu finden ist und darüber hinaus einen Vollzugriff ermöglicht (Xiao & Watson, 2019, S. 103). ACM Digital Library und Emerald Insight wurden in erster Linie aufgrund des zu dieser Arbeit passenden Datenbanktyps ausgewählt und sind in der folgenden Tabelle 3 nochmals einsehbar.

Datenbank	Ort/Verlag	Datenbanktyp	Sprache
Emerald Insight	Emerald Publishing Limited	• Informatik, E-Journals/E-Books • Informations- und Bibliothekswissenschaft, E-Journals/E-Books • Wirtschaft, E-Journals/E-Books	Englisch
ACM Digital Library	Association for Computing Machinery (ACM)	• Informatik – E-Journals/E-Books	Englisch

Tabelle 3: Für die Literaturrecherche gewählte Datenbanken

Nachdem die Datenbanken ausgewählt wurden, galt es in einem weiteren Schritt passende Suchbegriffe zu definieren, um auf themenadäquate Literatur zu stoßen. Die zuvor festgelegten Forschungsfragen bieten dabei eine erste Orientierung und geben vor, auf welche Begriffe bei der Recherche zurückgegriffen werden sollte (Brink, 2013, S. 54). Die Begriffe der Suchanfrage wurden daraufhin um Synonyme, alternative Schreibweisen, Abkürzungen sowie verwandte Begriffe erweitert, mit dem Ziel, ein möglichst umfängliches Ergebnis zu erhalten (Xiao & Watson, 2019, S. 104).

Die folgende Tabelle 4 beinhaltet die für die Literaturrecherche relevanten Suchbegriffe, deren Synonyme sowie auch englische Übersetzungen für die Eingabe in die Datenbanken.

Nr.	Suchbegriff	Kombinierter Suchbegriff (DE)	Kombinierter Suchbegriff (ENG)
1	Software	Software OR digitales Produkt	software OR digital product
2	Product-Led	Produktbasiert OR produktgesteuert OR produktorientiert	product led OR product oriented OR product based
3	Wachstum	Wachstum	growth
4	Metamodell	Modell OR Metamodell OR Metamodellierung	model OR metamodel OR metamodeling
5	Strategie/Taktik	Strategie OR Taktik	strategy OR tactic
6	Konzept	Konzept OR Ansatz	concept OR approach

Tabelle 4: Für die Literaturrecherche definierte Suchbegriffe

Nachdem die Suchbegriffe festgelegt wurden, ist es daraufhin erforderlich diese miteinander zu kombinieren, um möglichst relevante Suchergebnisse zu erhalten. Dabei haben sich drei Kombinationen als sinnvoll erschlossen.

Der erste kombinierte Suchbegriff ergibt sich aus den Termen „Software“, „Strategie“, „Konzept“ und „Wachstum“, mit dem Ziel der Auflistung aktueller Strategien und Konzepte zu Wachstumsmöglichkeiten im Softwarebereich.

Die zweite Schwerpunktsuche ist eine Kombination aus den Begriffen „Software“, „Wachstum“ und „Metamodell“. Hier gilt es herauszufinden, ob bereits etwaige Metamodelle zum Thema Wachstum im Softwarebereich existieren.

Die dritte Kombination, bestehend aus „Product-Led“ und „Metamodell“ stellt die spezifischste Schwerpunktsuche dar, durch welche es herauszufinden gilt, ob es in der Literatur bereits ein Metamodell zu dem Product-Led Growth Ansatz gibt.

Bei der Suche wurden die gebräuchlichsten Booleschen Operatoren *OR* (deutsch: ODER) und *AND* (deutsch: UND) genutzt. In der Regel dient *AND* zur Verknüpfung von Hauptbegriffen und *OR* zur Erweiterung durch Synonyme (Xiao & Watson, 2019, S. 104). Die Kombination der Suchbegriffe ist in der folgenden Tabelle 5 dargestellt.

Begriff	Kombination Suchbegriffe	Suche (ENG)
Software Strategie Konzept Wachstum	Software AND Strategie AND Konzept AND Wachstum	(software OR digital product) AND (strategy OR tactic) AND (concept OR approach) AND (growth)
Software Metamodell Wachstum	Software AND Metamodell AND Wachstum	(software OR digital product) AND (model OR metamodel OR metamodeling) AND (growth)
Product-Led Metamodell	Product-Led AND Metamodell	(product led OR product oriented OR product based) AND (model OR metamodel OR metamodeling)

Tabelle 5: Kombination der Suchbegriffe

Wie der in Abbildung 1 dargestellte Prozess vorgibt, müssen in einem nächsten Schritt Ein- und Ausschlusskriterien definiert werden. Um die Aktualität der Literatur zu wahren wurde bei der Suche eine zeitliche Begrenzung von fünf Jahren vorgenommen, bei welcher folglich nur Publikationen Berücksichtigung fanden, die zwischen den Jahren 2017 und 2021 veröffentlicht wurden. Da die zuvor gelisteten Datenbanken hauptsächlich englische Publikationen beinhalten, ist hervorzuheben, dass auf Grund dessen auch nur nach den englischen kombinierten Suchbegriffen geforscht wurde. Die Publikationsart beschränkte sich hauptsächlich auf Artikel aus Fachzeitschriften, da diese ein intensives Begutachtungsverfahren durch Experten des jeweiligen Fachgebiets durchlaufen (sog. *Peer Reviews*). Darüber hinaus wurden Konferenzbeiträge eingebunden (Bergener et al., 2019, S. 42). In der folgenden Tabelle 6 sind die Filter- sowie Auswahlkriterien aufgeschlüsselt dargestellt.

Nr.	Filter-/Auswahlkriterium	Art
1	Jahr	Ab 2017
2	Sprache	Englisch
3	Publikationsart	Artikel aus Fachzeitschriften Konferenzbeiträge
4	Zugang	Freier Zugang

Tabelle 6: Filter- und Auswahlkriterien der Literaturanalyse

In der nachfolgenden Tabelle 7 ist die Anzahl der Suchergebnisse für die jeweiligen kombinierten Suchbegriffe in den ausgewählten Datenbanken ersichtlich. Wie zu sehen, wurden bei der Online-Bibliothek U:search die meisten Suchergebnisse generiert. Grund dafür ist der Umfang an Publikationen aus mehreren verschiedenen Datenbanken, die in der Online-Bibliothek zugänglich sind.

Name	Emerald Insight	ACM	U:search
Software, Strategie, Konzept, Wachstum	14.000	3.857	342.761
Software, Metamodell, Wachstum	16.000	8.507	741.096
Product-Led, Metamodell	2.000	501	30.150

Tabelle 7: Anzahl Suchergebnisse in Suchmaschinen

Nach der Suche galt es aus der Summe an Suchergebnissen die relevantesten Publikationen auszuwählen. Im Zuge dessen, fand eine erste Auswahl nach Titel und *Abstract* (deutsch: Kurzfassung) statt (siehe Abbildung 1/ Schritt 5). Nach dieser ersten Auswahlrunde erfolgte anschließend eine Begutachtung des Volltextes, bei welcher die Qualität und Relevanz der Arbeiten geprüft und folglich als nicht relevant empfundene Publikationen ausgeschlossen werden konnten (Tabelle 8). Die verbleibenden wissenschaftlichen Arbeiten wurden in einer Ergebnisliste zusammengetragen und historisiert.

Datenbank	Ergebnisse nach Sichtung von Titel und Abstract	Anzahl Ergebnisse nach Sichtung der Volltexte
ACM Digital Library	48	16
Emerald Insight	58	11
U:Search	19	3

Tabelle 8: Anzahl der Ergebnisse nach Sichtung der Publikationen

Die im Rahmen dieser Arbeit durchgeführte systematische Literaturrecherche hat gezeigt, dass es grundsätzlich keine allgemeingültige Vorgehensweise zum Thema Wachstum im Softwarebereich gibt, die Unternehmen direkt auf ihr Produkt oder Dienstleistung umwälzen und anwenden können. Stattdessen wird in der Literatur auf bestimmte Konzepte verwiesen, die insbesondere bei der Entwicklung von Softwareprodukten für den Erfolg von Unternehmen eine Rolle spielen. Hervorzuheben ist, dass die unterschiedlichen Konzepte auch in unterschiedlichen Publikationen behandelt werden und somit keine wissenschaftlichen Studien existieren, die alle relevanten Wachstumskonzepte zusammengefasst beinhalten. In dieser Arbeit wird im Speziellen auf Literatur zurückgegriffen, die Parallelitäten zu Product-led Growth hinsichtlich relevanter Konzepte aufweisen. Im Folgenden wird auf zwei Publikationen näher eingegangen, die Praktiken im Hinblick auf den Software-Entwicklungsprozess von Software-Startups hervorheben.

1.4.1 Relevante Ergebnisse der Literaturrecherche

Souza et al. (2019) präsentieren in ihrer wissenschaftlichen Arbeit eine Fallstudie über Software-Startups, um Praktiken in Hinblick auf den Software-Entwicklungsprozess besser zu verstehen und haben zu diesem Zweck qualitative Interviews mit zehn Software-Startups durchgeführt. Resultat ihrer Publikation ist das sogenannte *Industrial Startup Model* (deutsch: industrielles Startup Modell), welches insgesamt neun Kernkategorien umfasst. Für die vorliegende Arbeit sind insbesondere die folgenden fünf Kernkategorien von Interesse.

- CAT 1: Speed-up Development
- CAT 2: Customer-driven Development
- CAT 3: Innovation-driven Development
- CAT 8: Evolutionary Approach
- CAT 9: User Experience has High Priority

Erstere Kategorie (CAT 1) bezieht sich darauf, dass sich Startups speziell auf schnelles Wachstum und Skalierbarkeit fokussieren und infolgedessen auf eine kontinuierliche Integration und Bereitstellung achten. Zu diesem Zweck kommen bei diesen Unternehmen vor allem agile Praktiken zur Anwendung. Ebenfalls konnte in der Studie herausgefunden werden, dass Startups laufend Prototypen entwickeln und diesen einer permanenten Validierung unterziehen, mit dem Ziel der Kundschaft das erwartete oder gewünschte Produkt zu liefern oder andernfalls das Geschäftsmodell anzupassen. Die Kundschaft wird in diesem Szenario folglich als das „Warum“ für alles betrachtet (CAT 2). CAT 3 betont die Schaffung von Wert für welchen die Kundschaft gewillt ist zu zahlen. In der Kategorie Evolutionary Approach (CAT 8) wird beschrieben, dass Startups insbesondere auf kontinuierliches Feedback seitens der Kundschaft angewiesen sind, um gegebenenfalls ihre Produkte oder Dienstleistungen

anpassen zu können. In der letzten Kategorie (CAT 9) wird hervorgehoben, dass vor allem die *User Experience* (deutsch: Benutzer:innenerfahrung) eine bedeutende Rolle bei der Entwicklung von Produkten oder Dienstleistungen spielt (Souza et al., 2019, S. 200–201).

Eine weitere interessante Studie von Teegne et al. (2019) beschäftigten sich ebenfalls mit Software-Entwicklungsmethoden von Software-Startups und identifizierten sowie analysierten im Zuge dessen wissenschaftliche Literatur, die zwischen den Jahren 2006 und 2017 veröffentlicht wurde. Um die Ergebnisse aus der Literaturstudie zu validieren wurde zudem eine empirische Studie durchgeführt, bei der insgesamt 14 Software Startups aus Finnland, Italien und Norwegen interviewt wurden. Es wurde herausgefunden, dass Agile sowie Lean Startup Methoden in Software Startups am häufigsten zum Einsatz kommen. Weitere relevante Konzepte, die im Rahmen dieser Studie genannt wurden, sind unter anderem *Minimum Viable Product* (deutsch: minimal brauchbares Produkt), *Business Model* (deutsch: Geschäftsmodell) sowie auch *Continuous Testing and Integration* (deutsch: kontinuierliches Testen und Integration) (Teegne et al., 2019, S. 501–506).

1.4.2 Schlussfolgerungen aus der Literaturrecherche

Grundsätzlich zeigte die Literaturrecherche, dass zu dem Themengebiet Product-led Growth bis zum heutigen Stand noch keine umfassende wissenschaftliche Literatur veröffentlicht wurde. Folglich ist davon auszugehen, dass es sich bei Product-led Growth um ein Konzept handelt, welchem innerhalb der Forschung nur geringe Beachtung geschenkt wird und vorwiegend in der Praxis Anwendung findet. Aus diesem Grund wurde für die Wissensgenerierung innerhalb dieser Domäne eine internetbasierte Recherche fortgeführt und ausschließlich auf die folgende Quelle zurückgegriffen: <https://www.productled.org/>.

Bei productled.org handelt es sich um eine Website, auf der ein Kollektiv von Experten oder Expertinnen über Product-led Growth informiert und regelmäßig Beiträge sowie Tipps und Hilfestellungen für Unternehmen publiziert. Veröffentlicht wurde die Website von Appcues, einem im Jahr 2013 gegründeten Unternehmen, welches sich auf Product-led Growth spezialisiert hat und Unternehmen eine Software zur Schaffung positiver Nutzer:innenerlebnisse bietet (Appcues, 2021).

Zusammenfassend konnte im Laufe der Recherche festgestellt werden, dass Product-led Growth sowohl fachliche als auch funktionale Anforderungen an eine Software beinhaltet, aus einer Zusammensetzung mehrerer Konzepte besteht und grundsätzlich als Weiterentwicklung von bereits existierenden Themengebieten angesehen werden kann. Die theoretischen Grundlagen zu Product-led Growth werden in Abschnitt 2.1 betrachtet.

Im folgenden Abschnitt 1.5 wird der weitere Aufbau der vorliegenden Arbeit vorgestellt und ein kurzer Einblick über den Inhalt der Kapitel geboten.

1.5 Struktur der Arbeit

Wie in Abbildung 2 ersichtlich, lässt sich die vorliegende Arbeit insgesamt in sechs Hauptkapitel mit entsprechenden Unterkapiteln gliedern.

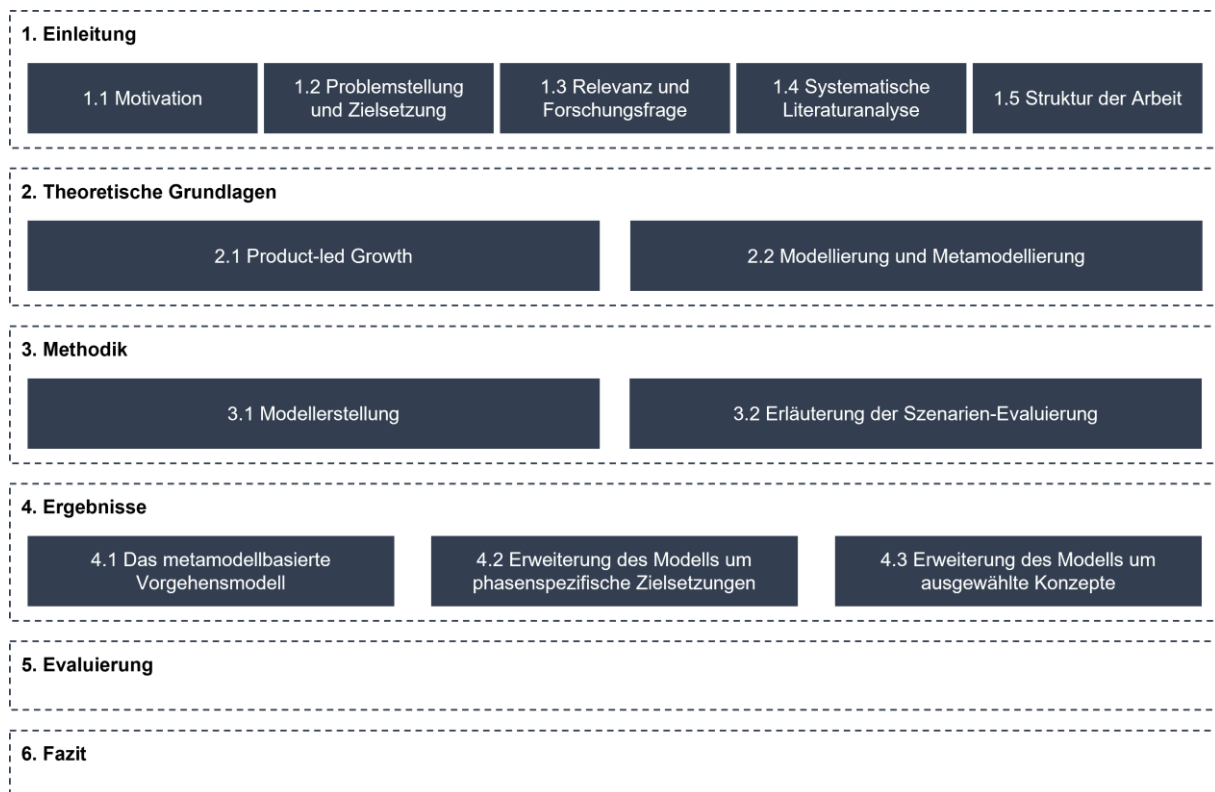


Abbildung 2: Strukturbild der vorliegenden Arbeit (Eigenerstellung)

Kapitel 1 hat eine kurze Einführung in die Thematik geliefert, den Fokus dieser Arbeit aufgezeigt sowie die durchgeführte systematische Literaturrecherche beschrieben.

Kapitel 2 bietet eine Einführung in die relevanten theoretischen Grundlagen dieser Arbeit. Darin wird zunächst in Abschnitt 2.1 das Konzept Product-led Growth mit den wesentlichsten Merkmalen und Konzepten vorgestellt. Anschließend werden in Abschnitt 2.2 die Grundlagen der Metamodellierung näher beleuchtet. Beide Unterkapitel dienen insbesondere dazu, Wissen innerhalb dieser Domänen aufzubauen, um schließlich ein Ergebnis im Sinne eines metamodellbasierten Vorgehensmodells auf Basis von Product-led Growth erzielen und die Forschungsfragen beantworten zu können.

Innerhalb des Kapitels 3 wird die im Rahmen dieser Arbeit zum Einsatz kommende Methodik vorgestellt. Abschnitt 3.1 gibt die Vorgehensweise zur Erstellung des Product-led Growth Modells wieder. Dabei wird zunächst das genutzte

Modellierungswerkzeug *CoChaCo4ADOxx* vorgestellt und anschließend der Prozess der Modell-Entwicklung aufgezeigt. Abschnitt 3.2 beschreibt das Vorgehen der Szenarien-Evaluierung.

Kapitel 4 beinhaltet die im Rahmen dieser Arbeit erarbeiteten Ergebnisse. Abschnitt 4.1 stellt das entwickelte metamodellbasierte Vorgehensmodell vor und gibt dazu eine detaillierte Erklärung. Der anschließende Abschnitt 4.2 beinhaltet das metamodellbasierte Vorgehensmodell erweitert um phasenspezifische Zielsetzungen. In Abschnitt 4.3 werden spezifische Konzepte im Zusammenhang mit Product-led Growth behandelt und das Modell erweitert um diese dargestellt.

Kapitel 5 befasst sich mit der Evaluierung des metamodellbasierten Vorgehensmodells. Durch eine Szenarien-Evaluation soll die konkrete Anwendung der erarbeiteten Ergebnisse anhand eines bestimmten Szenarios validiert werden. Im Zuge dessen wird im Hinblick auf das Szenario das Software Produkt ADOGRC der *BOC Group* (deutsch: BOC Gruppe) als Vertreter herangezogen.

Im abschließenden Kapitel 6 werden die wichtigsten Aussagen dieser Arbeit zusammengefasst, die gefundenen Ergebnisse diskutiert und ein Ausblick auf weiterführende Literatur gegeben.

2 Theoretische Grundlagen

Kapitel 2 gibt einen Einblick in die theoretischen Grundlagen der vorliegenden Arbeit, die für das Verständnis zur Erstellung des metamodellbasierten Vorgehensmodells auf Basis von Product-led Growth von Relevanz sind. Im Zuge dessen, wird zunächst in Abschnitt 2.1 das Konzept Product-led Growth vorgestellt und die wichtigsten Identifikationsmerkmale erläutert. In dem darauffolgenden Abschnitt 2.2 wird Modellierung sowie auch Metamodellierung behandelt. Der Wissensaufbau innerhalb dieser zwei Domänen ist für die Entwicklung des in Kapitel 4 dargestellten Product-led Growth Modells essenziell.

2.1 Product-led Growth

Product-led Growth stellt im Bereich der Software-Entwicklung ein immer häufiger zur Anwendung kommendes Konzept dar, das aufzeigt, wie die Markteinführung eines Produkts geschehen soll (Shein, 2022). Grundsätzlich finden bei einer Markteinführungsstrategie insbesondere die vier Elemente Produkt, Preis, Distribution und Kommunikation Berücksichtigung, die zusammengesetzt den sogenannten Marketing-Mix ergeben, welcher Einfluss auf die Kauf- oder Nutzungsentscheidung von Personen nimmt (Schuhmacher et al., 2018, S. 118). Die vier genannten Elemente sind in der folgenden Abbildung 3 dargestellt.



Abbildung 3: Berücksichtigung des Marketing-Mix bei der Go-to-Market Strategie (in Anlehnung an (Schuhmacher et al., 2018, S. 112))

Ziel einer Markteinführungsstrategie ist es, ein Produkt oder eine Dienstleistung möglichst schnell und zudem profitabel an die richtige Kundschaft zu bringen (Kuester et al., 2018, S. 66). Ein Kunde oder eine Kundin ist dann als „richtig“ anzusehen, wenn

die Person der Zielgruppe angehört, die das Unternehmen mit dem Produkt erreichen möchte (Schuhmacher et al., 2018, S. 119). Die Wahl der Markteinführungsstrategie ist weitgehend entscheidend für den Erfolg oder Misserfolg von Produkten (Kuester et al., 2018, S. 65).

Wie der Name Product-led Growth an sich bereits verdeutlicht, steht bei diesem Ansatz das Produkt im Zentrum der Betrachtung und zeigt auf, dass die Markteinführung sowie das weitere Wachstum des Marktanteils hauptsächlich durch das Produkt selbst erfolgt (Appcues, 2021). Appcues definiert Product-led Growth wie folgt:

“Product-led Growth (PLG) is a business methodology in which user acquisition, expansion, conversion, and retention are all driven primarily by the product itself” (Appcues, 2021).

Der Definition zufolge soll *User acquisition* (deutsch: Nutzer:innen-Generierung), *User expansion* (deutsch: Nutzer:innen-Erweiterung), *User conversion* (Nutzer:innen-Umwandlung) sowie auch *User retention* (deutsch: Nutzer:innen-Bindung) vorwiegend durch das Produkt selbst gesteuert und vorangetrieben werden. Die folgende Abbildung 4, abgeleitet aus der obigen Beschreibung, soll dieses Verhältnis veranschaulichen (Appcues, 2021).

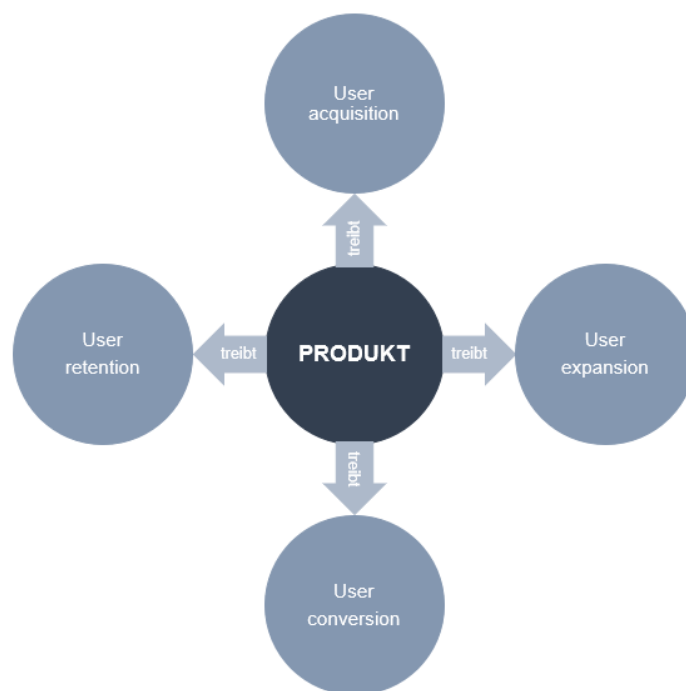


Abbildung 4: Das Produkt als hauptsächlicher Wachstumstreiber (in Anlehnung an (Appcues, 2021))

Im Rahmen von Product-led Growth wird hervorgehoben, dass ein Produkt als hauptsächlicher Wachstumstreiber nur erfolgreich sein kann, wenn es einerseits eine eindeutige Lösung für ein konkretes Problem einer Zielgruppe darstellt, folglich einen

bestimmten Mehrwert bietet und insgesamt auf die Schaffung eines einzigartigen und nahtlosen Produkt- und Nutzer:innenerlebnisses ausgerichtet ist. Die Begriffe „einzigartig“ und „nahtlos“ bedeuten in diesem Zusammenhang, dass jegliche Art von Reibung in Form von etwaigen entstehenden Unannehmlichkeiten für den oder die Nutzer:in des Produkts unbedingt vermieden werden sollten (Lemon & Verhoef, 2016, S. 84). Beispielsweise könnte es für Nutzer:innen ein Hindernis darstellen, für die erstmalige Verwendung eines Produkts zahlen zu müssen, ohne dieses Produkt zuvor je getestet zu haben. Eine weitere Unannehmlichkeit könnte sich daraus ergeben, dass das Produkt für den oder die Nutzer:innen nicht durchgehend zu Verfügung steht, wie bei Ausfällen einer Software aufgrund von etwaigen Updates. Zur Vermeidung der erstgenannten Problemstellung wird im Rahmen von Product-led Growth explizit auf die Verwendung eines sogenannten *Subscription Models* (deutsch: Abonnementmodelle), wie dem Business Modell *Freemium* (zusammengesetzter Begriff aus *Free* (deutsch: Gratis) und *Premium*) verwiesen. Bei solchen Abonnementmodellen können Nutzer:innen gewisse Funktionen eines Produkts unbegrenzt oder zumindest für einen gewissen Zeitraum kostenlos testen und nutzen (Shein, 2022).

Da Unternehmenserfolg auf der Schaffung von Werten basiert und diese ein wesentliches Unterscheidungsmerkmal von Unternehmen darstellen, ist es im Hinblick auf den zuvor erwähnten Mehrwert eines Produkts maßgeblich entscheidend, dass dieser dem oder der Nutzer:in unmittelbar erkenntlich ist (Lindič & Da Marques Silva, 2011, S. 1694). Im Rahmen von Product-led Growth wird in Bezug darauf auch häufig von dem sogenannten „Aha-Moment“ gesprochen, der sich auf den Zeitpunkt, in welchem der oder die Nutzer:in den eigentlichen Wert eines Produkts erkennt bezieht. Im Vordergrund von Product-led Growth steht daher die Generierung eines möglichst kurzen Time-to-Value (Appcues, 2021).

Bei Product-led Growth wird des Weiteren von einem auf den *End-User* (deutsch: Endnutzer:in) fokussierten Ansatz gesprochen, bei welchem nicht ausschließlich der oder die Käufer:in eines Produkts oder Services sondern vorwiegend der oder die Endnutzer:in mit seinen oder ihren Wünschen und Bedürfnissen betrachtet wird. Unter Endnutzer:in werden Personen verstanden, die die zur Nutzung erworbene Software persönlich verwenden. Der oder die Käufer:in muss infolgedessen nicht gleichzeitig auch der oder die Nutzer:in des Produkts sein und vice versa (Appcues, 2021).

Die Konzentration auf Endnutzer:innen erklärt sich dadurch, dass diese Personengruppe auf eine zuverlässige und dauernd zu Verfügung stehende Bereitstellung von Diensten sowie auch auf eine einfach zu bedienende *User Interface* (deutsch: Benutzerschnittstelle) angewiesen ist, um ihre Tätigkeiten ausführen zu können. Ein Konzept, welches bei der Abgrenzung von Zielgruppen unterstützen soll und im Rahmen von Product-led Growth vorgeschlagen wird, ist der Einsatz von

Personas. Unter Personas wird die visuelle Darstellung fiktiver Personen verstanden, mit Hilfe derer ein besseres Verständnis für eine spezifische Kunden- oder Kundinnen und Nutzer:innengruppe herbeigeführt werden soll. Ein weiteres Konzept, welches an dieser Stelle empfohlen wird, ist das *Customer Journey Mapping* (deutsch: Abbildung der Kunden- oder Kundinnenreise), bei welchem die gesamte Reise des oder der Kunden oder Kundin oder des oder der Nutzers oder Nutzerin mit einem Produkt oder einem Unternehmen dargestellt werden kann (Appcues, 2021).

Grundsätzlich lässt sich erkennen, dass sich Product-led Growth aus mehreren Konzepten zusammensetzt und als Weiterentwicklung von bereits existierenden Themen angesehen werden kann.

Eine genauere Erläuterung in Bezug auf Product-led Growth empfohlener Konzepte, ist in Abschnitt 4.3 zu finden. An dieser Stelle gilt es vorab zu betonen, dass es den Rahmen dieser Arbeit sprengen würde, jedes einzelne dieser Themen ausführlich zu beschreiben, weshalb hierbei der Kontext und die Verknüpfung mit dem Product-led Growth Ansatz im Vordergrund stehen soll. Der nachfolgende Abschnitt gibt einen Überblick über die Grundlagen der Metamodellierung.

2.2 Modellierung und Metamodellierung

In Abschnitt 2.2 wird zunächst aufgezeigt, was unter einem Modell verstanden wird und welche grundlegenden Merkmale Modelle aufweisen. Anschließend wird auf das Rahmenwerk der Modellierung eingegangen und im darauffolgenden Abschnitt Metamodellierung thematisiert.

2.2.1 Der Modell-Begriff

Modelle sind eine vereinfachte Abbildung der Wirklichkeit, sprich eine Darstellung der Realität oder einer Vision, mit denen beschrieben werden kann, wie etwas ist oder wie etwas sein soll. Bei der Modellerstellung werden folglich nur Teilaspekte eines Originals wiedergeben, Komplexität durch Abstraktion reduziert sowie irrelevante Objekte nicht berücksichtigt und folglich weggelassen (Karagiannis & Höfferer, 2006, S. 28).

Carolla (2015, S. 23–24) gibt an, dass Modelle durch drei Merkmale gekennzeichnet sind, die in der folgenden Tabelle 9 ersichtlich sind.

Merkmal	Beschreibung
Abbildungsmerkmal	Ein Modell stellt immer eine Repräsentation von einem natürlichen oder künstlichen Original dar. Es kann also durchaus von einem Modell ein weiteres Modell erstellt werden, in welchem ein Original um einen zusätzlichen Schritt vereinfacht dargestellt wird.

Verkürzungsmerkmal	Das Merkmal der Verkürzung besagt, dass ein Modell nur relevante Details eines Originals wiedergeben muss und Modelle folglich Abstraktionen des Originals sind.
Pragmatisches Merkmal	Modelle verfolgen stets einen bestimmten Zweck und beschreiben, für wen das Modell erstellt wird sowie auch wann und wozu das Modell genutzt wird.

Tabelle 9: Merkmale eines Modells

Modelle können nach ihrer jeweiligen Sprache, die zur Erstellung verwendet wurde klassifiziert werden, wobei eine Unterscheidung zwischen ikonischen und linguistischen Modellen getroffen wird. Erstere nutzen Zeichen und Symbole, die eine offensichtliche Ähnlichkeit zu der Realität die dargestellt werden soll besitzen. Linguistische Modelle hingegen kennzeichnen sich dadurch, dass sie keine offensichtliche Beziehung zu dem modellierten Teil der Realität aufweisen (Karagiannis & Höfferer, 2006, S. 28). Bevor nun auf den Begriff Metamodell näher eingegangen werden kann, ist es zunächst erforderlich aufzuzeigen, wie Modelle überhaupt erstellt werden können. Darauf wird im folgenden Abschnitt 2.2.2 näher eingegangen.

2.2.2 Rahmenwerk der Modellierung

Während in vielen wissenschaftlichen Arbeiten die Begriffe „Modellierungsmethode“ und „Modellierungssprache“ als Synonyme verwendet werden, fassen Karagiannis und Kühn (2002) ersteren Begriff abstrakter und sehen eine Modellierungssprache als einen von drei bedeutenden Teilen der Modellierungsmethode an. In Abbildung 5 ist das von ihnen entwickelte Rahmenwerk für Modellierungsmethoden dargestellt, welches aufzeigt, dass eine Modellierungsmethode aus den folgenden drei Hauptkomponenten besteht:

- Modellierungssprache
- Modellierungsvorgehen
- Mechanismus und Algorithmen

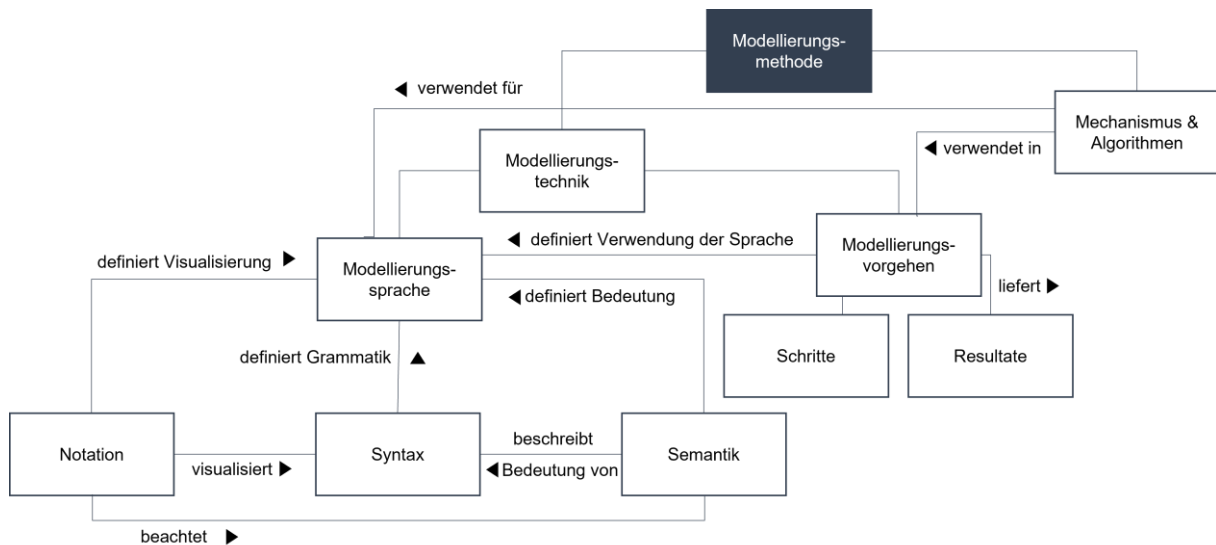


Abbildung 5: Modellierungs-Rahmenwerk (in Anlehnung an (Karagiannis & Kühn, 2002))

Modellierungssprache und Modellierungsvorgehen gehören zur Modellierungstechnik. Eine Modellierungssprache beinhaltet alle Elemente die notwendig sind, um ein Modell zu beschreiben und kann folglich als ein System von Zeichen verstanden werden, deren Verwendung durch bestimmte Regeln festgelegt ist. Allgemein gesprochen gibt die Modellierungssprache vor, welche Zeichen in einem Modell verwendet werden dürfen und in welcher Beziehung diese zueinander gesetzt werden können (Karagiannis & Kühn, 2002, S. 2).

Eine Modellierungssprache setzt sich aus einer Notation, Syntax und Semantik zusammen. Bei der Notation handelt es sich um die graphische Darstellung und Repräsentation der Elemente einer Modellierungssprache. Die Relevanz der Notation ergibt sich aus der einfachen Verständlichkeit für Anwender:innen, der prägnanteren Vermittlung von Informationen, der höheren Genauigkeit sowie der leichteren Merkbarkeit aufgrund des Bildüberlegenheitseffekts (Phänomen, bei dem Bilder eher in Erinnerung bleiben als Wörter). Die Syntax gibt Auskunft darüber, wie diese repräsentierten Elemente kombiniert werden dürfen. Die Semantik wiederum definiert die Bedeutung einer Modellierungssprache (Karagiannis & Kühn, 2002, S. 4).

Das Modellierungsvorgehen zeigt auf, wie die Konstrukte zur Modellierung kombiniert werden müssen, um ein valides Modell zu erstellen. Es enthält Anweisungen für den oder die Modellierer:in und gibt die notwendigen Schritte vor, die zum gewünschten Ergebnis führen sollen. An dieser Stelle gilt zu betonen, dass das Modellierungsvorgehen in einigen Fällen nicht definiert ist (Karagiannis & Kühn, 2002, S. 2).

Algorithmen und Mechanismus bieten Funktionalitäten, um Modelle, die in einer Modellierungssprache beschrieben sind zu verwenden und zu evaluieren. Beispiele dafür können sein, Simulationen, Transformationen, Importe und Exporte (Karagiannis & Kühn, 2002, S. 2)

Nachdem nun das Modellierungs-Rahmenwerk vorgestellt wurde, wird im folgenden Abschnitt näher auf das Metamodell im Speziellen eingegangen.

2.2.3 Der Metamodell-Begriff

Ein Metamodell legt die Konzepte, Beziehungen und Regeln fest, die zur Spezifikation einer Modellierungssprache dienen. In der Wissenschaft wird ein Metamodell auch häufig als ein Modell von einem (ähnlichen) Modell verstanden. Ein Modell kann also als Instanz eines Metamodells verstanden werden. Diese Definition kann anhand der folgenden Abbildung 6 erklärt werden (Karagiannis & Höfferer, 2006, S. 28).

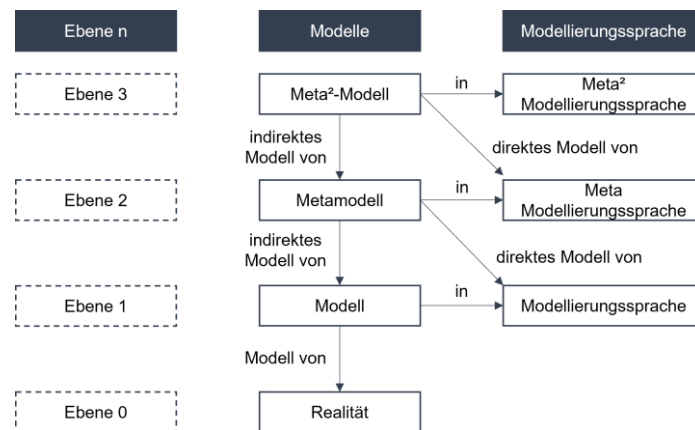


Abbildung 6: Metamodellierungs-Hierarchie (in Anlehnung an (Karagiannis & Höfferer, 2006))

Die unterste Ebene 0 gibt die Realität mitsamt den zu modellierenden Objekten und Fakten wieder. Von diesem Original wird ein Modell (Ebene 1) erstellt, das mit Hilfe einer bestimmten Modellierungssprache beschrieben wird. Die zur Verfügung stehenden Modellierungs-Konstrukte sind im Metamodell (Ebene 2) definiert, die besagen, wie die Modelle auf Ebene 1 aufgebaut sein müssen und geben Auskunft über Struktur, Elemente, Beziehungen und Eigenschaften. Ein Metamodell kann also als direktes Modell einer Modellierungssprache verstanden werden. Metamodelle werden folglich wiederum mit einer Meta-Modellierungssprache beschrieben, die in einem sogenannten Meta²-Modell (Ebene 3) abgebildet wird (Karagiannis & Kühn, 2002, S. 2).

Hinsichtlich der Bestandteile eines Metamodells (Ebene 2) sind innerhalb der Literatur verschiedene Begriffe zu finden und in Verwendung. Um dennoch auf die wichtigsten

Komponenten eines Metamodells näher eingehen zu können, wird zur Erklärung dieser, auf die in Abbildung 7 ersichtliche Terminologie zurückgegriffen.

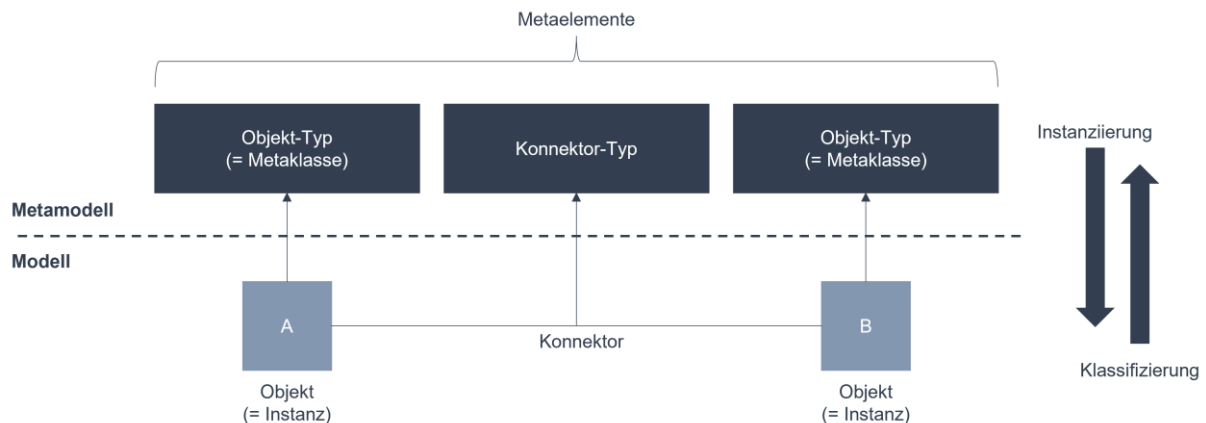


Abbildung 7: Metaelemente (in Anlehnung an (Bork et al., 2020, S. 2))

Jegliche Elemente die auf der Metamodell-Ebene zu finden sind, werden als Elemente des Metamodells oder auch Metaelemente bezeichnet. Diese Elemente können entweder ein Objekt-Typ oder ein Konnektor-Typ darstellen, wobei ersteres auch als Metaklasse verstanden werden kann. Objekt-Typ ist eine Menge von Objekten mit gleichen Merkmalen. Ein Konnektor-Typ verbindet Objekt-Typen. Bei Instanziierung der Metaelemente auf die Modell-Ebene werden Objekt-Typen zur Instanz „Objekt“ und Konnektor-Typen zur Instanz „Konnektor“. Bei Betrachtung der Modell-Ebene findet nach oben hin auf die Metamodell-Ebene wiederum eine Klassifizierung statt (Bork et al., 2020, S. 2).

In der folgenden Abbildung ist ein vereinfachtes Beispiel eines Metamodells und seinen Konzepten angeführt.

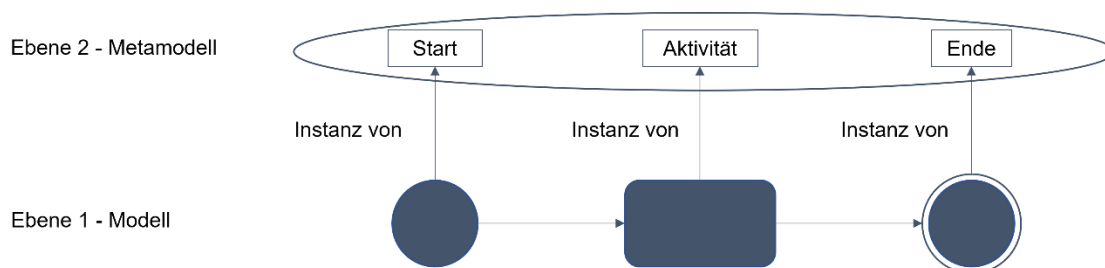


Abbildung 8: Vereinfachtes Beispiel eines Metamodells (in Anlehnung an (Karagiannis et al.))

Ma et al. (2013, S. 560) geben in ihrer wissenschaftlichen Arbeit an, dass einem Metamodell grundsätzlich die folgenden vier Eigenschaften zugesprochen werden können. Diese sind in der folgenden Tabelle 10 zu entnehmen.

Nr.	Eigenschaft
1	Metamodelle bestehen aus einer Syntax, Semantik und gegebenenfalls einer Notation, die zur Modellerstellung dienen.
2	Metamodelle sind in einer Metamodellierungssprache definiert.
3	Metamodelle werden in der Regel zur Modellierung in einem bestimmten Bereich (Domäne) verwendet.
4	Metamodelle werden durch Modellierer:innen rasch verstanden und können folglich ohne Schwierigkeiten Einsatz finden.

Tabelle 10: Eigenschaften eines Metamodells

Die in Kapitel 2 behandelten theoretischen Grundlagen sind essenziell für das Verständnis der im folgenden Kapitel 3 aufgezeigten Methodik sowie für das in Kapitel 4 dargestellte metamodellbasierte Vorgehensmodell.

3 Methodik

Für ein besseres Verständnis des in Kapitel 4 dargestellten Product-led Growth Modells ist es von Bedeutung die Methodik, sprich das Vorgehen bei der Modellerstellung sowie Evaluierung des Modells näher zu erläutern. Kapitel 3 gibt daher einen detaillierten Einblick, wie aus dem Konzept Product-led Growth ein metamodellbasiertes Vorgehensmodell entwickelt werden konnte (Abschnitt 3.1). In Abschnitt 3.2 wird anschließend das Vorgehen der Szenarien-Evaluierung aufgezeigt und beschrieben.

3.1 Modellerstellung

Für die Modellerstellung wurde das Modellierungswerkzeug CoChaCo4ADOxx herangezogen, welches in Abschnitt 3.1.1 näher erläutert wird. Anschließend wird in Abschnitt 3.1.2 auf die konkrete Erstellung des PLG-Modells mithilfe dieses Tools eingegangen.

3.1.1 Einführung in CoChaCo

Bei der Erstellung des Product-led Growth Modells wurde auf das kostenlose Modellierungs-Werkzeug CoChaCo4ADOxx zurückgegriffen, welches von o. Univ.-Prof. Dr. Dimitris Karagiannis und Dip.-Ing. Patrik Burzynski entwickelt wurde und auf www.omilab.org abrufbar ist. OMILAB (kurz für Open Models Laboratory) ist ein Labor für die Entwicklung von Modellierungsmethoden und bietet Konzepte, Werkzeuge sowie auch eine Modellierungsplattform. So verweist auch der Name des Werkzeugs CoChaCo4ADOxx indirekt auf die zugrundeliegende Plattform ADOxx (Burzynski & Karagiannis, 2020, S. 3).

Bei CoChaCo handelt es sich um eine Modellierungsmethode, die Modellierer oder Modelliererinnen bei der Entwicklung von Metamodellen unterstützen soll. Das Akronym CoChaCo steht für die Konstrukte *Concept-Characteristic-Connector* (deutsch: Konzept-Eigenschaft-Konnektor), durch welche die Zusammensetzung eines Metamodells beschrieben werden kann. Weitere Konstrukte wie Zweck und Funktionalität erlauben es zudem den Nutzen des Metamodells wiederzugeben. Die unterschiedlichen Typen sowie der Möglichkeit deren Instanzen zueinander in Beziehung zu setzen, erlauben die Erstellung von Metamodellen unabhängig von einer Plattform, einem bestimmten Rahmenwerk oder einer Implementierung. So soll es ermöglicht werden, sich auf den Kern der Metamodellierung zu konzentrieren. Im Vordergrund von CoChaCo steht folglich der allgemeine Wert der Modellierung als Mittel der Wissensdarstellung (Karagiannis et al., 2019, S. 203).

Bei der Entwicklung eines Metamodells mit CoChaCo wird empfohlen, in einem ersten Schritt Überlegungen hinsichtlich dem Zweck des Metamodells sowie auch der relevanten *Stakeholder* (deutsch: Interessensgruppen) anzustellen. Anschließend sollten Zielsetzungen identifiziert und weitere Konstrukte wie Konzepte, Eigenschaften, Konnektoren und Funktionalitäten einbezogen werden, die bei der Erreichung der zuvor definierten Ziele unterstützend wirken. Die identifizierten Elemente sollten in einem weiteren Schritt in eine sinnvolle Beziehung zueinander gebracht werden, um die Struktur des Metamodells zu beschreiben und um aufzuzeigen, wie sie genutzt werden sollen (Burzynski & Karagiannis, 2020, S. 4). Die folgende Abbildung 9 zeigt das von CoChaCo empfohlene Modellierungsvorgehen:

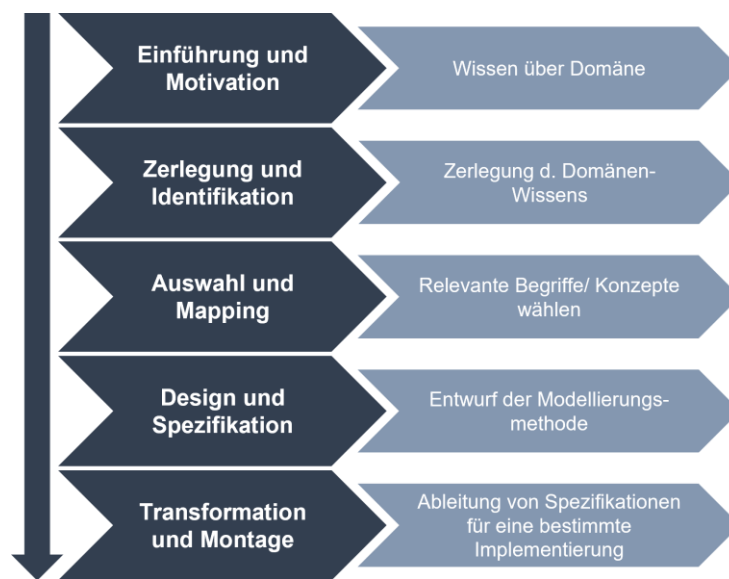


Abbildung 9: Von CoChaCo empfohlene Schritte zur Erstellung eines Metamodells (in Anlehnung an (Burzynski & Karagiannis, 2020, S. 4))

Als Modellierungsmethode beinhaltet CoChaCo als weiteren wichtigen Baustein die Struktur der Sprache. Dabei stehen die in Abbildung 10 gezeigten Hauptkonstrukte im Fokus von CoChaCo:

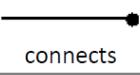
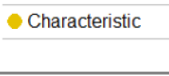
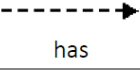
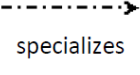
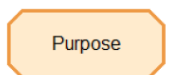
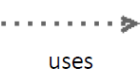
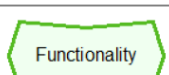
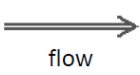
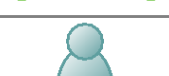
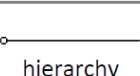
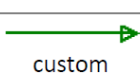
ELEMENTS		RELATIONS	
	Something that is relevant and worth to be represented or modelled.		Indicates that something (typically a Connector) is able to connect things.
	Something that describes something else, being a feature, an aspect or a property of the other thing.		Indicates that something has or owns something.
	Something that is primarily used to connect, relate or link things together.		Indicates that something is a more special thing of something else or that it is divided into specializations based on something else.
	Something that is desired to be achieved through the use of the meta-model.		Indicates that something makes use of something else.
	Something that should assist or that is required to realize a purpose and uses parts from the meta-model.		Indicates a flow between things, either in a sense of ordering elements or of flowing into elements.
	Someone who has some form of interest or stake in the designed meta-model.		Indicates some form of hierarchy between things, from the higher-level to the lower-level.
			A custom relation with any chosen meaning or without a specific meaning.

Abbildung 10: Hauptkonstrukte von CoChaCo (in Anlehnung an (Burzynski & Karagiannis, 2020, S. 5))

Zusätzlich zu den Hauptelementen gibt es noch weitere Konstrukte, wie beispielsweise Kategorien, Aggregationen und Notizen, die, wenn für das Metamodell nützlich, eingesetzt werden können.

CoChaCo bietet dem oder der Modellierer:in drei unterschiedliche Modelltypen, welche jeweils auf einen anderen Aspekt fokussieren und in der Tabelle 11 aufgeschlüsselt sind.

Modelltyp	Beschreibung
Hierarchiemodell	In diesem Modell können alle Kernelemente gesammelt und in einer Hierarchie angeordnet werden. Es dient als Art <i>Repository</i> (deutsch: Lagerstätte), durch welches es ermöglicht wird, sich auf für das Metamodell relevante Begriffe zu konzentrieren und kann folglich als Brainstorming oder auch als Mind Map genutzt werden.
Übersichtsmodell	Dieses Modell dient insbesondere dazu Kernelemente näher darzustellen und zueinander in Beziehung zu setzen.
Vorgehensmodell	Das Vorgehensmodell dient dazu, relevante aktive Komponenten eines Metamodells in einem allgemeinen Verfahren zu beschreiben.

Tabelle 11: Von CoChaCo unterstützte Modelltypen (Burzynski & Karagiannis, 2020)

Vorwegzunehmen ist, dass in der vorliegenden Arbeit nur das Hierarchiemodell sowie das Übersichtsmodell Berücksichtigung finden.

Nachdem die Funktionsweise von CoChaCo erläutert wurde, wird im folgenden Kapitel 3.2.2 die konkrete Umsetzung näher betrachtet.

3.1.2 Modellerstellung mit CoChaCo

In der vorliegenden Arbeit wurde für die Erstellung des Product-led Growth Modells auf die in Abbildung 9 von CoChaCo empfohlenen Schritte zurückgegriffen und zum Großteil nach diesen vorgegangen. Die gewählten Schritte sind in Abbildung 11 ersichtlich, auf welche in diesem Abschnitt folglich näher eingegangen wird.



Abbildung 11: Vorgehensweise bei der Erstellung des PLG-Modells (in Anlehnung an (Burzynski & Karagiannis, 2020, S. 4))

Schritte 1 & 2: Hauptzweck bestimmen und Stakeholder identifizieren: Die ersten beiden Schritte aus Abbildung 11 wurden bereits in Abschnitt 1.2 bei der Definition der Problemstellung und Zielsetzung behandelt. Demzufolge ist der Hauptzweck des metamodellbasierten Vorgehensmodells als Blueprint im Sinne eines Konzepts zu dienen, aus welchem phasenspezifische Zielvorgaben sowie auch ausgewählte exemplarische Konzepte hervorgehen und ablesbar sind. Die Anwendung des Konzepts der Metamodellierung würde eine einfachere Instanziierung von Product-led Growth für (metamodellbasierte) Software-Produkte ermöglichen. Demnach ist das entworfene Modell insbesondere für Personen und Unternehmen relevant, die entweder ein Software-Angebot entwickeln oder aber ein bereits existierendes Software-Angebot unter Berücksichtigung von Product-led Growth umstrukturieren möchten.

Schritt 3: Hierarchiemodell erstellen: In einem weiteren Schritt galt es, alle in Bezug auf Product-led Growth relevanten Anforderungen zu sammeln und festzuhalten. Für

diesen Prozess des Mind Mappings wurde das Hierarchiemodell von CoChaCo herangezogen. Jede Anforderung wurde in das Hierarchiemodell überführt, einem der in Abbildung 12 ersichtlichen Elemente zugeordnet, konkretisiert und wenn notwendig auch erweitert.

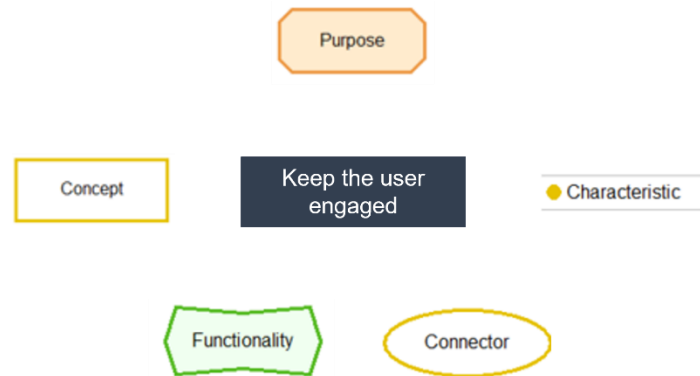


Abbildung 12: Zuordnung einer Anforderung zu einem der fünf Konstrukte (Eigendarstellung)

Durch die Kategorisierung hat sich ergeben, dass für die Anwendung von Product-led Growth vorwiegend die beiden Elemente *Purpose* (deutsch: Ziel) und *Concept* (deutsch: Konzept) von Bedeutung sind. Folglich konnten alle Anforderungen entweder in ein Element vom Typ Purpose oder in ein Element vom Typ Concept eingeordnet werden. Bei Bedarf können Elemente laufend in ein anderes Element umgewandelt werden. Beispielsweise könnte eine Anforderung, die zunächst als Konzept-Element definiert wurde, in ein Ziel-Element umgeformt werden.

Nachdem alle relevanten Anforderungen ins Hierarchiemodell überführt worden waren, hat sich herauskristallisiert, dass bestimmte Anforderungen als zusammengehörig angesehen und als Gruppen dargestellt werden konnten. Aus diesen Gruppierungen haben sich sieben Phasen ergeben, die bei Anwendung von Product-led Growth als relevant anzusehen sind.

Hervorzuheben ist, dass der Prozess des Mind Mappings mit CoChaCo nicht als starr anzusehen, sondern viel mehr von laufender Überarbeitung gezeugt ist. Die folgende Abbildung 13 zeigt das erstellte Hierarchiemodell mitsamt den relevanten Phasen, Zielvorgaben und ausgewählten exemplarischen Konzepten. Wie im Hierarchiemodell ersichtlich, beinhaltet nicht jede Phase gleich viele Zielvorgaben und Konzepte, da im Rahmen dieser Arbeit nur die Relevantesten explizit aufgezeigt werden und Berücksichtigung finden.

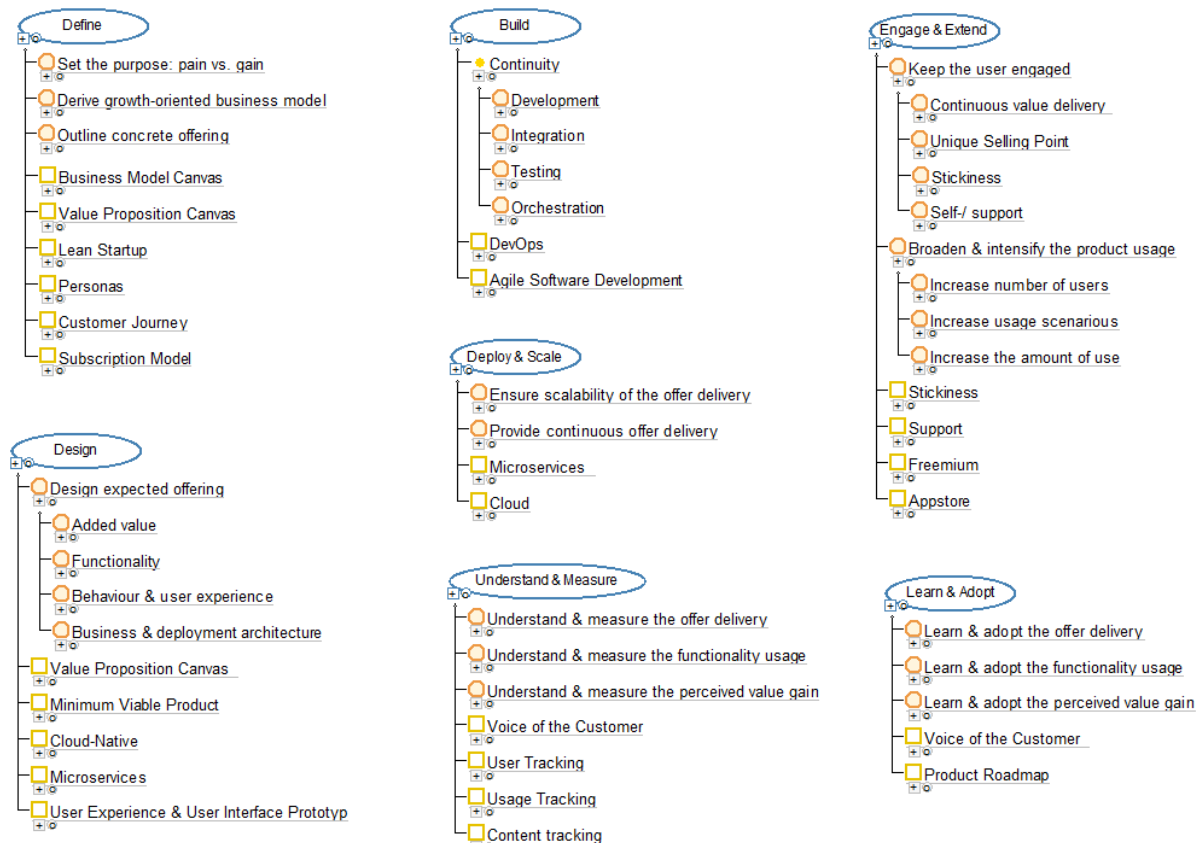


Abbildung 13: Nutzung des Hierarchiemodells als Mind Mapping (Eigendarstellung)

Schritt 4: Übersichtsmodell erstellen: Nachdem das Hierarchiemodell vorerst als vollständig angesehen werden konnte, galt es in einem weiteren Schritt das dazugehörige Übersichtsmodell zu erstellen. In einer ersten Version wurden zunächst nur die sieben Phasen überführt, durch welche sich das in Abschnitt 4.1 dargestellte metamodellbasierte Vorgehensmodell, sprich das Kernmodell, ergeben hat. Dabei wurden die Phasen zuerst in einer logischen Reihenfolge geordnet und anschließend mit passenden Konnektoren in Verbindung gebracht. Für die Darstellung der Phasen wurde das Element „Aggregation“ gewählt und mit dem Element „Flow“ verbunden (siehe Abbildung 14). In einer weiteren Modellversion (siehe Abschnitt 4.2) wurde das Kernmodell erweitert, indem die identifizierten Zielsetzungen den einzelnen Phasen zugeordnet wurden. Die dritte Modellversion beinhaltet schlussendlich das Kernmodell, erweitert um phasenspezifische Zielsetzungen sowie auch ausgewählte Konzepte (siehe Abschnitt 4.3).

In Abbildung 14 sind die für die Erstellung des PLG-Modells zum Einsatz kommenden Elemente nochmals aufgelistet:

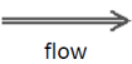
Elemente	Beschreibung
	Zielsetzungen die im Zusammenhang mit Product-led Growth identifiziert wurden
	Etwas, das zur Zielerreichung beiträgt/unterstützend wirkt
	Phasen, die im Prozess einer Product-led Growth Strategie durchlaufen werden müssen
	Zeigt die vorgegebene Flussrichtung und den Informationsaustausch zwischen den einzelnen Phasen

Abbildung 14: Elemente des PLG-Modells (in Anlehnung an (Burzynski & Karagiannis, 2020))

Nachdem die Vorgehensweise zur Modellerstellung erläutert wurde, wird im folgenden Abschnitt auf die Szenarien-Evaluierung näher eingegangen.

3.2 Erläuterung der Szenarien-Evaluierung

Wie bereits verdeutlicht, ist das Hauptziel dieser Arbeit die Entwicklung eines metamodellbasierten Vorgehensmodells auf Basis von Product-led Growth (Ergebnis 1), welches um phasenspezifische Zielsetzungen (Ergebnis 2) sowie auch ausgewählte Konzepte (Ergebnis 3) erweitert werden soll.

Aufgrund der Zielsetzung erscheint die Anwendung einer Hypothesen-Evaluation als nicht geeignet. Vielmehr soll durch eine Szenarien-Evaluation die konkrete Anwendung der erarbeiteten Ergebnisse anhand eines bestimmten Szenarios validiert werden. Im Zuge dessen wird im Hinblick auf das Szenario das Software-Produkt *ADOGRC* des Unternehmens BOC Group als Vertreter herangezogen.

Die im Jahr 1995 gegründete BOC Group entwickelt in Zeiten der digitalen Transformation Softwareprodukte und Services für eine effektive Unternehmensführung und bietet darüber hinaus fachliche Beratung und Trainings in den Bereichen Geschäftsprozessmanagement, IT- und Unternehmensarchitektur sowie auch *Governance, Risk & Compliance* an. Das Software Produkt *ADOGRC* stellt das jüngste Produkt von insgesamt drei Software Angeboten (neben *ADONIS* und *ADOIT*) dar und deckt die Domäne Governance, Risk & Compliance ab (BOC Group, 2022).

Im Zuge der Szenarien-Evaluation findet innerhalb dieser Arbeit ein Soll-Ist-Vergleich statt, bei welchem eine Gegenüberstellung des entwickelten metamodellbasierten Vorgehensmodells, erweitert um phasenspezifische Zielsetzungen und ausgewählte

exemplarische Konzepte, und dem aktuellen Vorgehen zur Produktentwicklung des Software Produkts ADOGRC erfolgt. Dieser Vergleich hat zum Zweck, etwaige Abweichungen aufzudecken, um anschließend gegebenenfalls Maßnahmen zur Reduzierung dieser Abweichungen umsetzen zu können. An dieser Stelle gilt zu betonen, dass im Zuge der Szenarien-Evaluation die beiden Produktmanager des Software Produkts ADOGRC mittels qualitativen Interviews einbezogen werden. Die folgende Abbildung 15 veranschaulicht das Vorgehen des Soll-Ist-Vergleichs.

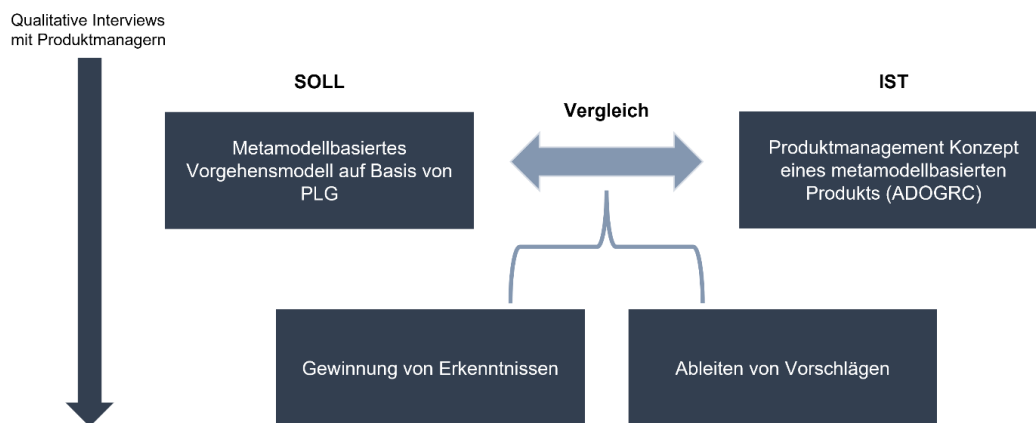


Abbildung 15: Vorgehen Szenarien-Evaluierung (Eigendarstellung)

In einem ersten Schritt besteht die Notwendigkeit, den beiden Produktmanagern das metamodellbasierte Vorgehensmodell auf Basis von Product-led Growth zu präsentieren, sie in die Thematik einzuführen und etwaige Unklarheiten vorab aufzuklären. Im Rahmen dessen erfolgt bereits eine erste Bewertung des entwickelten Modells durch die Produktmanager hinsichtlich etwaiger Verbesserungsmöglichkeiten.

In einem weiteren Schritt gilt es, die Ausgangssituation, sprich den Status quo des Szenarios, bei welchem es sich um das Software Produkt ADOGRC handelt, zu ermitteln und zu analysieren sowie alle relevante Informationen für das Gelingen des Soll-Ist-Vergleichs zusammenzutragen. Im Zuge dessen werden qualitative Interviews geführt, bei welchen das entworfene metamodellbasierte Vorgehensmodell mitsamt seinen Phasen, Zielsetzungen und Konzepten als Orientierung dienen soll.

Nachdem der Ist-Stand erhoben wurde, kann im weiteren Verlauf der eigentliche Soll-Ist-Vergleich erfolgen. Während dieser Gegenüberstellung wird das metamodellbasierte Vorgehensmodell auf Basis von Product-led Growth mit dem aktuellen Vorgehen zur Produktentwicklung von ADOGRC verglichen. Der Vergleich hat zum Zweck, dass Erkenntnisse über etwaige Optimierungspotenziale im Hinblick auf Product-led Growth gewonnen werden können. Darüber hinaus ist das Ziel definiert, Schlüsse zur allgemeinen Anwendbarkeit des Modells zu generieren sowie auch konkrete Vorschläge für das Software-Angebot ADOGRC hinsichtlich Product-led Growth abzuleiten.

4 Ergebnisse

In diesem Kapitel 4 wird nun das mit CoChaCo erstellte Modell vorgestellt. Während in Abschnitt 4.1 zunächst das metamodellbasierte Vorgehensmodell präsentiert wird, wird in Abschnitt 4.2 das Vorgehensmodell, erweitert um phasenspezifische Zielsetzungen dargestellt. In Abschnitt 4.3 findet eine Zuordnung ausgewählter Konzepte zu den Vorgehensphasen statt. Zudem werden ausgewählte Konzepte näher erläutert und abschließend zusammengefasst.

4.1 Das metamodellbasierte Vorgehensmodell

Wie bereits in Abschnitt 3.1.2 erläutert, haben sich durch die Gruppierung zusammengehöriger Zielsetzungen mehrere Phasen ergeben, die es hinsichtlich Product-led Growth als metamodellbasiertes Konzept zu berücksichtigen gilt. Diese Phasen inklusive ihrer Verbindungen stellen das in Abbildung 16 ersichtliche Vorgehensmodell mit der Bezeichnung „Product-led Growth-oriented cycle for software-offerings“ (deutsch: Product-led Growth orientierter Zyklus für Software-Angebote) dar. Dieses Modell kann als Kernmodell verstanden werden und zeigt ein konkretes Vorgehen zur Entwicklung moderner, softwarebasierter Produkte auf Basis von Product-led Growth. Das Vorgehensmodell besteht aus insgesamt sieben iterativen, dynamischen und miteinander in Verbindung stehenden Phasen, die im Folgenden näher erläutert werden.

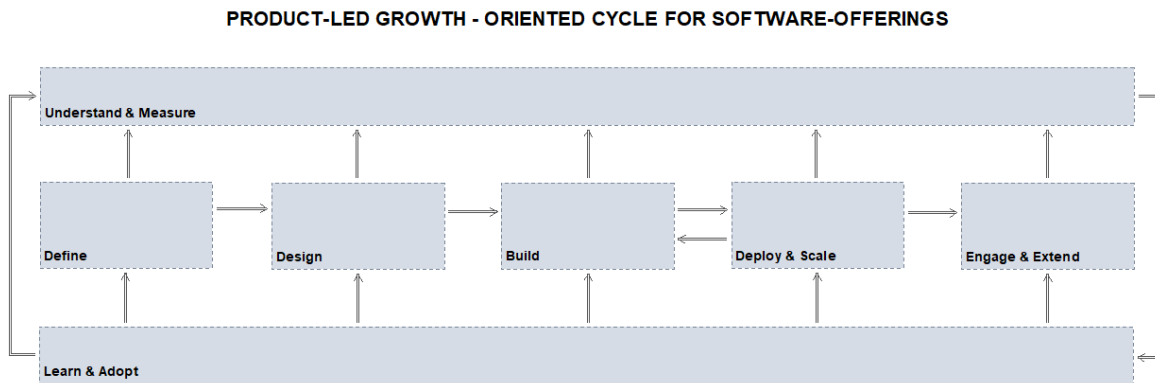


Abbildung 16: Metamodellbasiertes Vorgehensmodell „PLG-oriented cycle for software-offerings“ (Eigendarstellung)

Das Modell lässt sich in die Phasen *Define* (deutsch: definieren), *Design* (deutsch: gestalten), *Build* (deutsch: erstellen), *Deploy & Scale* (deutsch: bereitstellen und skalieren), *Engage & Extend* (deutsch: beschäftigen und erweitern), *Understand & Measure* (deutsch: verstehen und messen) und *Learn & Adopt* (deutsch: lernen und

anpassen) gliedern. Diese Phasen sind in Tabelle 12 nochmals mit Nummerierung aufgelistet.

Nummerierung der Phasen	Bezeichnung der Phasen
Phase 1	Define
Phase 2	Design
Phase 3	Build
Phase 4	Deploy & Scale
Phase 5	Engage & Extend
Phase 6	Understand & Measure
Phase 7	Learn & Adopt

Tabelle 12: Phasen des metamodellbasierten Vorgehensmodells

Während die Phasen 1 bis 5 grundsätzlich sequentiell aufeinander aufbauend sind (Phase 3 und Phase 4 erfolgen abwechselnd), werden die Phasen 6 und 7 als sogenannte Querschnittsphasen betrachtet. Diese Querschnittsphasen erstrecken sich über alle Phasen von 1 bis 5 hinweg. Ebenfalls in Abbildung 16 ersichtlich ist, dass alle sieben Phasen durch Konnektoren (gekennzeichnet durch Pfeile) miteinander in Verbindung stehen und als Sequenz-, Informations- oder Datenflüsse angesehen werden können. Durch die spezielle Anordnung der Phasen und deren spezifischen Zusammenhänge ergibt sich ein Kreislauf fortlaufenden Testens, Validierens und Anpassens.

Nachdem die Bestandteile des Kernmodells im Groben aufgeschlüsselt sowie deren Verbindungen erklärt wurden, wird im Folgenden auf jede einzelne Phase nochmals eingegangen.

Phase 1 - Define: Der Product-led Growth-oriented cycle for software-offerings beginnt mit der Phase des Definierens. Innerhalb dieser Phase gilt es, ein konkretes Bedürfnis oder eine Problemstellung am Markt zu identifizieren, daraufhin eine Lösung im Sinne eines Software-Angebots zu finden sowie eine Konkretisierung der Angebots-Idee vorzunehmen. Nach Ausarbeitung und Erfassung des Scopes des Software-Angebots, wird in die Phase 2 übergegangen. Über Phase 6 (Understand & Measure) fließen etwaig entstandene Erkenntnisse in Phase 7 (Learn & Adopt) ein und können als Gelerntes im Sinne von Verbesserungen in Phase 1 wiederum übernommen werden.

Phase 2 - Design: Die Phase 2 handelt von der Ausgestaltung des in Phase 1 definierten Software-Angebots. Vorwegzunehmen ist, dass der Begriff Design nicht ausschließlich mit dem reinen äußeren Erscheinungsbild des Software-Angebots assoziiert werden soll. Beinhaltet sind ebenso die Ausgestaltung von Funktionalität,

die Ausgestaltung einer guten User Experience sowie die Generierung des konkreten Mehrwerts für die Nutzenden. Nähere Informationen dazu sind in Abschnitt 4.2 zu finden. So wie auch bei Phase 1 sind hier ebenfalls wieder die Phasen 6 (Understand & Measure) und 7 (Understand & Adopt) von Relevanz. Nach der Phase des Designs erfolgt die Phase 3 Build.

Phase 3 - Build: Mit Phase 3 wird die tatsächliche Umsetzung des in Phase 2 ausgestalteten Software-Angebots eingeleitet. Insbesondere geht es in dieser Phase um die kontinuierliche (Weiter-)Entwicklung, Integration, Orchestrierung und Testung des Software-Produkts. Auch hier findet ein fortlaufender Austausch mit den Phasen 6 und 7 sowie auch mit der anschließenden Phase 4 (Deploy & Scale) statt.

Phase 4 - Deploy & Scale: Wie die Bezeichnung der Phase 4 bereits verdeutlicht, geht es in dieser um die Bereitstellung und Skalierbarkeit des Software-Angebots. An dieser Stelle gilt es die beidseitige Austauschbeziehung zwischen Phase 3 und Phase 4 hervorzuheben. Dies bedeutet, dass die Phase Deploy & Scale nicht nur Informationen und oder Daten von Phase 3 erhält, sondern auch direkt Botschaften an Phase 3 abgibt. Verbindungen bestehen zudem auch wieder zu den Phasen 6 und 7. Nach Phase 4 erfolgt die darauffolgende Phase 5 (Engage & Extend).

Phase 5 - Engage & Extend: Im Fokus dieser Phase steht die kontinuierliche Einbindung und Beschäftigung der Nutzenden sowie die stetige Erweiterung und Intensivierung der Produktnutzung. Die Phase Engage & Extend ist ebenfalls wie die bisherigen Phasen geprägt von einem Austausch mit Phase 6 und Phase 7.

Phase 6 - Understand & Measure: Wie bereits erläutert, erstreckt sich die Phase 6 über alle anderen Phasen hinweg. Ihr Hauptzweck ist es, jede der anderen sechs Phasen laufend zu analysieren, das Untersuchte zu verstehen und daraus abgeleitete Erkenntnisse direkt an Phase 7 (Learn & Adopt) weiterzugeben.

Phase 7 - Learn & Adopt: Die in Phase 6 gefundenen Erkenntnisse fließen schlussendlich in die Phase 7 ein, bei der im Vordergrund das Lernen aus den zuvor erhaltenen Informationen und Daten steht und aus welchen sich etwaige notwendige Anpassungen in Bezug auf das Software-Angebot ergeben. Diese gilt es folglich aufzugreifen und wiederum in der relevanten Phase umzusetzen.

Zusammenfassend lässt sich sagen, dass das metamodellbasierte Vorgehensmodell durch einen Prozess unentwegten Aufnehmens und Abgebens von Informationen und Daten sowie der daraus resultierenden Erkenntnisse zu etwaigen Anpassungen gezeugt ist. Weiters wurde festgestellt, dass sich aus dem Modell drei zentrale Validierungszyklen ableiten lassen – die angebotsorientierte Validierung, die umsetzungsorientierte Validierung und die betriebsorientierte Validierung. Im Folgenden wird auf diese drei Validierungszyklen näher eingegangen.

Angebotsorientierte Validierung

Der angebotsorientierte Validierungszyklus erschließt sich aus den Phasen 1 (Define), 6 (Understand & Measure) und 7 (Learn & Adopt). Mittels angebotsorientierter Validierung wird ersichtlich, ob an der grundlegenden Idee des Software-Angebots, sprich in der Phase 1 (Define), etwas angepasst oder abgeändert werden muss. Ein Beispiel dafür wäre die Erkenntnis, dass das Software-Angebot seitens der Nutzenden nicht als Lösung für ihr Bedürfnis oder Problem angesehen wird oder, dass die Nutzenden keinen konkreten Mehrwert für sich erkennen. Folglich muss das Konzept zum Software-Angebot von Grund auf neu überdacht und adaptiert werden. Der angebotsorientierte Validierungszyklus ist in Abbildung 17 bildlich dargestellt.

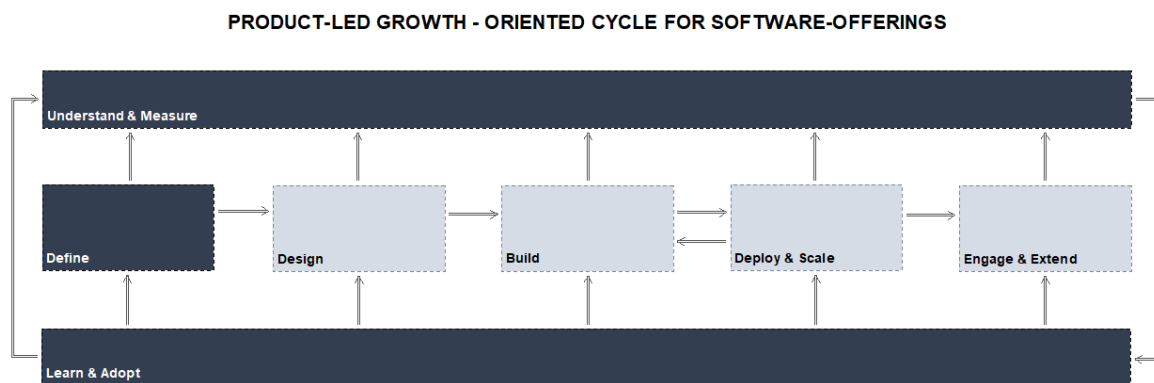


Abbildung 17: Angebotsorientierte Validierungszyklus (Eigendarstellung)

Umsetzungsorientierte Validierung

Bei der umsetzungsorientierten Validierung finden die Phasen 2 (Design), 6 (Understand & Measure), 5 (Engage & Extend) und 7 (Learn & Adopt) Berücksichtigung. Konkret geht es bei diesem Validierungszyklus um gewonnene Erkenntnisse in Bezug auf etwaig notwendiger Anpassungen hinsichtlich der Ausgestaltung des Software-Angebots. Der Bedarf einer Adaption des Software-Produkts erschließt sich aus Phase 5. Wird durch diese Phase erkenntlich, dass das Software-Angebot beispielsweise durch Personen nicht ausreichend genutzt wird, könnte dies ein Indiz für eine ungenügende User Experience sein. Folglich sollte dies analysiert werden, um anschließend in der Design-Phase Anpassungen vornehmen zu können. Der umsetzungsorientierte Validierungszyklus ist in Abbildung 18 ersichtlich.

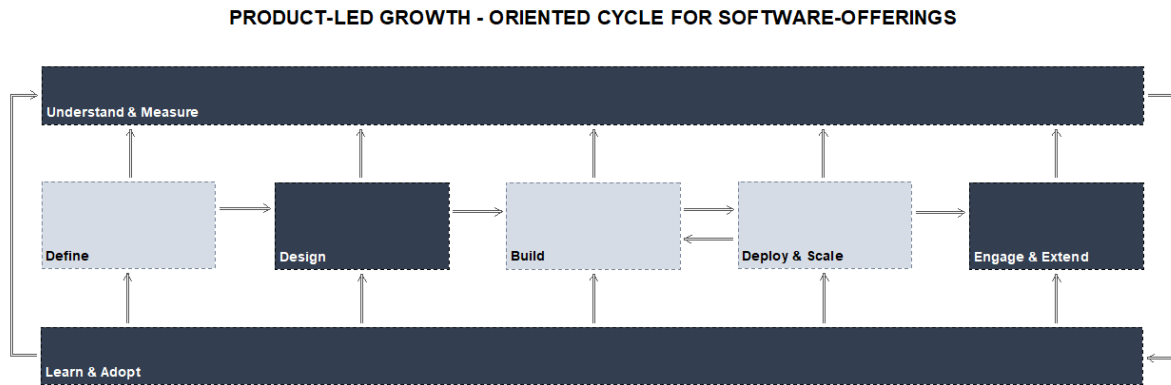


Abbildung 18: Umsetzungsorientierte Validierung (Eigendarstellung)

Betriebsorientierte Validierung

Der betriebsorientierte Validierungszyklus erschließt sich aus den Phasen 3 (Build), 4 (Deploy & Scale), 6 (Understand & Measure) und 7 (Learn & Adopt). Wie erwähnt findet zwischen Phase 3 und 4 ein bidirektionaler Austausch von Informationen und oder Daten statt. Sollten in Phase 4 Probleme hinsichtlich der Bereitstellung oder Skalierbarkeit des Software-Angebots auftreten, so wird das direkt an Phase 3 rückgemeldet und zeigt, dass etwas innerhalb dieser Phase angepasst werden muss. Der betriebsorientierte Validierungszyklus ist zur Veranschaulichung in Abbildung 19 dargestellt.

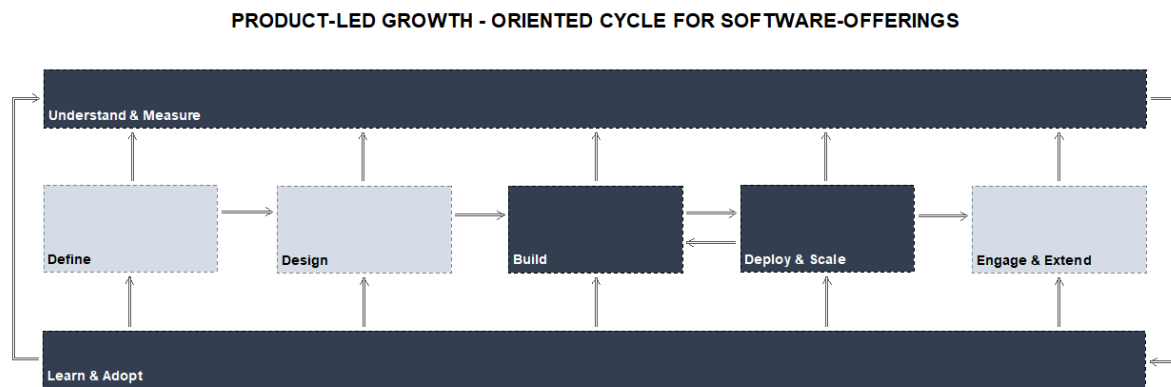


Abbildung 19: Betriebsorientierte Validierung (Eigendarstellung)

Das metamodellbasierte Vorgehensmodell wird im folgenden Abschnitt 4.2 um phasenspezifische Zielsetzungen erweitert, die unter Berücksichtigung des Product-led Growth Konzepts erreicht werden sollten.

4.2 Erweiterung des Modells um phasenspezifische Zielsetzungen

In diesem Abschnitt 4.2 wird das bereits vorgestellte Vorgehensmodell um phasenspezifische Zielsetzungen erweitert und auf jede einzelne Phase nochmals näher eingegangen. Zudem werden in diesem Abschnitt bereits hilfreiche Konzepte, die positiv zur Zielerreichung beitragen, genannt, wobei in Abschnitt 4.3 eine detaillierte Erklärung zu finden ist. Eine Darstellung des Kernmodells, erweitert um konkrete Zielsetzungen ist in Abbildung 20 gegeben (eine größere Darstellung findet sich im Anhang).

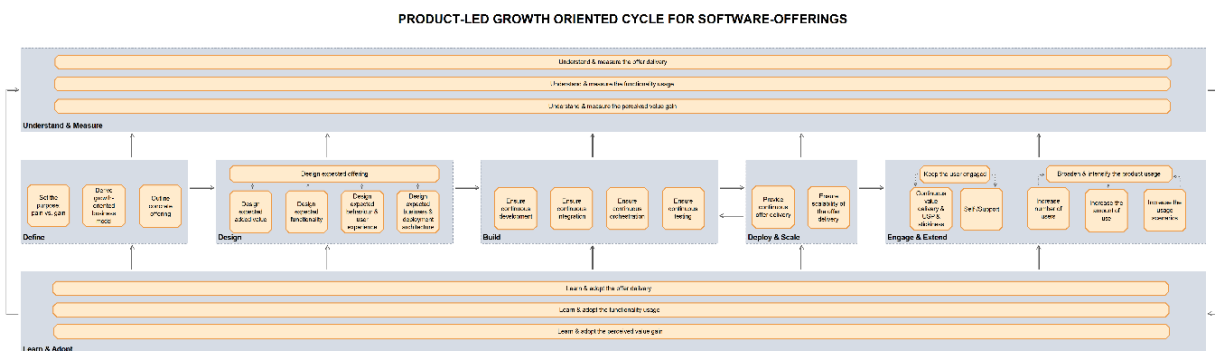


Abbildung 20: Erweiterung des Vorgehensmodells um phasenspezifische Zielsetzungen (Eigendarstellung)

Die Vorgabe phasenspezifischer Zielsetzungen sind für Nutzende des Modells von Bedeutung, da sie eine Orientierung bieten, auf was unter Anwendung des Product-led Growth Konzepts zu beachten gilt. Durch diese wird Anwender:innen auf einem Blick erkenntlich gemacht, was in welcher der Phasen relevant ist und auf was im Speziellen fokussiert werden soll.

Phase 1 – Define: Wie bereits in Abschnitt 4.1 kurz erläutert, findet in der Phase Define die Ideengenerierung sowie die Spezifikation des Software-Angebots statt. Für diese Phase konnten drei konkrete Zielsetzungen eruiert werden, welche in Abbildung 21 dargestellt sind. Sie ergeben sich aus „*Set the purpose: pain vs. gain*“ (deutsch: festlegen des Zwecks: Schmerz vs. Nutzen), „*Derive growth-oriented business model*“ (deutsch: Ableitung eines wachstumsorientierten Geschäftsmodells) und „*Outline concrete offering*“ (deutsch: konkretes Angebot skizzieren).

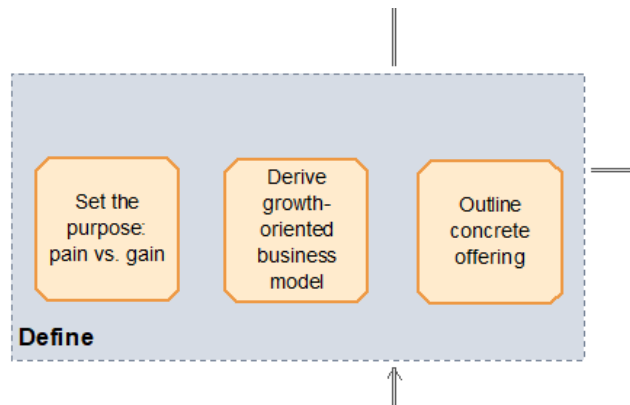


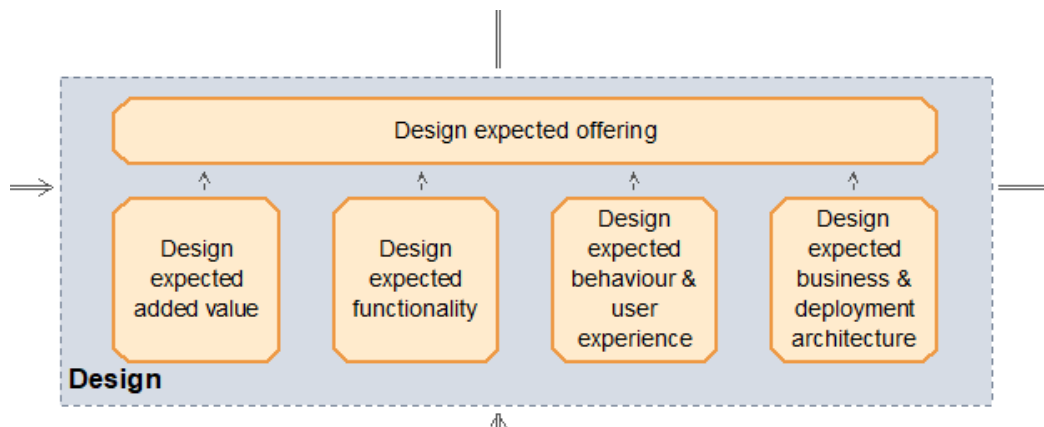
Abbildung 21: Phase 1 (Define) erweitert um Zielsetzungen (Eigendarstellung)

Aus der ersten der drei Zielsetzungen geht hervor, dass zunächst ein spezifisches Bedürfnis seitens der Nutzer:innen aufgedeckt werden muss, um überhaupt ein Software-Angebot als Lösung entwickeln und anbieten zu können. Bei diesem Bedürfnis oder auch der Problemfeststellung einer Nutzer:innengruppe wird häufig von dem sogenannten Pain gesprochen. Für diesen Pain gilt es folglich ein Software-Angebot als Lösung zu entwickeln. Dabei ist es von Bedeutung, dass die gefundene Lösung einen eindeutigen Mehrwert (auch Gain genannt) potenziellen Nutzenden bietet und dieser auch eindeutig erkenntlich ist. Die erste Zielsetzung ist aus Sicht von Product-led Growth insbesondere deshalb so bedeutend, da es ohne Pain auch keinen Gain geben kann und somit ohne dem Vorhandensein eines Bedarfs auch keine dazugehörige Lösung benötigt wird (Appcues, 2021).

Wurde ein konkretes Bedürfnis und eine korrespondierende Lösung identifiziert, gilt es in einem weiteren Schritt ein wachstumsorientiertes Geschäftsmodell zu erstellen. Das Konstrukt des Geschäftsmodells dient als ganzheitliche Beschreibung und Verknüpfung der wichtigsten Geschäftsprozesse eines Unternehmens (Foss & Saebi, 2017, S. 202). Sie stellen eine Art Architektur dar, aus der sich ableiten lässt, auf welche Art und Weise ein Unternehmen Werte für seine Stakeholder schafft (Carvalho et al., 2020, S. 504; Metelskaia et al., 2018, S. 35; Teece, 2018, S. 40). Osterwalder et al. (2005, S. 2) beschreiben Geschäftsmodelle als den „Bauplan der Unternehmensführung“. Strategische Belange wie Positionierung und Ziele werden in ein konzeptionelles Modell überführt, welches explizit aufweist, wie das Unternehmen funktioniert. Ein Geschäftsmodell erlaubt es die Unternehmensstruktur und seine Systeme, welche die operative und physische Form eines Unternehmens ausmachen darzustellen (Osterwalder et al., 2005, S. 2). Shepherd & Gruber (2021a, S. 973) schreiben in ihrer Arbeit, dass auf Grundlage von Geschäftsmodellen, Hypothesen zur Gründung und zum Wachstum von Unternehmen aufgestellt werden können, um zu ermitteln, ob ein Bedürfnis durch ein Produkt oder Dienstleistung einerseits gelöst werden kann beziehungsweise einen Mehrwert bringt. Da auf Basis von Business Modellen strategische Entscheidungen getroffen werden, tragen sie maßgeblich zum

Unternehmenserfolg bei (Metelskaia et al., 2018, S. 35). Häufig zum Einsatz kommende Geschäftsmodelle zur Konkretisierung des Mehrwerts sind das *Business Model Canvas* (siehe Abschnitt 4.3.2) oder auch das *Value Proposition Canvas* (Abschnitt 4.3.3) von Osterwalder. Ein weiteres Geschäftsmodell, welches im Sinne von Product-led Growth Berücksichtigung findet ist das Abonnementmodell Freemium (siehe Abschnitt 4.3.4). Eine genauere Erläuterung zu diesen Geschäftsmodellen sowie auch zu den Konzepten des *Lean Startups*, der Erstellung von Personas (siehe Abschnitt 4.3.5) oder auch des Customer Journey Mapping (siehe Abschnitt 4.3.6), welche ebenfalls positiv zur Zielerreichung innerhalb dieser Phase beitragen können, ist in Kapitel 4.3 zu finden. Die dritte Zielsetzung ergibt sich schlussendlich aus der Ausgestaltung des konkreten Software-Angebots.

Phase 2 – Design: In der Phase des Designs geht es um Zielsetzungen wie „*Design expected added value*“ (deutsch: Gestaltung des erwarteten Mehrwerts), „*Design expected functionality*“ (deutsch: Gestaltung der erwarteten Funktionalität), „*Design expected behaviour & user experience*“ (deutsch: Gestaltung des erwarteten Verhaltens und der Benutzererfahrung) und „*Design expected business & deployment architecture*“ (deutsch: Gestaltung der erwarteten Geschäfts- und Bereitstellungsarchitektur). Generell gesprochen geht es um die Ausgestaltung des in Phase 1 skizzierten Software-Angebots. Die Zielsetzungen sind in Abbildung 22 ersichtlich.



**Abbildung 22: Phase 2 (Design) erweitert um konkrete Zielsetzungen
(Eigendarstellung)**

Die erste Zielsetzung innerhalb dieser Phase zielt auf die Ausgestaltung des konkreten erwarteten Mehrwerts des Software-Angebots ab. Wie auch schon in Kapitel 2.1 erwähnt, ist es von besonderer Bedeutung, dass das Software-Angebot den Nutzenden einen eindeutigen Mehrwert bietet, welcher einerseits sofort erkenntlich ist und andererseits als relevant empfunden wird. Durch diesen wahrgenommenen Nutzen können sich Unternehmen von ihrer Konkurrenz abgrenzen und abheben (Lindič & Da Marques Silva, 2011, S. 1694). Eine weitere Zielsetzung ergibt sich aus

der Ausgestaltung der erwarteten Funktionalität des Software-Angebots. Dabei sollte der Fokus insbesondere auf eine benutzerfreundliche User Interface (siehe Abschnitt 4.3.8) gerichtet werden. Auch die User Experience sollte in dieser Phase bereits Berücksichtigung finden. Für eine positive User Experience ist es von Bedeutung jegliche Art von Reibung zu vermeiden, mit dem Ziel ein einzigartiges Produkterlebnis zu schaffen. Konzepte wie Minimum Viable Product machen es möglich, rasch Feedback seitens der Nutzer:innen einzuholen um anschließend umgehend Anpassungen und Verbesserungen vornehmen zu können. An dieser Stelle sind auch die Vorteile von *Cloud-Native* (siehe Abschnitt 4.3.10) sowie die einer *Microservices*-Architektur (siehe Abschnitt 4.3.11) zu nennen, da diese positiv zur Zielerreichung beitragen. Eine Microservice-Architektur verhindert aufgrund der losen Kopplung der Microservices, dass beispielsweise die Software bei Updates nicht zur Verfügung steht. Zudem ist für Entwicklungsteams möglich, schnell neue Anwendungskomponenten zu bauen. Letztere Zielsetzung fokussiert auf die zum Einsatz kommenden Prozesse, Hardware und weitere Technologien, die für die spätere Bereitstellung des Software-Angebots notwendig sind (Waseem et al., 2021).

Phase 3 – Build: In der Phase Build geht es um die Umsetzung des in Phase 2 ausgestalteten Software-Angebots. Bezogen auf das Software-Offering finden die in Abbildung 23 dargestellten vier Zielsetzungen Berücksichtigung: „*Ensure continuous development*“ (deutsch: kontinuierliche Entwicklung sicherstellen), „*Ensure continuous integration*“ (deutsch: kontinuierliche Integration sicherstellen), „*Ensure continuous orchestration*“ (deutsch: kontinuierliche Orchestrierung sicherstellen) und „*Ensure continuous testing*“ (deutsch: kontinuierliche Überprüfung sicherstellen).

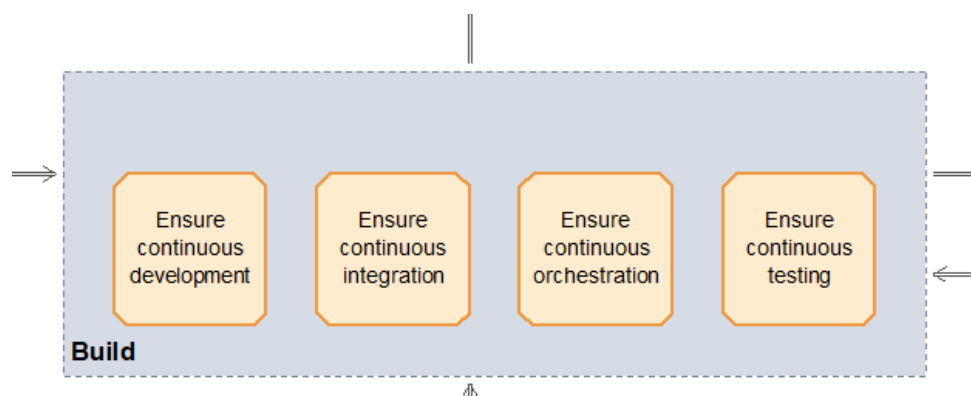


Abbildung 23: Phase 3 (Build) erweitert um konkrete Zielsetzungen (Eigendarstellung)

Im Fokus dieser Zielsetzungen steht das Adjektiv „kontinuierlich“, welches auf die zunehmende Notwendigkeit verweist, einen durchgängigen Fluss zwischen Nutzer:innen- und Kunden- oder Kundinnennachfrage und der schnellen Lieferung eines Produkts oder Dienstleistung herzustellen (Fitzgerald & Stol, 2017, S. 185). Es geht also um eine möglichst schnelle und fortlaufende Entwicklung, Integration sowie auch Überprüfung des Software-Angebots, mit dem Ziel einer zeitnahen Auslieferung

an die Nutzer:innen. Um das zu ermöglichen, verlangt es nach Konzepten wie Agile Softwareentwicklung (siehe Abschnitt 4.3.12) oder DevOps (siehe Abschnitt 4.3.13).

Phase 4 - Deploy & Scale: Innerhalb dieser Phase konnten zwei Zielsetzungen identifiziert werden, welche in Abbildung 24 dargestellt sind. Erstere ergibt sich aus „*Provide continuous offer delivery*“ (deutsch: kontinuierliche Auslieferung von Angeboten). Zweitere ist „*Ensure scalability of offer delivery*“ (deutsch: Gewährleistung der Skalierbarkeit des Angebots).

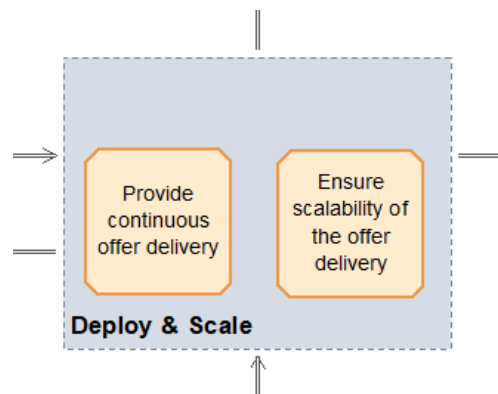


Abbildung 24: Phase 4 (Deploy & Scale) erweitert um konkrete Zielsetzungen (Eigendarstellung)

Bei „Continuous delivery“ handelt es sich um eine Softwareentwicklungsmethode, die kurze Entwicklungszyklen ermöglicht und die rasche Auslieferung von Software-Updates an die Nutzer:innen sicherstellt. Skalierbarkeit meint wiederum die Fähigkeit einer Software zu wachsen, um so eine erhöhte Arbeitslast bewältigen zu können (Fitzgerald & Stol, 2017, S. 183).

Phase 5 - Engage & Extend: In der Phase Engage & Extend sind einerseits Zielsetzungen wie „*Keep the user engaged*“ (deutsch: den oder die Nutzer:in beschäftigt halten), „*Continuous value delivery & USP & Stickiness*“ (deutsch: kontinuierliche Mehrwertlieferung & USP (*Unique Selling Proposition*; deutsch: Alleinstellungsmerkmal) & Klebrigkeit/Haftung) , „*Self-/Support*“ (deutsch: Selbst-/Support) sowie auch „*Broaden & intensify the product usage*“ (deutsch: Produktnutzung erweitern und intensivieren) , „*Increase number of users*“ (deutsch: Erhöhung der Zahl der Nutzer:innen), „*Increase the amount of use*“ (deutsch: den Umfang der Nutzung erhöhen) und „*Increase the usage scenarios*“ (deutsch: Nutzungsszenarien erweitern) zu finden. In Abbildung 25 werden diese Zielsetzungen nochmals aufgezeigt.

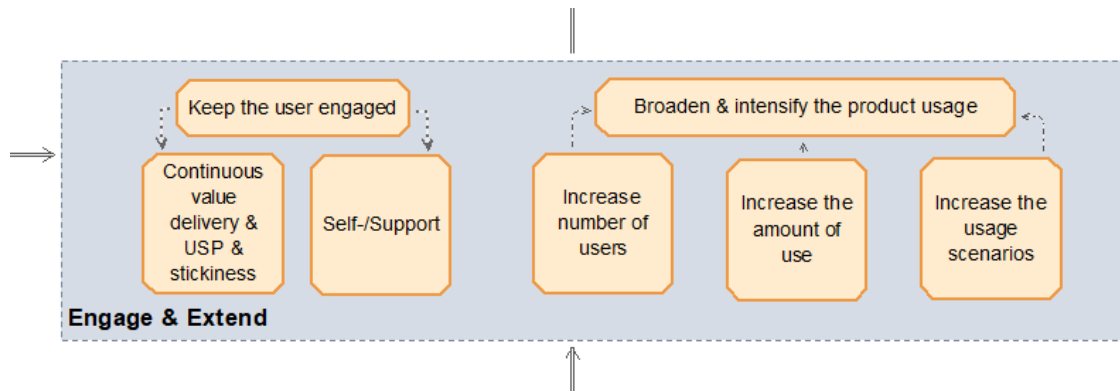


Abbildung 25: Phase 5 (Engage & Extend) erweitert um konkrete Zielsetzungen (Eigendarstellung)

Wie bereits erläutert, geht es in dieser Phase darum, die Nutzer:innen an das Produkt zu binden und sie fortlaufend darin zu motivieren sich mit dem Produkt zu beschäftigen und mit diesem zu interagieren. Es ist daher wichtig dem oder der Nutzer:in durchgehend den Wert und den USP des Software-Angebots zu vermitteln.

Aus Sicht von Product-led Growth ergibt sich eine weitere wichtige Zielsetzung aus dem Self-/Support. Wie auch schon zuvor angesprochen, wollen die heutzutage technikaffinen User:innen einerseits Produkte eigenständig erforschen sowie auch für etwaige auftretende Probleme rasch eine Lösung finden und nicht auf eine Problembeseitigung seitens des Unternehmens warten müssen. Diese Art von Reibung könnte nämlich wiederum negative Auswirkungen auf die in der Phase 2 schon angesprochene User Experience haben und unmittelbar zu Frustration führen. Dennoch ist zu sagen, dass das Angebot eines persönlichen Supports durch das Unternehmen nicht vernachlässigt werden darf, falls Nutzer:innen doch etwaige Unterstützung benötigen. Insgesamt geht also um einen umfassenden Self-/Support, der den Nutzenden stets zur Verfügung steht.

Die nächsten Zielsetzungen beziehen sich auf die Ausweitung und Intensivierung der Produktnutzung sowie auch die Erhöhung des Umfangs der Nutzung. In der Literatur wird häufig das Konzept Stickiness hervorgehoben. Grundsätzlich kann dieses Konzept aus zwei verschiedenen Blickwinkeln betrachtet werden. Einerseits wird unter Stickiness die Intention der Nutzer:innen verstanden, eine Website oder ein Produkt immer wieder zu besuchen, dahin zurückzukehren und zu benutzen. Andererseits bezieht sich Stickiness auf die Verweildauer, sprich die Zeit die Nutzende bei der Interaktion mit einer Website oder auch einem Produkt verbringen (Friedrich et al., 2019, S. 4). Ziel sollte es ebenfalls sein, laufend die Anzahl der Nutzenden zu steigern. Beim Product-led Growth Ansatz wird davon ausgegangen, dass sich die Anzahl der Nutzer:innen insbesondere durch Viralität erhöht. Viralität meint, dass Nutzer:innen das Produkt so sehr wertschätzen, dass es via Mund-zu-Mund Propaganda rasch Verbreitung am Markt findet. Bei der letzten Zielsetzung innerhalb dieser Phase geht

es um die Erweiterung der Nutzungsszenarien und folglich die Ausweitung und den Ausbau des Angebots.

Phase 6 - Understand & Measure: In der Phase Understand & Measure konnten drei relevante Zielsetzungen identifiziert werden. Diese ergeben sich aus „*Understand & measure the offer delivery*“ (deutsch: Verstehen und Messen der Angebotslieferung), „*Understand & measure the functionality usage*“ (deutsch: Verstehen und Messen der Funktionsnutzung) sowie „*Understand & measure the perceived value gain*“ (deutsch: Verstehen und Messen des wahrgenommenen Mehrwerts) und sind in Abbildung 26 abgebildet.

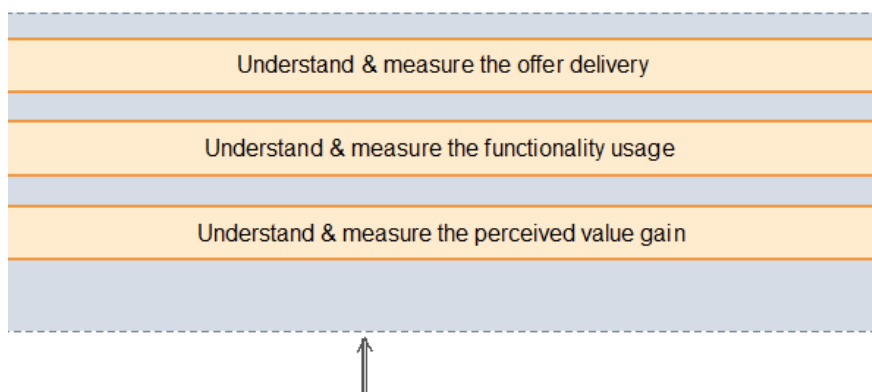


Abbildung 26: Phase 6 (Understand & Measure) erweitert um konkrete Zielsetzungen (Eigendarstellung)

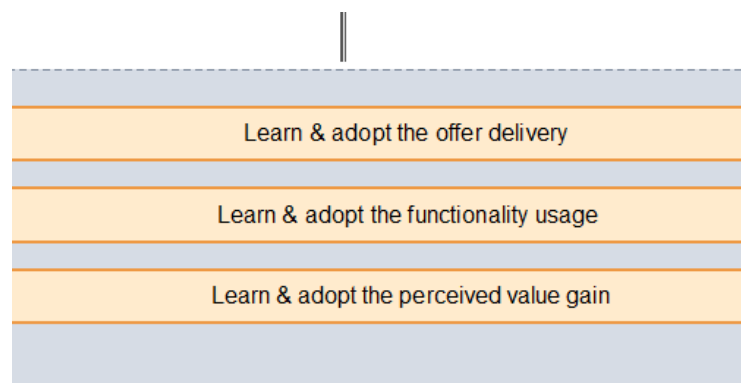
Wie schon in Abschnitt 4.1 erläutert, ist es von Bedeutung jede der anderen Phasen laufend zu messen und zu überprüfen, um davon anschließend zu lernen und gegebenenfalls Anpassungen sowie auch Verbesserungen vornehmen zu können. Die erste Zielsetzung ergibt sich aus dem Verstehen und Messen der Angebotsauslieferung, welches auf die laufende Überprüfung der korrekten Funktionsweise und gewünschten Leistung des Software-Angebots bezogen ist.

Die zweite Zielsetzung ergibt sich aus dem Verstehen und Messen der Funktionsnutzung. Im Vordergrund steht hier beispielsweise die Benutzerfreundlichkeit, die essentiell zu einer positiven User Experience (siehe Abschnitt 4.3.7) beiträgt.

Die dritte Zielsetzung fokussiert auf das Verstehen und Messen des wahrgenommenen Mehrwerts durch die Nutzer:innen. Im Speziellen geht es darum festzustellen, ob den Nutzern oder Nutzerinnen der Mehrwert des Software-Angebot bewusst ist und sie diesen als solchen erkennen. Diese Phase kann insbesondere durch Konzepte wie beispielsweise *User Tracking* (deutsch: Nutzer:innen Nachverfolgung) oder auch *Voice of the Customer* (kurz VOC) (siehe Abschnitt 4.3.14) unterstützt werden. Unter

User Tracking wird die Sammlung von Daten über das Nutzer:innenverhalten verstanden, welches maßgeblich zur Verbesserung der Benutzeroberfläche beitragen kann. Hervorzuheben gilt, dass das Tracking des Verhaltens grundsätzlich anonym geschieht und nicht in dem Sinne verstanden werden soll, dass jeder oder jede Nutzer:in durchleutet wird. Ein weitere Methode, welche in dieser Phase zur Anwendung kommen kann, um die Zielerreichung zu unterstützen ist eben das zuvor genannte Konzept Voice of the customer.

Phase 7 - Learn & Adopt: Die Phase Learn & Adopt beinhaltet, wie auch in Abbildung 27 ersichtlich, drei konkrete Zielsetzungen. Sie ergeben sich aus „*Learn & adopt the offer delivery*“ (deutsch: Lernen & Anpassen der Angebotslieferung), „*Learn & adopt the functionality usage*“ (deutsch: Lernen & Anpassen der Funktionsnutzung), „*Learn & Adopt the perceived value*“ (deutsch: Lernen und Anpassen des wahrgenommenen Mehrwerts).



**Abbildung 27: Phase 7 (Learn & Adopt) erweitert um konkrete Zielsetzungen
(Eigendarstellung)**

Wie bereits erläutert, geht es in Phase 7 darum, aus dem Gemessenen von Phase 6 zu lernen und wenn notwendig etwaige Anpassungen sowie auch Verbesserungen am Produkt vorzunehmen. Hier bieten beispielsweise Produkt-Roadmaps einen guten Anhaltspunkt, um aufzuzeigen, wie sich das Software-Angebot über die Zeit hinweg entwickelt.

Nachdem das Kernmodell mitsamt der identifizierten Zielsetzungen erläutert wurde, wird im folgenden Abschnitt 4.3 auf ausgewählte Konzepte eingegangen, die positiv zur Zielerreichung beitragen.

4.3 Erweiterung des Modells um ausgewählte Konzepte

Abschnitt 4.3 gibt das in 4.2 vorgestellte metamodellbasierte Vorgehensmodell inklusive Zielsetzungen wieder, erweitert um ausgewählte Konzepte, die bei der

Erreichung der identifizierten Zielsetzungen unterstützen sollen. In der folgenden Tabelle 13 ist eine Auflistung der Konzepte zu finden, auf die im Rahmen dieses Abschnittes näher eingegangen wird.

4.3.1 Lean Startup	4.3.8 User Interface Design
4.3.2 Business Model Canvas	4.3.9 Minimum Viable Product
4.3.3 Value Proposition Canvas	4.3.10 Cloud-Native
4.3.4 Business Model Freemium	4.3.11 Microservices
4.3.5 Personas	4.3.12 Agile Softwareentwicklung
4.3.6 Customer Journey Mapping	4.3.13 DevOps
4.3.7 User Experience Design	4.3.14 Voice of the Customer

Tabelle 13: Übersicht ausgewählter Konzepte

An dieser Stelle gilt erneut zu betonen, dass es nicht Ziel dieser Arbeit ist, jedes Konzept eingehend zu behandeln und zu erklären. Der Abschnitt soll lediglich einen Einblick geben, welche Konzepte mit dem Product-led Growth Ansatz zusammenspielen. Die Darstellung des metamodellbasierten Vorgehensmodells, erweitert um ausgewählte Konzepte ist dem Anhang zu entnehmen. Abschnitt 4.3.15 beinhaltet darüber hinaus eine Matrix, in der auf den ersten Blick ersichtlich ist, welche Konzepte in der jeweiligen Phase des Vorgehensmodells (siehe Abschnitt 4.1) relevant sind.

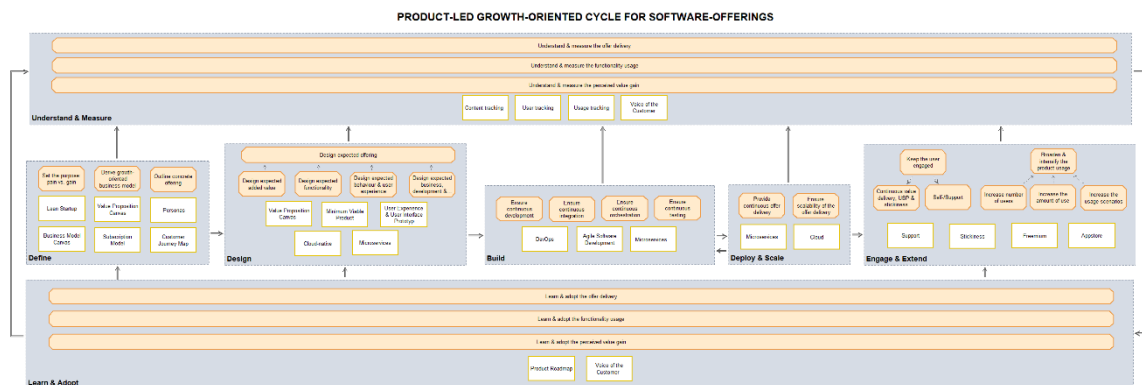


Abbildung 28: Metamodellbasiertes Vorgehensmodell erweitert um ausgewählte Konzepte (Eigendarstellung)

4.3.1 Lean Startup

Das Konzept des *Lean Startups* wurde ursprünglich entwickelt, um das Risiko des Misserfolgs bei Firmengründung zu reduzieren. Hintergrund ist, dass Startups bei der Perfektionierung einer Produkt- oder Dienstleistungsidee zu viel Zeit, Aufwand und vor

allem finanzielle Ressourcen investierten, ohne dabei die Bedürfnisse potenzieller Kunden oder Kundinnen in den Vordergrund zu stellen oder sogar generell zu berücksichtigen. Das Abzielen auf die Bedürfnisbefriedigung potenzieller Kunden oder Kundinnen und Nutzer:innen ist jedoch maßgeblich ausschlaggebend für den Erfolg eines Produkts oder Dienstleistung und somit für die Erwirtschaftung von Einnahmen von hoher Priorität. Das Lean Startup Konzept schlägt daher vor, eine nach außen hin gerichtete Lernhaltung einzunehmen und den oder die Kunden oder Kundin und Nutzer:in in den Entwicklungsprozess einer Startup-Idee zu involvieren. Unternehmen sollen im Zuge dessen laufend Hypothesen über ihre Startup-Idee aufstellen, diese testen und anschließend etwaige Anpassungen beim ursprünglichen Konzept vornehmen. Dieser Prozess der Validierung wird so lange wiederholt, bis ein tragfähiges Geschäftsmodell gefunden wurde. Innerhalb des Lean Startups wird dabei auch von der sogenannten *Build-Measure-Learn Feedback Loop* (deutsch: Entwickeln-Messen-Lernen Feedback Schleife) gesprochen (Shepherd & Gruber, 2021a, S. 968–969). Zur besseren Veranschaulichung ist diese in der nachfolgenden Abbildung 29 dargestellt.

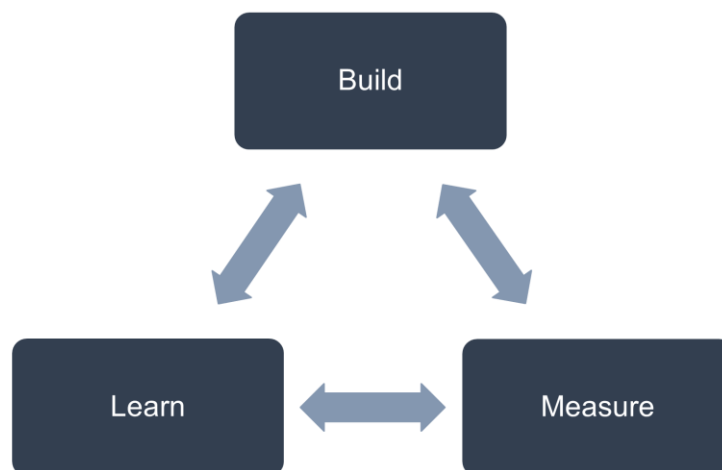


Abbildung 29: Build-Measure-Learn Feedback Loop (in Anlehnung an (Shepherd & Gruber, 2021a))

Das Lean Startup Framework setzt sich aus fünf primären Bausteinen zusammen, welche sich aus dem Finden von Marktchancen, dem Entwurf von Geschäftsmodellen, validiertes Lernen, der Entwicklung von Minimum Viable Products sowie die sich daraus ergebende Erkenntnis, ob etwaige Anpassungen an dem ursprünglichen Konzept vorgenommen werden müssen, ergeben (Shepherd & Gruber, 2021a, S. 970).

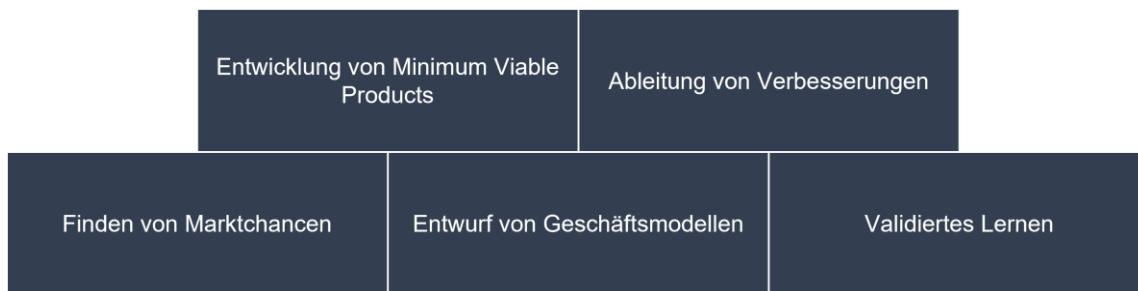


Abbildung 30: Bausteine des Lean-Startup Frameworks (in Anlehnung an (Shepherd & Gruber, 2021a, S. 970))

Der erste Baustein des Lean Startup Konzepts rückt das Finden von Marktchancen in den Mittelpunkt der Betrachtung und bezieht sich auf die konkrete Domäne, in der ein Unternehmen konkurrieren, für Kunden oder Kundinnen Wert schaffen und existenzfähig werden möchte (Shepherd & Gruber, 2021a, S. 971). Der zweite Baustein stellt einen bedeutenden Meilenstein für Unternehmen dar, denn er richtet sich auf die Entwicklung eines Geschäftsmodells, welches dazu dient, aufzuzeigen wie die konkrete Umsetzung der Startup-Idee erfolgen soll, um schlussendlich ein lebensfähiges Unternehmen schaffen zu können. Geschäftsmodelle können als ein Gerüst angesehen werden, auf dessen Grundlage Hypothesen zur Gründung und zum Wachstum von Unternehmen aufgestellt werden können, mit dem Ziel, herauszufinden, ob ein bestimmtes Bedürfnis oder Problem einer Zielgruppe durch das Produkt oder die Dienstleistung gedeckt, beziehungsweise gelöst werden kann und insbesondere mehrwertstiftend ist. Zwei häufig empfohlene Geschäftsmodelle, die den Kunden oder Kundinnen und den Nutzen der Geschäftsidee in den Vordergrund stellen, sind das Business Model Canvas (siehe Kapitel 4.3.2) und das Value Proposition Canvas (siehe Abschnitt 4.3.3) (Shepherd & Gruber, 2021a, S. 973). Der nächste Baustein ist das validierte Lernen, welches besagt, dass die auf dem Geschäftsmodell basierenden und zuvor aufgestellten Hypothesen getestet und validiert werden sollen (Shepherd & Gruber, 2021a, S. 976). Das Testen der Hypothesen kann unter Einbindung eines sogenannten Minimum Viable Products (vierter Baustein) erfolgen. Konkret handelt es sich dabei, um eine mit minimalem Aufwand und kurzer Entwicklungszeit geschaffene erste Version eines Produkts, die einmal die komplette Build-Measure-Learn Schleife durchläuft. Dieses Minimum Viable Product soll nur die relevantesten Merkmale eines geplanten Produkts beinhalten, da das möglichst schnelle Testen der Hypothesen und das schnelle Einholen von Feedback im Vordergrund stehen. Eine genauere Erklärung zu dem Minimum Viable Product ist in Abschnitt 4.3.9 zu finden (Shepherd & Gruber, 2021a, S. 978). Im letzten Baustein geht es darum, aus den getesteten Hypothesen zu lernen und wenn notwendig Anpassungen am ursprünglichen Konzept beziehungsweise an der generellen Startup-Idee vorzunehmen (Shepherd & Gruber, 2021a, S. 980).

4.3.2 Business Model Canvas

Ein im Software-Bereich häufig empfohlenes und zum Einsatz kommendes Geschäftsmodell ist das von Osterwalder entwickelte Business Model Canvas (kurz BMC). Im Gegensatz zum ursprünglichen Business Plan, steht beim Business Modell Canvas die Visualisierung im Vordergrund, welche darauf abzielt auf einen Blick erkenntlich zu machen, wie die Umsetzung einer Geschäftsidee geschehen soll (Klimanov & Tretyak, 2019, S. 125; Osterwalder et al., 2005). Da sich das Business Modell Canvas vor allem durch seine Einfachheit und Kompaktheit auszeichnet, findet es als intuitives Werkzeug große Beliebtheit im Startup-Bereich (Carvalho et al., 2020, S. 505).

Das Business Modell Canvas setzt sich konkret aus neun unterschiedlichen Bereichen zusammen, die wiederum in die vier übergeordneten Elemente Wertangebot, Architektur der Beziehung zwischen Unternehmen und Stakeholder, Tätigkeiten des Unternehmens und finanzielle Aspekte eingeordnet werden können (Carvalho et al., 2020, S. 505). Diese Eingliederung der neun Bausteine in die vier genannten Elemente ist in der folgenden Tabelle 14 nochmals ersichtlich.

Elemente	BMC
Wert-Angebot	Wert-Angebot
Architektur der Beziehung zwischen Unternehmen und Austauschpartnern	Schlüsselpartner, Kunden- oder Kundinnenbeziehung, Kunden- oder Kundinnensegment und Kanäle
Tätigkeiten des Unternehmens	Schlüsselaktivitäten und Schlüsselressourcen
Finanzielle Aspekte	Kostenstruktur und Ertrag

Tabelle 14: Einordnung der neun Bereiche des BMC in vier Elemente

Abbildung 31 zeigt den Aufbau des Business Model Canvas, mitsamt den neun Bausteinen, auf die nun folglich näher eingegangen wird.



Abbildung 31: Business Model Canvas (in Anlehnung an (Osterwalder & Pigneur, 2011))

Ausgangspunkt und Zentrum des Business Model Canvas ist der Baustein des Value Proposition (deutsch: Wert-Angebot), in welchem alle Produkte und Dienste aufgelistet werden sollen, die Werte für ein bestimmtes Kundensegment oder Kundinnensegment generieren (Carvalho et al., 2020, S. 505). In Bezug auf die Art der Auflistung des Wert-Angebots sind in der Literatur unterschiedliche Herangehensweisen zu finden. So treffen Metelskaia et al. (2018, S. 36) in ihrer Arbeit beispielsweise eine Unterscheidung zwischen quantitativen (Preis, Leistung, Marke) und qualitativem Wert-Angebot (Kundenzufriedenheit oder Kundinnenzufriedenheit, Design, Benutzer:innenfreundlichkeit). Der nächste bedeutende Baustein bezieht sich auf das Kundensegment oder Kundinnensegment, auf welches Unternehmen abzielen und mit ihrem Produkt oder Dienstleistung Wert, sprich einen konkreten Nutzen schaffen wollen. Dieser Nutzen muss in einem weiteren Schritt (Baustein Kanäle) über bestimmte Kanäle an das spezifische Kundensegment oder Kundinnensegment vermittelt und kommuniziert werden. Kanäle können einerseits Vertriebskanäle aber auch Marketingkanäle sein, direkter oder indirekter Art sowie online oder auch offline Kanäle (Carvalho et al., 2020, S. 505). Neben der Gewinnung neuer Kunden oder Kundinnen steht insbesondere auch der Aufbau langfristiger Beziehungen zu diesen im Vordergrund. Neben dem Beziehungsaufbau zu Kunden oder Kundinnen gilt es auch die Schaffung zuverlässiger Beziehungen zu diversen Schlüsselpartnern, wie beispielsweise Lieferunternehmen nicht zu vernachlässigen. Unter Schlüsselpartner werden all jene Stakeholder verstanden, die für das Fortbestehen eines Unternehmens unverzichtbar sind. Innerhalb des Bereichs der Schlüsselressourcen sollen jegliche Ressourcen aufgeschlüsselt werden, die für das Funktionieren des Geschäftsmodells relevant sind, wie beispielsweise menschliche, geistige, finanzielle oder auch

materielle Ressourcen. Ähnlich der Schlüsselressourcen beziehen sich Schlüssel-Aktivitäten ebenfalls auf Aktivitäten die für die Umsetzung des Geschäftsmodell notwendig sind. Der achte Baustein des Business Model Canvas zeigt die Einnahmenquellen, durch welche das Unternehmen seinen Umsatz generiert. Der letzte Baustein ist die Kostenstruktur, welcher die für das gesamte Geschäftsmodell anfallenden Kosten beinhalten soll. Grundsätzlich kann hier zwischen den beiden Formen der kostenorientierten Geschäftsmodelle und der wertorientierten Geschäftsmodelle unterschieden werden. Während kostenorientierte Geschäftsmodelle auf die Minimierung von Kosten gerichtet sind, rücken anfallende Kosten beim zweiten Modell in den Hintergrund und zielen stattdessen verstärkt auf eine kontinuierliche Nutzensteigerung zugunsten des oder der Kunden oder Kundin ab (Metelskaia et al., 2018, S. 36).

4.3.3 Value Proposition Canvas

Basierend auf dem Business Model Canvas hat sich ein weiteres Geschäftsmodell, mit der Bezeichnung Value Proposition Canvas (kurz: VPC) entwickelt. Es kann als Erweiterung des Business Model Canvas verstanden werden, bei welcher die beiden Bausteine Value Proposition und Kundensegment oder Kundinnensegment im Mittelpunkt der Betrachtung stehen. Mit Hilfe von diesem Geschäftsmodell soll aufgezeigt werden, wie das Wert-Angebot zum jeweiligen Profil der Zielgruppe passt und wie deren Bedürfnisse richtig adressiert werden können (Belleflamme & Neysen, 2021, S. 6). Wie aus der folgenden Abbildung 32 hervorgeht, wird beim Value Proposition Canvas das konkrete Wert-Angebot der vom Unternehmen ausgewählten Zielgruppe gegenübergestellt. Auf der rechten Seite des Value Proposition Canvas wird im Bereich des Kundensegments oder Kundinnensegments zunächst jede Zielgruppe aus den drei Blickwinkeln Probleme, Nutzen und Aufgaben betrachtet. So soll in einem ersten Schritt analysiert werden, mit welchen grundlegenden Aufgaben sich die Zielgruppe befasst, welchen Nutzen oder Gewinn sie sich daraus versprechen und welche Probleme sich bei ihrer Aufgabenbewältigung möglicherweise ergeben. Die linke Seite nimmt die Sicht der Anbietenden ein, bei der das Wert-Angebot, sprich das angebotene Produkt oder Dienstleistung die zuvor genannten drei Bereiche des Zielsegments adressieren soll. Das VPC ermöglicht es, abstrakte Konzepte auf einen Blick zu visualisieren und gilt daher als einfaches und häufig zum Einsatz kommendes Instrument (Belleflamme & Neysen, 2021, S. 6).

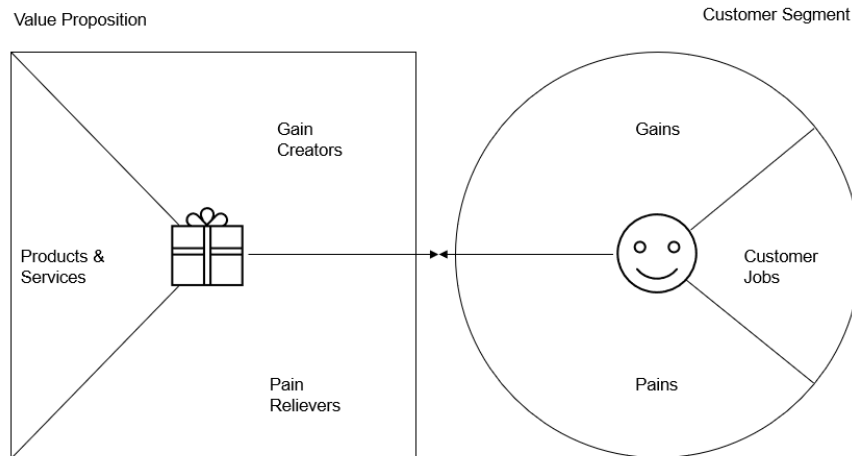


Abbildung 32: Value Proposition Canvas (in Anlehnung an (Osterwalder & Pigneur, 2011))

4.3.4 Business Model Freemium

Das Modell der Freemium-Strategie ist im Unternehmensumfeld zu einem populären Geschäftsmodell herangewachsen und findet insbesondere beim Verkauf digitaler Produkte Anwendung (Boudreau et al., 2021). Der Begriff „Freemium“ setzt sich aus „free“ (deutsch: Frei) und „premium“ (deutsch: Premium) zusammen und bezieht sich auf das Angebot der kostenlosen Nutzung einer Basis-Version eines Produkts oder Services durch die Nutzenden. Gegen Bezahlung haben Nutzende in den meisten Fällen die Möglichkeit auf eine Premium-Version umzusteigen, mit welcher zusätzliche Funktionen des Angebots freigeschaltet werden können (Gu et al., 2018, S. 10).

Freemium Modelle erfreuen sich deshalb an großer Beliebtheit, da durch sie hohe Besucher:innenzahlen generiert werden können und Unternehmen einen direkten Zugang zu potenziellen Kunden oder Kundinnen geboten wird, ohne in Marketingaktivitäten investieren zu müssen. Das Business Modell Freemium kann nämlich eigens als Marketinginstrument fungieren, indem durch die Möglichkeit der kostenlosen Nutzung des Produkts oder Services eine größere Reichweite generiert und mehr Personen angesprochen werden können. Bei Befürwortung durch den oder die Nutzer:in findet es zudem insbesondere über Kanäle wie Mundpropaganda und dem Phänomen des Netzwerkeffekts Verbreitung (Gu et al., 2018, S. 10).

Aus Sicht der Kunden oder Kundinnen bringt die Freemium-Strategie den Vorteil, dass ein Produkt oder Service kostenlos getestet und erlebt werden kann (Gu et al., 2018, S. 10). Dies ist vor allem bei digitalen Produkten von großer Relevanz, da es sich bei diesen größtenteils um Erfahrungsgüter handelt, bei denen es für die Nutzenden

schwierig sein kann, den wahren Wert des Produkts zu entdecken (Boudreau et al., 2021).

4.3.5 Personas

Kunden oder Kundinnen und Nutzende zu verstehen sowie deren Bedürfnisse und Wünsche zu kennen ist für den Erfolg eines Unternehmens maßgeblich entscheidend. Ohne dieses Wissen findet das Produkt oder die Dienstleistung möglicherweise keine Abnehmenden wodurch folglich kein Umsatz generiert werden kann. Erkenntnisse über Wünsche und Bedürfnisse ist daher schon bei der Gestaltung und Entwicklung von Produkten und Dienstleistungen relevant. Eine Möglichkeit dieses Wissen zu generieren und die Zielgruppe besser zu verstehen, ist die Erstellung von sogenannter Personas. Eine Persona ist eine fiktive aber realistische visuelle Darstellung einer bestimmten Kunden- oder Kundinnen-/Nutzer:innengruppe, welche konkrete Merkmale wie beispielsweise Ziele, Bedürfnisse, Verhalten und demographische Daten aufweist (Jain et al., 2019, S. 72). Die Erstellung von Personas ist insbesondere für die Aufschlüsselung der im nachfolgenden Abschnitt 4.3.6 erklärten Customer Journey entscheidend.

4.3.6 Customer Journey Mapping

Unter Customer Journey wird die gesamte Reise, die eine zur Zielgruppe gehörende Person bei der Interaktion mit einem Unternehmen erlebt, verstanden. Beinhaltet sind alle direkten und indirekten Kontaktpunkte eines Nutzers oder eine Nutzerin mit einem Produkt, einem Service oder einer Marke. Dabei können sich positive, negative oder auch neutrale Erfahrungen ergeben, die wiederum Einfluss auf die zukünftige Reise dieser Person nehmen (Lemon & Verhoef, 2016, S. 75).

Die Customer Journey kann grob in drei Phasen unterteilt werden und erstreckt sich dabei von der Vorkaufsphase über die Kaufphase bis hin zur Nachkaufphase. Hervorzuheben ist, dass zusätzlich auch vergangene Erfahrungen und Erlebnisse Berücksichtigung finden, die Einfluss auf jede einzelne Phase nehmen können (Lemon & Verhoef, 2016, S. 75). Die Phasen der Customer Journey sowie zugehörige Verhaltensweisen seitens der Kunden oder Kundinnen und Nutzenden sind in der folgenden Abbildung 33 ersichtlich.

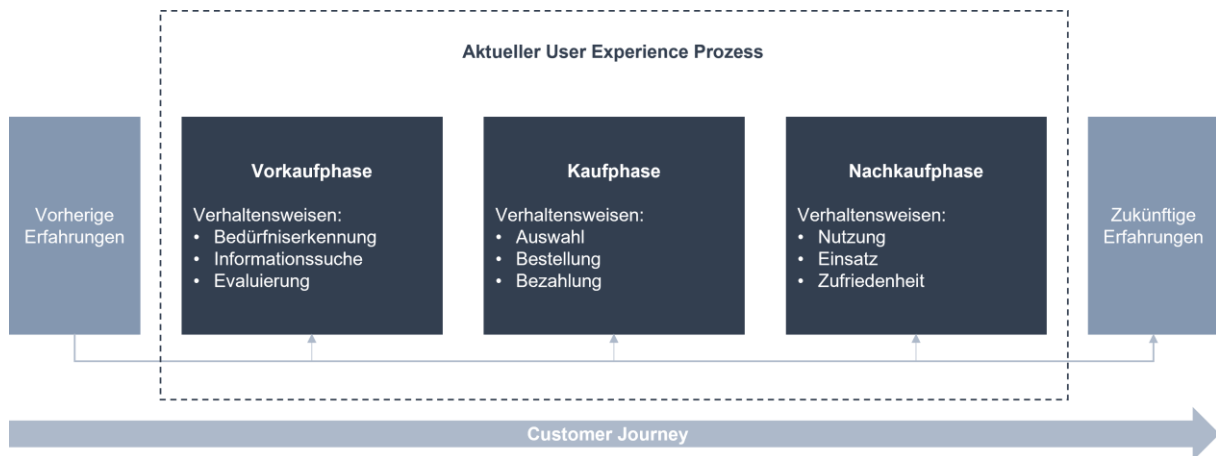


Abbildung 33: Phasen der Customer Journey (in Anlehnung an (Lemon & Verhoef, 2016))

Die Vorkaufphase innerhalb der Customer Journey beinhaltet jegliche Interaktionen die eine Person vor einer Kaufransaktion durchlebt. Typische Verhaltensweisen die dieser Phase zugeordnet werden können sind die Bedürfniserkennung, die Informationssuche sowie auch die Evaluierung des Produkts oder der Dienstleistung (Lemon & Verhoef, 2016, S. 76).

Die zweite Phase der Customer Journey zeichnet sich durch sämtliche Interaktionen aus, die während dem Kauf stattfinden. Demzufolge sind Verhaltensweisen wie die Auswahl, die Bestellung oder auch die Bezahlung in der Kaufphase zu finden (Lemon & Verhoef, 2016, S. 76).

Wie der Name der dritten Phase zu verstehen gibt, geht es in dieser um alle Interaktionen die sich erst nach dem Kauf ereignen. Im Speziellen handelt es sich beispielsweise um die Nutzung des Produkts, dessen Einsatz sowie auch die Zufriedenheit der Abnehmenden (Lemon & Verhoef, 2016, S. 76). Nach der Nachkaufphase ist der Prozess allerdings nicht als beendet anzusehen, da dieser dynamisch und iterativ ist.

Für die Abbildung von Customer Journeys empfiehlt sich Customer Journey Mapping. Dabei handelt es sich um eine visuelle Darstellung der Abfolge von Ereignissen, sprich Interaktionen die ein oder eine Benutzer:in während seiner oder ihrer Reise mit einem Produkt oder Dienstleistung erlebt. Bei Customer Journey Maps müssen zahlreiche Aspekte berücksichtigt werden, die Einfluss auf die User Experience nehmen können, wie beispielsweise alle Schritte und Handlungen, Erwartungen und Wahrnehmungen, Zufriedenheit sowie auch etwaige demographische Informationen (Alvarez et al., 2020, S. 5).

Customer Journey Mapping hilft bei der laufenden Verbesserung der User Experience und ist bereits im Design Prozess eines Produkts oder Services von Bedeutung

(Alvarez et al., 2020, S. 5). Customer Journey Maps unterstützen dahingehend, dass durch die Aufschlüsselung der einzelnen Kontaktpunkte Verbesserungspotentiale rasch aufgedeckt und Anpassungen an dem Produkt oder der Dienstleistung vorgenommen werden können (Alvarez et al., 2020, S. 3).

4.3.7 User Experience Design

Wie bereits im vorherigen Abschnitt 4.3.6 erwähnt, stellt das Konzept der User Experience ein bedeutendes Themenfeld in der heutigen Designpraxis dar. Dem Konzept wird sowohl in der Wissenschaft, als auch in der Wirtschaft zunehmend Beachtung geschenkt, da sich insbesondere Unternehmen im Softwarebereich dadurch voneinander unterscheiden und einen bedeutenden Wettbewerbsvorteil erlangen können (Alvarez et al., 2020, S. 2).

User Experience gilt als zentrales, multidimensionales Konzept heutiger digitaler Geschäftsstrategien (Luther et al., 2020, S. 1), unter welchem die gesamte Erfahrung eines oder einer Nutzers oder Nutzerin verstanden werden kann, die bei der Nutzung eines Produkts oder Dienstleistung entsteht (van der Linden et al., 2019, S. 98–99). Im Mittelpunkt der User Experience steht die nahtlose Erfüllung von Bedürfnissen der Nutzenden. Nahtlos bedeutet in diesem Kontext, die Bedürfnisbefriedigung ohne jegliche Kontroversen und Unannehmlichkeiten entlang der kompletten Customer Journey (Badran & Al-Haddad, 2018, S. 3).

Das Konzept der User Experience bietet eine ganzheitliche Perspektive auf die subjektiven Reaktionen von Nutzenden, die bei der Nutzung des Produkts oder Dienstleistung entstehen. Dabei werden insbesondere die psychologischen Prozesse, die sich aus der Nutzung ergeben beleuchtet (van der Linden et al., 2019, S. 98–99).

Obwohl dem Konzept der User Experience in der Wissenschaft viel Beachtung geschenkt wird, existiert bis dato keine einheitliche Definition des Konzepts (Alvarez et al., 2020, S. 2).

Das folgende CUE-Model (Components of User Experience Model) von Thüring & Mahlke spezifiziert die Kernelemente der User Experience. Es gilt als eines der umfassendsten Modelle, welches die Einflussfaktoren auf die User Experience darstellt (van der Linden et al., 2019, S. 99).

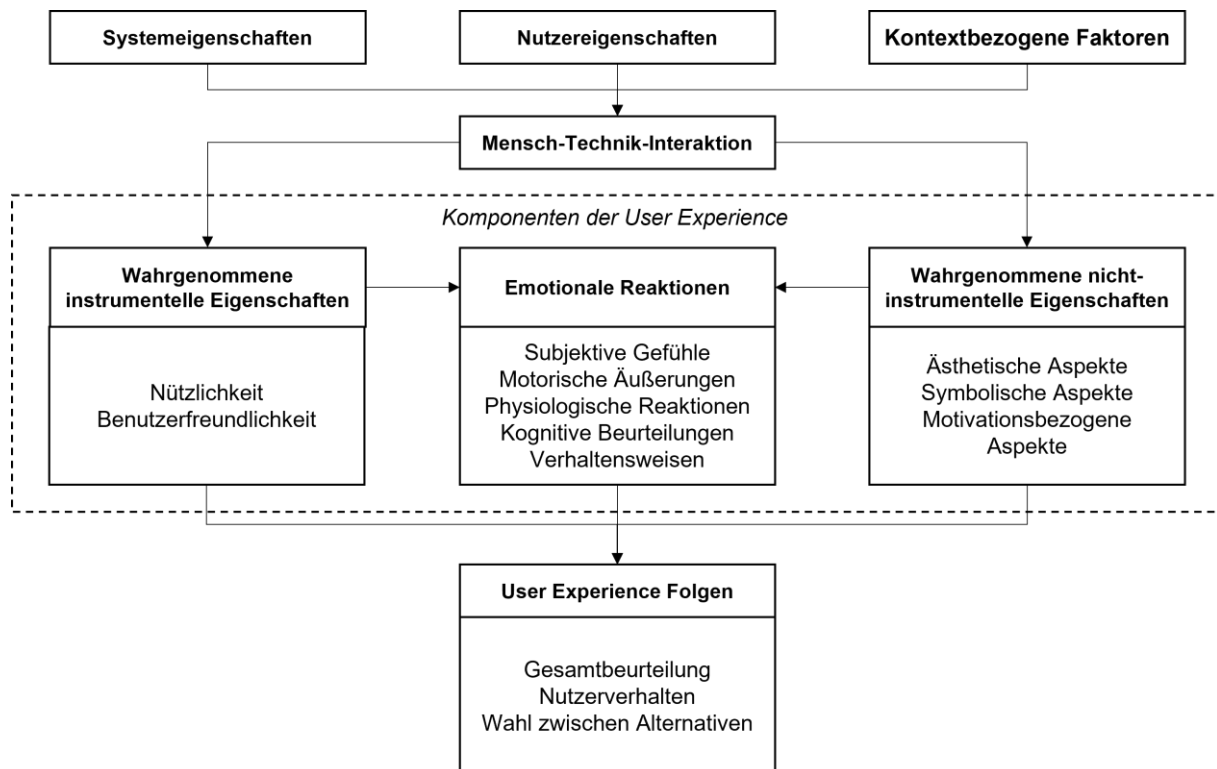


Abbildung 34: Komponenten der User Experience (in Anlehnung an (Thüring & Mahlke, 2007, S. 262))

Wie in Abbildung 34 ersichtlich, entsteht User Experience bei der Interaktion einer Person mit einem Produkt oder Dienstleistung, durch welche in der Regel eine spezifische Aufgabe gelöst werden soll.

Die Interaktion wird begünstigt durch Eigenschaften der Nutzenden (Erfahrungen, Fähigkeiten etc.), Eigenschaften des Systems (Funktionalität, Design etc.) aber auch durch kontextbezogene Faktoren, welche Einfluss auf die Komponenten der User Experience haben (Thüring & Mahlke, 2007, S. 262–263).

Bei den Komponenten der User Experience wird zwischen den wahrgenommenen instrumentellen Eigenschaften und den wahrgenommenen nicht-instrumentellen Eigenschaften des Systems unterschieden. Ersteres bezieht sich auf die Unterstützung, die das System der Person bei der Nutzung bietet wie beispielsweise Benutzerfreundlichkeit oder generelle Nützlichkeit. Wahrgenommene nicht-instrumentelle Eigenschaften beziehen sich auf Aspekte, wie Ästhetik, Motivation oder Symbolik. Beide Arten von Eigenschaften nehmen Einfluss auf die dritte Komponente der User Experience, nämlich die emotionalen Reaktionen, die sich aus der Nutzung des Produkts oder der Dienstleistung ergeben. So kann beispielsweise eine lange Wartezeit beim Laden einer Website, die wahrgenommene Leistungsfähigkeit beeinflussen und zu Ungeduld beziehungsweise Frustration führen. Ein innovatives Design wiederum könnte auf die wahrgenommene visuelle Ästhetik Einfluss nehmen

und in Neugierde oder Vergnügen resultieren. Alle drei Komponenten der User Experience wirken auf die Gesamtbeurteilung eines Produkts oder einer Dienstleistung ein und bestimmen zudem zukünftige Entscheidungen und Verhaltensweisen eines Nutzers oder einer Nutzerin (Thüring & Mahlke, 2007, S. 262–263).

Hinderks et al. (2019, S. 38) heben in ihrer Arbeit hervor, dass für die Schaffung einer guten User Experience, das Produkt einfach zu erlernen sowie effizient in der Nutzung sein sollte sowie zusätzliche Kriterien wie Ästhetik, Freude an der Nutzung, Neuartigkeit oder Attraktivität erfüllen sollte.

User Experience gilt insgesamt als entscheidender Faktor, wenn es um die Schaffung optimierter Produkte geht und sollte bereits während dem Entwicklungs- und Designprozess eines Produkts oder einer Dienstleistung Berücksichtigung finden (Yang et al., 2019, S. 173). Personas und Customer Journey Maps werden dabei immer häufiger als User Experience-Tools eingesetzt, da durch sie eine regelmäßige und einfache Validierung sowie auch Überarbeitung erfolgen kann (Alvarez et al., 2020, S. 3). Eine genaue Erläuterung zu Personas und Customer Journey Mapping ist in den vorherigen Abschnitten 4.3.5 und 4.3.6. zu finden.

In engem Zusammenhang mit dem User Experience Design steht das Konzept des User Interface Design. Es gilt als Haupteinflussfaktor der User Experience, da über die Schnittstelle jegliche Interaktion zwischen Person und System erfolgt (Jokinen et al., 2018, S. 378). Folglich wird auf dieses Konzept ebenfalls näher eingegangen.

4.3.8 User Interface Design

Das User Interface Design gewinnt im Bereich der Informationstechnologie (IT) zunehmend an Bedeutung und gehört zu einem der essentiellen Elemente bei der Entwicklung und Gestaltung einer Anwendung (Ruiz et al., 2021, S. 47). Die User Interface ist der erste Teil eines Systems der von den Nutzenden gesehen wird und verbindet den oder die End-Nutzer:in mit den Funktionen einer Software (Pastushenko et al., 2021, S. 1).

Der Grund für die zunehmende Bedeutung der User Interface liegt in der immer stärker werdenden Heterogenität der Charakteristiken der Nutzer:innen sowie deren Erwartungen in Bezug auf die Nutzer:innenfreundlichkeit einer Plattform. Das User Interface Design sollte daher von Beginn an auf die Bedürfnisse und Fähigkeiten sowie etwaigen Einschränkungen der Nutzenden abgestimmt sein und fortlaufend evaluiert und verbessert werden (Ruiz et al., 2021, S. 47).

4.3.9 Minimum Viable Product

Bei einem Minimum Viable Product handelt es sich um einen Prototypen, der nur die notwendigsten und relevantesten Anforderungen umfasst, um am Markt bestehen zu können (Elberzhager et al., 2017, S. 535–537). Es besteht also lediglich aus den Kernfunktionen und gilt als erste minimal funktionsfähige Iteration eines Produkts. Die Sinnhaftigkeit solcher minimal brauchbaren Produkte ergibt sich durch die Möglichkeit, in relativ kurzen Schleifen Feedback einholen und anschließend rasch Optimierungen am Produkt vornehmen zu können. Dadurch sollen jegliche Fehlentwicklungen, die den Anforderungen der Nutzenden nicht entsprechen, verhindert werden (Shepherd & Gruber, 2021b, S. 978). Die folgende Abbildung 35 dient zur Veranschaulichung.

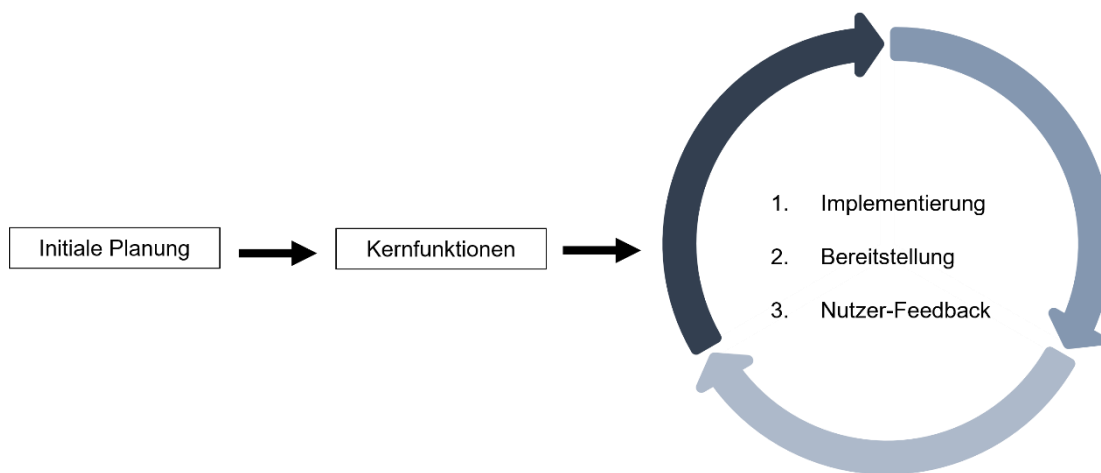


Abbildung 35: Minimum-Viable-Product-Validierungsschleife (in Anlehnung an (Shepherd & Gruber, 2021b, S. 970))

4.3.10 Cloud-Native

Toffetti et al. (2017, S. 166) definieren in ihrer wissenschaftlichen Arbeit den Begriff Cloud-Native wie folgt: „Any application that runs on a cloud infrastructure is a „cloud application“, but a “cloud-native application” is an application that has been specifically designed to run in a cloud environment“.

Bei Cloud-Native handelt es sich folglich um eine agile Methode, bei der Anwendungen vollständig in der Cloud (skalierbare Infrastruktur die dazu dient, mehrere Cloud Computing Dienste zu unterstützen und zu verbinden) entwickelt und betrieben werden. Im Vordergrund steht die Form der Entwicklung von Anwendungen und nicht der Ort der Entwicklung, da sie von Anfang an in einer Cloud-Umgebung laufen. Als wesentlichste Technologien kommen Microservices (siehe Kapitel 4.3.11) und Container (Zusammenfassung einer Anwendung mitsamt aller für die Ausführung benötigten Dateien) zum Einsatz. Zusätzliche Unterstützung bringen agile Methoden (Siehe Kapitel 4.3.12) sowie DevOps-Konzepte (Siehe Kapitel 4.3.13). Die zum

Einsatz kommenden Technologien ermöglichen die Umsetzung entkoppelter Systeme, welche sich insbesondere durch mehr Belastbarkeit und Handhabbarkeit auszeichnen und bei welchen Änderungen und etwaige Anpassungen flexibel und schnell vorgenommen werden können. Ein weiteres Merkmal besteht darin, dass Cloud-Native Applikationen mögliche Ausfälle und Qualitätsschwankungen vorhersehen und diese somit fortlaufend den Nutzenden zur Verfügung stehen (Toffetti et al., 2017, S. 166) .

4.3.11 Microservices

Microservices bauen auf dem Konzept der Modularisierung auf. Folglich stellt jedes Modul ein Microservice dar, welches als unabhängiges System implementiert und betrieben wird (Jamshidi et al., 2018, S. 25). Eine Applikation ist demnach als eine Reihe unabhängiger Dienste aufgebaut, da jedes Microservice autonom entwickelt, bereitgestellt und getestet wird (Waseem et al., 2021, S. 2).

Microservices sind mit einigen Vorteilen verbunden. Im Gegensatz zu anderen Softwarearchitekturen bieten sie eine schnellere Bereitstellung, größere Autonomie bessere Skalierbarkeit und insgesamt mehr Flexibilität (Jamshidi et al., 2018, S. 25).

4.3.12 Agile Softwareentwicklung

Gheorghe et al. schreiben in ihrer Arbeit, dass Agile Softwareentwicklung mehr mit einer Philosophie, sprich mit dem Prozess hin zur Entwicklung einer Software gleichzusetzen ist. Sie definieren “Agil” als *„a set of values and principles and involves the delivery of a good software product to the customer, using an adaptive, incremental and iterative way of working by cross-functional and self-organized teams”* (Gheorghe et al., 2020, S. 90).

Agile Softwareentwicklung ist aus dem Bedarf nach mehr Flexibilität heraus entstanden. Im Gegensatz zu traditionellen Methoden, bei denen die komplette Anwendung am Ende eines Entwicklungsprozesses geliefert wird, geht es bei der Agilen Softwareentwicklung um die laufende Bereitstellung von individuellen, kleinen Teilen einer Software (Gheorghe et al., 2020, S. 90).

Der große Vorteil Agiler Softwareentwicklung ist die Fähigkeit der dynamischen Weiterentwicklung von Teams bei Fokussierung auf Lieferung eines dauerhaft hohen Produktwerts. Im Vordergrund der Agilen Softwareentwicklung steht die Zusammenarbeit im Team und das Verständnis der für den Prozess relevanten und zugewiesenen spezifischen Rollen. Ebenso kann bei Agiler Softwareentwicklung eine dauerhaft hohe Qualität, aufgrund laufender Tests während dem Softwareentwicklungsprozess, zugesprochen werden. Durch regelmäßige Tests haben Entwicklungsteams die Möglichkeit, etwaige Fehler rasch aufzudecken und folglich auch zu beheben (Gheorghe et al., 2020, S. 90).

In der Arbeit von Gheorghe et al. gehen zwölf Prinzipien hervor, durch welche sich Agile Softwareentwicklung auszeichnet. Diese Prinzipien sind in der folgenden Tabelle 15 aufgelistet.

Die zwölf Prinzipien der Agilen Softwareentwicklung	
1	Herstellung von Kunden- oder Kundinnenzufriedenheit durch Continuous Delivery
2	Herunterbrechen großer Arbeitsbrocken in kleine Teile, um eine rasche Fertigstellung zu ermöglichen
3	Selbst-organisierte Teams
4	Bereitstellung der benötigten Arbeitsumgebung
5	Schaffen von Prozessen, die nachhaltige Bemühungen der Teammitglieder fördern
6	Konstantes Tempo bei Arbeitsschritten
7	Offenheit für sich ändernde Anforderungen
8	Tägliche Absprachen zwischen Teammitgliedern
9	Regelmäßiges Überdenken der Effektivität bei der Arbeit
10	Messung des Fortschritts
11	Kontinuierliches Streben nach Spitzenleistungen
12	Wandel als Wettbewerbsvorteil nutzen

Tabelle 15: Prinzipien der Agilen Softwareentwicklung (in Anlehnung an (Gheorghe et al., 2020))

4.3.13 DevOps

Allgemein gesprochen, kann unter dem Begriff DevOps diverse Praktiken, Tools und eine Philosophie verstanden werden, die die Prozesse zwischen Softwareentwicklung und Betrieb automatisiert und integriert. DevOps-Teams arbeiten während dem ganzen Produktlebenszyklus eng zusammen. Das Konzept basiert auf Prinzipien wie Zusammenarbeit, Automatisierung, Messungen und Monitoring. DevOps ermöglichen schnellere Bereitstellungszyklen und erhöhen die Qualität des Software-Deployments (Hrusto et al., 2021, S. 1).

4.3.14 Voice of the Customer

Meinungen von Kunden oder Kundinnen und Nutzer:innen werden als äußerst hilfreich erachtet, wenn es um die Identifikation und Validierung von Bedürfnissen und Ideen geht. Dies betrifft insbesondere die frühen Phasen der Produktentwicklung. Zu diesem Zweck können sich Unternehmen verschiedener Methoden bedienen wie beispielsweise Umfragen, Interviews oder auch Fokusgruppen. Eine Methode mit der

die und Nutzer:innen frühzeitig in den Produktentwicklungsprozess involviert werden können ist das Konzept Voice of the Customer. Dabei handelt es sich um ein Marketing- beziehungsweise Qualitätswerkzeug, welches Unternehmen dabei unterstützt bessere Entscheidungen zu treffen im Hinblick auf die Bedürfnisse der Kunden oder Kundinnen und Nutzer:innen (Melandar, 2020, S. 216). Im Detail handelt es sich dabei um eine strukturierte Methode zur Sammlung von Kunden- oder Kundinnen-/Nutzer:inneninformationen und deren Analyse. Unternehmen können VOC nutzen, um ein Produkt, eine Dienstleistung oder einen Prozess zu verbessern und so das Angebot für Kunden- oder Kundinnen und Nutzer:innen innovativer gestalten. VOC hilft also Unternehmen insgesamt dabei, sich mehr auf Nutzer:innen und ihre konkreten Bedürfnisse zu konzentrieren (Melandar, 2020, S. 219).

Nachdem die im Zusammenhang mit der Product-led Growth Strategie relevantesten Konzepte erläutert wurden, ist im folgenden Abschnitt 4.3.15 nochmals eine Zusammenfassung der in diesem Kapitel behandelten Punkte zu finden.

4.3.15 Übersicht der Konzepte

Der abschließende Abschnitt des Kapitels 4 bietet noch einmal eine Übersicht der Konzepte und zeigt in nachfolgender Tabelle 16 deren Zuordnung zu den jeweiligen Phasen des Product-led Growth Modells.

Phasen \ Konzepte	1. Define	2. Design	3. Build	4. Deploy & Scale	5. Engage & Extend	6. Under- stand & Measure	7. Learn & Adopt
Lean Startup siehe 4.3.1	X						
BMC siehe 4.3.2	X						
VPC siehe 4.3.3	X	X					
Business Model Freemium	X				X		
Personas siehe 4.3.5	X	X					
Customer Journey Mapping siehe 4.3.6	X						
UX/UI Prototyp siehe 4.3.7, 4.3.8		X					
MVP siehe 4.3.9		X					
Microservices siehe 4.3.11		X					
Cloud-Native siehe 4.3.10		X					
DevOps siehe 4.3.13			X	X			
Agile Software- entwicklung siehe 4.3.12			X	X			
Self-/Support					X		
Stickiness					X		
App Store					X		
User Tracking						X	
Usage Tracking						X	
VOC siehe 4.3.14						X	X
Product Roadmap							X
RFIs		X					

Tabelle 16: Zuordnung ausgewählter Konzepte zu Phasen

Wie in der Tabelle ersichtlich, sind in der Spalte ganz links alle Konzepte und in der obersten Zeile alle Phasen die im Product-led Growth Vorgehensmodell berücksichtigt

wurden aufgeschlüsselt. Konzepte, die keine genauere Erklärung benötigen, wie beispielsweise der allgemein bekannte App Store wurden zwar in Abschnitt 4.2 erwähnt, allerdings nicht detailliert beschrieben. Konzepte wiederum, deren Funktionalität sich nicht durch ihre Bekanntheit oder ihren Namen erschließen lassen, wurden ausführlicher behandelt. Das eingetragene „X“ zeigt, welches Konzept welcher Phase zuzuordnen ist. Manche Konzepte sind in mehreren Phasen auffindbar. Es gilt zu betonen, dass in der Theorie und Praxis weitaus mehr Konzepte als im Rahmen dieser Arbeit vorgestellt, zu finden sind, die ergänzend eingesetzt werden können oder andere Konzepte ersetzen. Die in dieser Arbeit ausgewählten Konzepte kommen bei Berücksichtigung von Product-led Growth explizit zum Einsatz oder aber sind für die Erreichung der in Abschnitt 4.2 vorgestellten Zielsetzungen empfehlenswert.

5 Evaluierung

Wie bereits in Abschnitt 3.2 erläutert, soll durch eine Szenarien-Evaluation die konkrete Anwendung der erarbeiteten Ergebnisse anhand des Software Produkts ADOGRC der BOC Group validiert werden. Im Zuge dessen wird der Status quo des gewählten Szenarios mit Hilfe eines qualitativen Interviews erhoben und anschließend Erkenntnisse sowie konkrete Vorschläge im Hinblick auf Product-led Growth abgeleitet. Um einen besseren Überblick zu verschaffen, wird die Beschreibung des derzeitigen IST-Stand des Produkts bereits in die sieben Phasen des entwickelten metamodellbasierten Vorgehensmodells untergliedert. Da es den Rahmen dieser Arbeit sprengen würde, jede Empfehlung aufzuzeigen und das Software-Produkt ADOGRC im Detail zu erläutern, wird nur auf einzelne ausgewählte Informationen zum Produkt und abgeleitete Vorschläge näher eingegangen. Das besagte Interview mit dem technischen Produktmanager der BOC Group, Michael Höhr, hat am 13. Oktober 2022 stattgefunden.

5.1 Define

Die erste Phase Define befasst sich mit der grundlegenden Fragestellung, welches Software-Produkt überhaupt entwickelt und am Markt angeboten werden soll. Bei der BOC Group ist die Idee des Software-Angebots ADOGRC wie folgt entstanden:

Die Komponente Governance, Risk & Compliance wurde vor dem Jahr 2020 als Erweiterung des Software-Produkts ADONIS, einem Geschäftsprozessmanagement-Tool, angeboten. Der Betrieb von ADONIS war daher die technische Grundlage für die Komponente Governance, Risk & Compliance, welche früher vermehrt auf das Szenario internes Kontrollsystems (kurz: IKS) fokussierte. Aufgrund der immer stärker werden Nachfrage nach der alleinstehenden GRC-Anwendung und der Erkenntnis, dass Personen auf der Suche nach einem Software-Angebot sind, welches zugleich verschiedene Szenarien aus der Domäne Governance, Risk & Compliance abdecken kann, wurde die Komponente vollständig von ADONIS abgelöst. Seither hat sich dies als eigenständiges Software-Produkt mit dem Namen ADOGRC etabliert.

ADOGRC kann heute als Multiszenarienplattform verstanden werden, die neben den Szenarien Risikomanagement, Compliance-Management und Internes Kontrollsystem, auch Performance Management, IT-Risikomanagement, Auditmanagement, ESG und viele weitere umfasst. In Abbildung 36 sind die genannten Szenarien inklusive deren Anwenderkreis, Normen oder relevante Parameter ersichtlich. In der Abbildung ersichtliche Abkürzungen sind dem Abkürzungsverzeichnis zu entnehmen.

Um die diversen, inhomogenen Szenarien und Stakeholder bestmöglich unterstützen zu können, ist ADOGRC modular aufgebaut. ADOGRC fußt auf einem flexiblen, metamodel-basierten Konzept, um sicherzustellen, dass sämtliche aktuellen sowie kommenden Methoden schnellst-möglich abgebildet und an die Erfordernisse der Kundschaft angepasst werden können.



Abbildung 36: Szenarien aus dem Bereich GRC des Produkts ADOGRC (BOC Group)

Um das Software-Angebot auf das Profil der Zielgruppe noch genauer abstimmen und deren Bedürfnisse richtig adressieren zu können, bedient sich das Produktmanagement-Team von ADOGRC dem Value Proposition Canvas. Mithilfe dessen sind stets alle Erfordernisse seitens Zielgruppe auf einem Blick ersichtlich und können nach Bedarf angepasst und ergänzt werden. Anpassungen sind beispielsweise in Fällen notwendig, in denen sich die Pains, Jobs oder Gains (siehe Abschnitt 4.3.3) der Kunden oder Kundinnen ändern oder neue Faktoren hinzukommen. In der folgenden Abbildung ist das Value Proposition Canvas mit Fokus auf die Zielgruppe IKS-Manager exemplarisch dargestellt.

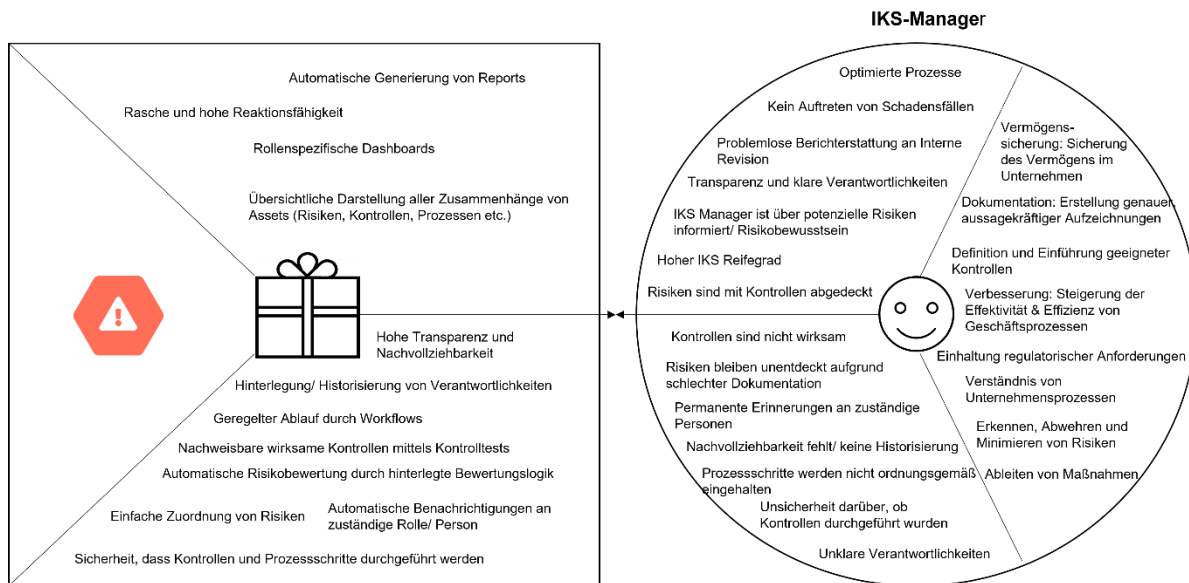


Abbildung 37: Value Proposition Canvas mit Fokus IKS-Manager (BOC Group)

Bei ADOGRC tritt insbesondere die Szenarien-Flexibilität sowie auch die Möglichkeit der Integration verschiedener *Assets* (deutsch: Vermögenswerte) aus anderen Businesssszenarien eines Unternehmens in den Vordergrund. Bei dieser Software erfolgt das Wachstum durch die stetige Erweiterung um Szenarien und deren Überschneidungen. Um ein Beispiel anzuführen: Risikomanagement benötigt Maßnahmen zur Mitigation, genauso wie ein Internes Kontrollsystem Maßnahmen zur Verbesserung benötigt. Sind diese Maßnahmen definiert und im Tool vorhanden, so können damit *Audit-Findings* (Überprüfung der Szenarien und der abgeleiteten notwendigen Verbesserungen) getrackt und somit Audits wiederum ins Tool eingeplant werden.

Um weiteres Wachstum zu fördern ist es für andere Unternehmen und potenzielle Kunden oder Kundinnen bereits möglich von ADOGRC eine Demo-Version anzufordern und diese über eine gewisse Zeitdauer kostenlos zu nutzen. Durch das Angebot dieses Subscription Modells haben User:innen die Möglichkeit den Mehrwert der Software ADOGRC zu testen und zu erfahren. Nachdem das Business Modell von ADOGRC ausgearbeitet wurde, hat das Produktmanagement-Team das Software-Angebot weiter ausspezifiziert und in einem weiteren Schritt die in Abbildung 38 ersichtliche Customer Journey erstellt. Diese besteht aus den Phasen *Discover/Consider*, *Explore/Convince*, *Negotiate/Purchase*, *Deploy/Use*, *Support/Service* und *Engage/Extend*.



Abbildung 38: Customer Journey (BOC Group)

Potenzielle Kunden oder Kundinnen werden insbesondere über folgende Aktivitäten auf das Produkt ADOGRC aufmerksam gemacht: *Social Media*, *Paid Ads*, Empfehlungen, einem Produktblog und Ausschreibungen. In der Phase 2 der Customer Journey wird dem oder der User:in fachspezifischer Content zur Verfügung gestellt, beispielsweise qualifizierte Whitepaper zum Download, detaillierte Blog-Artikel oder auch Anleitungen. Um den oder die User:in in Richtung eines Kaufs zu bewegen, wird wie bereits weiter oben erwähnt, eine kostenlose Demoversion angeboten, mit welcher die User:innen die Möglichkeit haben das Produkt zu testen. Zusätzlich erhalten Interessenten eine Produktpräsentation sowie auch anfängliches Coaching, um das Produkt mit seinen verschiedenen Funktionen sowie Einsatzszenarien zu veranschaulichen. Phase 4 der Customer Journey ist gekennzeichnet durch laufende Kunden- oder Kundinnenbetreuung, welche hauptsächlich durch das Consulting-Team und das Hotline-Team erfolgt. Um die Kunden oder Kundinnen über Neuigkeiten zu dem Produkt und der Domäne Governance, Risk & Compliance zu informieren, wird zudem weiterer Content zur Verfügung gestellt. Darunter fallen beispielsweise Blogthemen zur besseren Nutzung des Produkts, Newsletter oder Updates zu produktspezifischen Erweiterungen. Als grundlegendes Ziel gilt hier (potenzielle) Nutzer:innen zu informierten (potenziellen) Nutzern oder Nutzerinnen umzuwandeln.

Empfehlung unter Berücksichtigung von Product-led Growth

Unter Berücksichtigung von Product-led Growth empfiehlt sich eine stärkere Fokussierung auf Marketinginitiativen, um den Bekanntheitsgrad des Software-Produkts ADOGRC zu erhöhen und gezielt mehr Personen zu erreichen. Insbesondere Content Marketing sollte dabei weiterhin im Fokus stehen und Berücksichtigung finden. Mit informierenden, unterhaltenden sowie auch beratenden Inhalten können Personen vom Leistungsangebot überzeugt und als Kunden oder Kundinnen gewonnen oder gehalten werden. Ziel des Governance, Risk & Compliance Blogs der BOC Group ist es, konkrete Problemstellungen und Suchanfragen relevanter Personen in Blog-Einträgen aufzugreifen und Lösungen im Kontext des Software-Angebots ADOGRC anzubieten. Da der Erfolg eines Blogs maßgeblich mit der Suchmaschinen-Optimierung (englisch: Search Engine Optimisation; kurz: SEO) zusammenhängt, empfiehlt es sich darauf zukünftig mehr zu fokussieren. Hintergrund ist, dass nur mittels SEO, Blog-Einträge unter der Vielzahl an Ergebnissen in Suchmaschinen gefunden werden können und folglich Verbreitung finden.

Eine weitere Empfehlung an dieser Stelle gilt die Konkretisierung der Customer Journey, abgestimmt auf die verschiedenen Buyer Persona Profile. Durch die Erstellung von Personas und das Abbilden ihrer spezifischen Journey können zusätzliche Potenziale des Produkts aufgedeckt und genutzt werden. Dies wäre zudem für die Phase 5 (Engage & Extend) des metamodellbasierten Vorgehensmodells im Hinblick auf die Erweiterung der Use Cases relevant.

5.2 Design

Wie in obigem Abschnitt erwähnt, handelt es sich bei dem Produkt ADOGRC um eine Multiszenarienplattform, welche unterschiedliche Szenarien der Domäne Governance, Risk & Compliance unterstützt. Einen Ausschnitt von damit abgedeckten Szenarien bietet die folgende Abbildung 39.

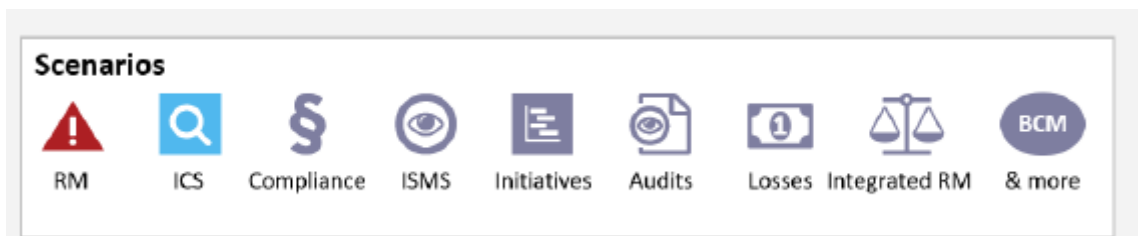


Abbildung 39: Abgedeckte Szenarien des Produkts ADOGRC (BOC Group)

Um benötigte Funktionalitäten seitens der Nachfragenden zu kennen, werden unter anderem klassische Request for Information (kurz: RFI; deutsch: Informationsanfrage) Kataloge herangezogen. Request for Information bezieht sich auf den Prozess der Software-Ausschreibung von Unternehmen und ist ein Teil der Vorauswahl potenzieller Software-Lösungen, bei welchem Unternehmen unverbindlich Preis- und Leistungsinformationen bei Softwareanbietern einholen. Unternehmen haben so die Möglichkeit, sich einen ersten Überblick über die Angebote auf dem Softwaremarkt zu schaffen. Anfragen können beispielsweise sein: allgemeine Kennzahlen des Software Anbieters, Service Portfolio, Branchenabdeckung etc. Neben der Nutzung von RFIs können auch Projekte der Kundschaft dazu beitragen, Bedürfnisse von Nutzenden hinsichtlich etwaiger Funktionalitäten aufzudecken.

Um eine nahtlose User Experience zu ermöglichen, wird insbesondere das Konzept Voice of the Customer herangezogen. Dabei werden Gespräche mit Kundschaft geführt und mögliche Verbesserungsvorschläge aufgegriffen. Diese werden wiederum im Produkt adaptiert und eingearbeitet. Des Weiteren finden auch interne Abstimmungen statt. Die sogenannten Story Workshops (vom Produktmanagement-Team angestoßen) dienen dazu, intern Feedback einzuholen und daraufhin gegebenenfalls weitere Anpassungen vorzunehmen. Insbesondere das Consulting-

Team wird aufgrund der fachlichen Expertise sowie aufgrund der engen Zusammenarbeit mit der Kundschaft bei diesem Prozess involviert.

Wie in Abbildung 40 ersichtlich, sind alle erforderlichen Prozesse in Bezug auf das Software-Angebot ADOGRC genau definiert, aufgeschlüsselt und dokumentiert. Aus Gründen der Geheimhaltung kann auf die detaillierte Prozessabwicklung nicht näher eingegangen werden. Ebenso einen wichtigen Stellenwert nimmt der Tech-Stack ein, in welchem ersichtlich ist, welche Tools für die Entwicklung des Software-Produkts benötigt werden. Zum Einsatz kommende Tools für die Entwicklung des Produkts sind beispielsweise *Java*, *RhodeCode* oder *Git*.

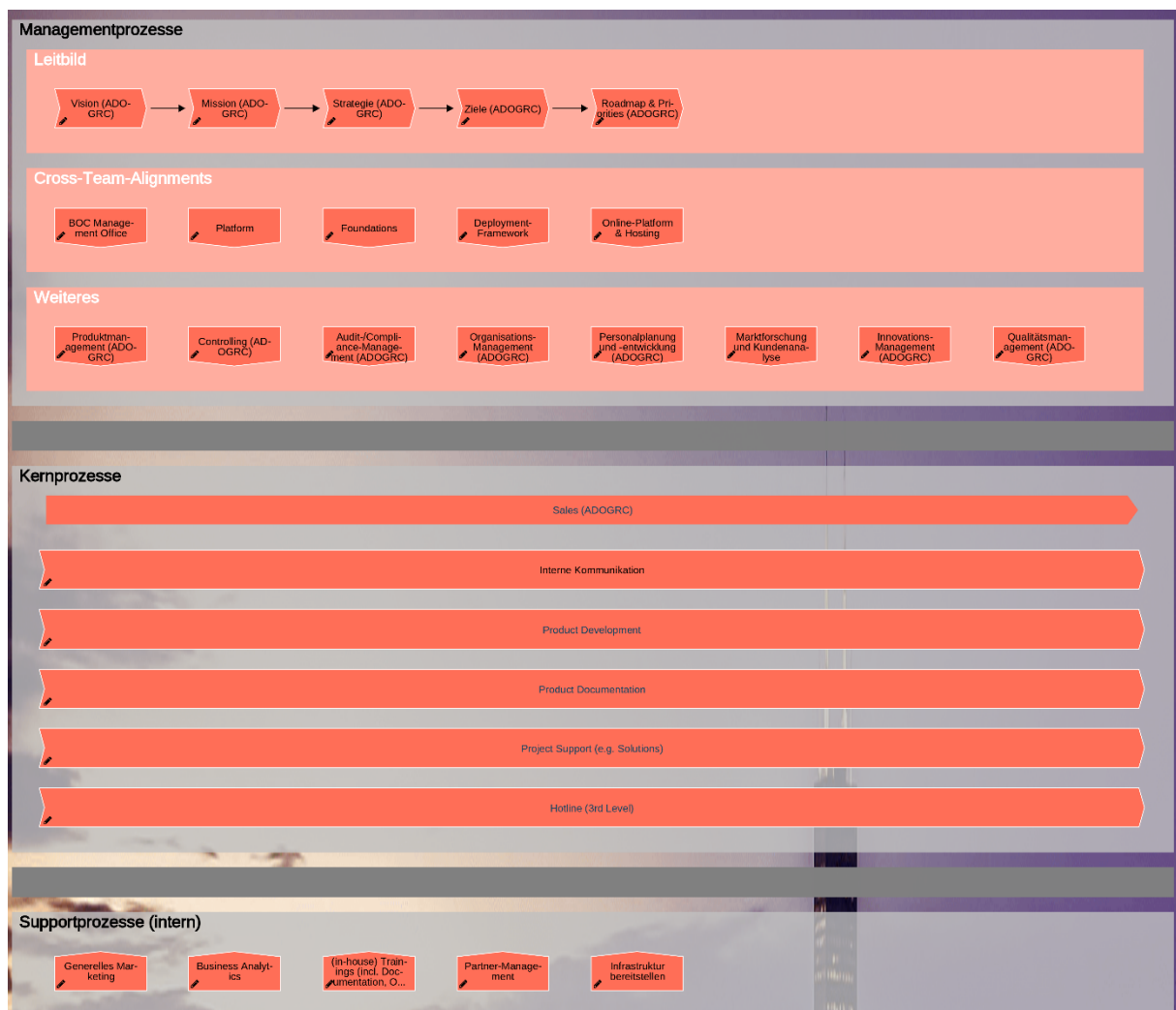


Abbildung 40: ADOGRC Produktorganisation (BOC Group)

Empfehlung unter Berücksichtigung von Product-led Growth

Als Empfehlung gilt hier zukünftig automatische Messpunkte (Mengendaten, User:innen Statistiken etc.) direkt ins Produkt einzubinden, um die Journey der User:innen innerhalb der Software zu tracken. In der Software können verschiedene Nutzende eine Reihe von Aktionen durchführen, welche abhängig vom

Nutzer:innenprofil ist und somit jeweils eine individuelle Journey ergibt. Durch das Tracken der Pfade können Verbesserungen an der User Experience vorgenommen werden. Um nutzer:innenspezifische Routen aufzustellen gilt wiederum die Dokumentation und Ausarbeitung der einzelnen Personas als essenzielle Vorarbeit.

5.3 Build

Das Produktmanagement-Team von ADOGRC verfolgt bei ihrem Software-Entwicklungsprozess einen iterativen Ansatz, welcher nach der Reihenfolge Entwicklung, Integration, (Orchestrierung) und Testen ausgeführt wird.

Die Entwicklung der Software erfolgt grundsätzlich versionsbasiert, sprich mit vorab definierten Zielen und versionsbasierten Release-Zyklen. Konzepte (Architektur) für Produkt-Releases und die Roadmap Planung entspringen einerseits aus der Produktvision und andererseits aus dem Kundenschaftsmarkt im Zuge von Feedback-Zyklen, sprich Voice of the Customer (siehe Abschnitt 4.3.14). Nachdem die Anforderungen spezifiziert wurden, werden diese ins sogenannte *Update Product Backlog* überführt. Aus dem Product Backlog wird folglich ein Sprint mit einer Dauer von zwei Wochen und spezifischen Aufgaben definiert. An dieser Stelle gilt zu betonen, dass sich das Vorgehen des Entwicklungsteams nach dem Modell Scrum orientiert. Bei Scrum handelt es sich um den Ansatz der Agilen Softwareentwicklung. Alle Änderungen und Neuerungen, die innerhalb eines Sprints implementiert werden, werden anschließend im Rahmen einer Sprint Demo relevanten Stakeholdern gezeigt und vorgestellt.

Im Hinblick auf das Testing finden Qualitätsmanagement-Prozesse teilweise parallel zum Scrum Prozess statt. Prozesse, wie *Security Review* (Untersuchung des Quellcodes zur Identifizierung etwaiger Sicherheitslücken) und *Code Signing* (Sicherstellung, dass der Code seit der Signierung nicht verändert oder beschädigt wurde) oder *Web Test Automation* (strukturierte, selbsterklärende, zuverlässige Testreihe, bei der der Schwerpunkt auf der Ausführung von Arbeitsabläufen und der Korrektheit der Daten liegt und kritische Komponenten und Funktionen der Software abdeckt) werden im Gegensatz zu den Sprints permanent, sprich fortlaufend, ausgeführt. *Smoke Tests* (erste grundlegende Probelauf einer Software) erfolgen in der Regel am Ende einer jeden Version durchgeführt und durch *Multi-User-Tests* (Gruppe an Personen testet eine Version einer Software auf unstrukturierte Weise) ergänzt. Zusätzlich gibt es noch *User-Acceptance Tests* (Überprüfung, ob die Software aus Sicht der Benutzenden wie beabsichtigt funktioniert), welche bei Bedarf über die Consultants getestet werden. *Performance Tests* werden ebenfalls bei Bedarf eingeplant. Bei den Releases werden konkret drei Release-Typen unterschieden – *Support Release* (Unterstützung der Nutzung), *Hotfix Release* (Ausbessern eines Fehlers) und *Feature Release* (neue Funktionalität).

Empfehlung unter Berücksichtigung von Product-led Growth

Unter Berücksichtigung des Product-led Growth Ansatzes empfiehlt sich eine Forcierung auf eine kontinuierliche Entwicklung, Integration, Orchestrierung und Testen. Im Vordergrund steht demnach der Begriff Kontinuität, durch welchen sich einige Mehrwerte, wie beispielsweise die nahtlose Nutzung des Produkts im Zuge von Updates, ergeben. Der Begriff Kontinuität ist im Rahmen des Product-led Growth Konzepts ebenfalls für die Phase Design, bezogen auf den Tech-Stack, sowie auch für die gewählte Architektur von Bedeutung.

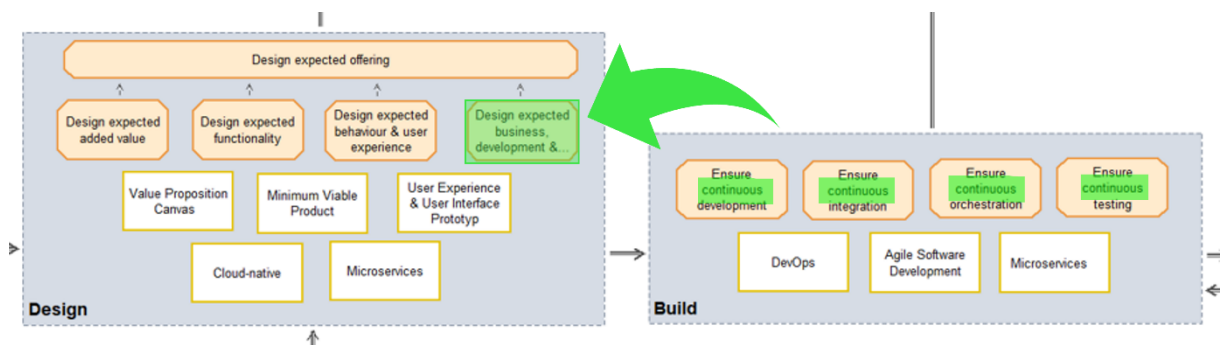


Abbildung 41: Auswirkung von Kontinuität auf die Phase Design (Eigendarstellung)

Die Umstellung auf Kontinuität gelingt durch das Verfolgen des Microservices-Ansatzes, da bei Microservices kein volles Rollout einer Version sondern ein Upgrade von isolierten Komponenten geschieht und Versionshebungen folglich mit einem reduzierteren Risiko einhergehen. Dies hat zum Vorteil, dass im Falle eines Fehlers kurzfristig wieder auf die alte Version zurückgestellt werden kann. Sofern keine Fehler bestehen, kann die Version automatisiert auf die gesamte Kundschaft ausgerollt werden. Des Weiteren können auch Updates problemlos während dem laufenden Betrieb der Software erfolgen, da nur einzelne Teile ausgetauscht werden. Folglich ergibt sich auch als weiterer Vorteil, dass keine *Downtime* bei den Nutzenden entsteht und die Software stets weiter genutzt werden kann.

5.4 Deploy and Scale

Für das Produkt ADOGRC gibt es zwei Möglichkeiten der Auslieferung – *On-Premise* (deutsch: vor Ort) und *Software-as-a-Service* (kurz: SaaS; deutsch: Software als Dienstleistung). Während On-Premise Lösungen in der Infrastruktur der Kundschaft installiert werden, wird SaaS-Angebote von dem anbietenden Unternehmen, in diesem Fall der BOC Group, gehostet. Bei On-Premise muss die Kundschaft eine Lizenz der Software erwerben, um diese nutzen zu können.

Empfehlung unter Berücksichtigung von Product-led Growth

Unter Berücksichtigung von Product-led Growth empfiehlt sich eine weitere Ausrichtung auf eine Microservice-orientierte Cloud-Architektur. Mittels Cloud-Native (siehe Abschnitt 4.3.10) eröffnet sich erst die Möglichkeit der Erreichung der beiden Zielsetzungen „Provide continuous offer delivery“ und „Ensure scalability of the offer delivery“. Diese Verbindung ist in Abbildung 42 veranschaulicht. Lokale Anwendungen (On-Premise) bieten der Kundschaft zwar im Gegensatz zur Cloud ein Maß an Kontrolle, allerdings gilt es an dieser Stelle auch zu bedenken, dass der Kundschaft für den Aufbau, Erhalt und Finanzierung der Infrastruktur verantwortlich ist sowie das gesamte Know-How für den Betrieb benötigt.

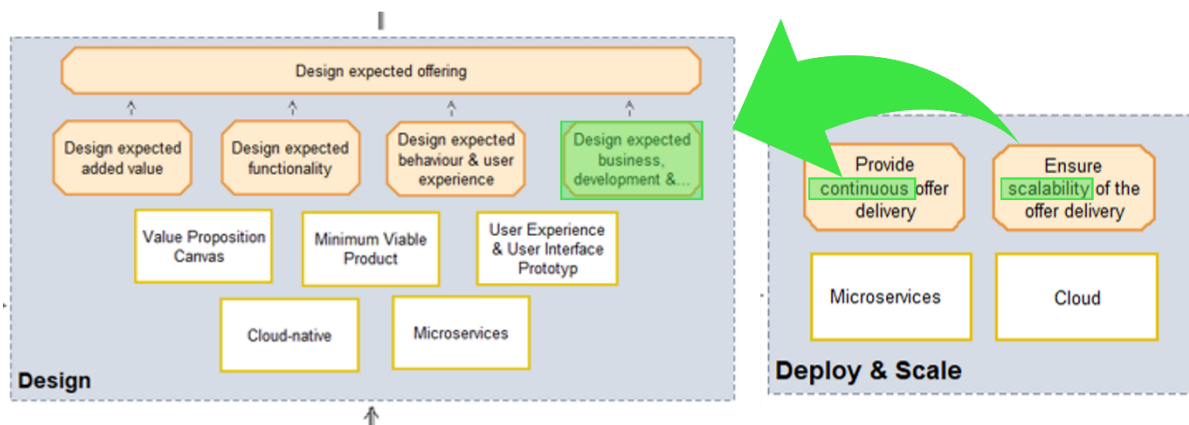


Abbildung 42: Zusammenhang zwischen der Phase Deploy & Scale und Phase Design (Eigendarstellung)

5.5 Engage and Extend

Wie bereits erwähnt, ergibt sich der grundlegende Mehrwert des Software-Produkts ADOGRC aus den verschiedenen Fachszenarien, die mithilfe der Software bedient werden können. So haben Nutzende die Möglichkeit mit den verschiedenen Erweiterungen all ihre Anforderungen aus dem Bereich Governance, Risk & Compliance abzudecken. Aufgrund des modularen Aufbaus haben Nutzende alles in einer Software integriert zur Verfügung stehen. In Bezug auf die sogenannte Stickiness, werden den User:innen verschiedene Features, wie beispielsweise regelmäßige notwendige Bewertungen, Zertifizierungen oder auch regelmäßiges Reporting, an die Hand gegeben, welche dazu anhalten, fortlaufend das Software-Produkt zu nutzen. In Bezug auf Self-/Support hat die Kundschaft die Möglichkeit den Support über die zur Verfügung stehende Hotline oder das GRC Consulting Team zu beziehen. Auch mittels Blog-Beiträgen wird versucht, Nutzende zu lehren und ihnen Hilfestellung zu geben. Durch die stetige Erweiterung von Modulen können einerseits die Nutzungsszenarien erhöht und auch die User:innen-Anzahl erweitert werden. Beispielsweise konnte das Szenario ESG (Environmental, Social, Governance; deutsch: Umwelt, Soziales, Governance) mit der Software aufgegriffen werden,

wodurch eine weitere Zielgruppe, sogenannte Nachhaltigkeitsbeauftragte, erschlossen werden konnte.

Empfehlung unter Berücksichtigung von Product-led Growth

Für die Erreichung der Ziele aus Phase 5 (Enage & Extend) nimmt die Phase 6 (Understand & Measure) einen bedeutenden Stellenwert ein. Dieser Stellenwert ergibt sich daraus, dass ohne stetigem Messen verschiedener Parameter, wie beispielsweise der User Experience keine Erkenntnisse gewonnen und folglich keine Verbesserungen der Software vorgenommen werden können. Beispielsweise lässt sich ohne Tracken (siehe Abschnitt 5.6) nicht feststellen, ob gewisse Funktionen durch User:innen genutzt oder nicht genutzt werden. Um das Tracken innerhalb der Software zu ermöglichen sollte der Ansatz von Microservices weiterverfolgt werden. Mithilfe von Microservices könnten beispielsweise Umfragen direkt ins Tool eingebaut und Klickpfade nachverfolgt werden. Um das Ziel des Self-Supports noch stärker zu forcieren, würde sich die Bereitstellung von E-learnings oder auch Video-Tutorials anbieten, mit welchen User:innen die Möglichkeit haben etwaige Probleme oder Fragestellungen zur Software eigenständig zu lösen.

5.6 Understand and Measure/Learn and Adopt

Wie in Kapitel 4 bereits erwähnt ist die Phase 6 essenziell, um Erkenntnisse im Hinblick auf das Produkt zu gewinnen und Optimierungen vorzunehmen. Dieses Messen und Analysieren sollte mittels ausgewählter Key Performance Indicators (deutsch: Leistungskennzahlen; kurz: KPIs) über die komplette Customer Journey hinweg erfolgen und für jede einzelne Phase Berücksichtigung finden. In der folgenden Abbildung 43 sind ausgewählte relevante KPIs ersichtlich, mithilfe derer Erkenntnisse in Bezug auf das Produkt ADOGRC gewonnen werden können.

PM Dashboard			
Phase	Marketing & Pre-Sales	Sales & Engage	Retention & Operations
KPIs	- Product Page Traffic Over Time - Page Views - Page Users - Demo Request Page Traffic Over Time - Page Views - Page Users - Requests Received - ADOGRC Traffic Sources Over Time - ADOGRC Interactions Over Time - Total Number of Interactions Over Time - Distribution of Interactions by BOC Operation - ADOGRC Blog Subscribers Over Time - ADOGRC LinkedIn Followers Over Time	- Number of Leads Over Time - Leads by Country - Leads by Source - Leads by Type - Number of Opportunities Over Time - Changes regarding active Opportunities - Number of Opportunities Over Time - Volume of Opportunities Over Time - Ratio of Lost vs. Won Opportunities - Ratio of New vs. Existing Business Opportunities - Product Sales Reporting - Product Licence Purchase - Product Licence Recurring - Product Licence SaaS - Product Licence by Operational Unit	- Churn Rate - Hotline - Number of Tickets Over Time - By Client - By Operation - By Deployment Type - By Severity - By Hotline Level - Response Time Over Time - Time Spent On Tickets - Bug Count

Abbildung 43: Key Performance Indicators (BOC Group)

Für eine übersichtliche Darstellung und Analyse der Zahlen, werden die Ergebnisse in speziellen, vorgefertigten Dashboards zusammengefasst und aufgeschlüsselt. So können beispielsweise die Leistung von Marketinginitiativen, den Bekanntheitsgrad und die Lead-Generierung für das Produkt ADOGRC nachvollzogen und bei Bedarf Handlungen vorgenommen werden.

Empfehlung unter Berücksichtigung von Product-led Growth

Eine Empfehlung an dieser Stelle wäre auch hier wieder, den Ansatz von Microservices weiter zu verfolgen. Durch Microservices eröffnet sich die Möglichkeit, *Behaviour tracking* (deutsch: Verhaltensbeobachtung) innerhalb der Software vorzunehmen. So kann beispielsweise festgestellt werden, welche Funktionen von Personen genutzt oder nicht genutzt werden. Über die Auswertung des Nutzungsverhaltens lässt sich folglich herausfinden, welche Funktionen optimiert oder entfernt werden sollten. Zu betonen gilt, dass das Verhalten nur anonym verfolgt wird.

Für das derzeitige Metamodell von ADOGRC würde das bedeuten, dass die Funktionen um das Konzept Tracking, welches sich in *Content Tracking*, *User Tracking* und *Usage Tracking* teilt, erweitert werden.

6 Fazit

Ziel der vorliegenden Masterarbeit war es, ein metamodellbasiertes Vorgehensmodell auf Basis von Product-led Growth für Unternehmen zu entwickeln. So sollen Unternehmen und Organisationen anhand des Modells instanziiieren (anwenden), Erkenntnisse in Bezug auf ihr Produkt gewinnen und basierend darauf Maßnahmen sowie Handlungsempfehlungen ableiten können. Aus diesem Grund wurde zunächst eine systematische Literaturrecherche zu in der Praxis vorkommenden Modellen und Konzepten durchgeführt. Die Literaturrecherche hat gezeigt, dass diverse Konzepte zu finden sind, von denen sich einige unter dem Begriff Product-led Growth zusammenfassen lassen aber auch weitere, wie beispielsweise das Industrial Startup Model, das sich grundsätzlich mit der gleichen Herausforderung wie Product-led Growth beschäftigt, ähnliche sowie auch weitere Konzepte beinhaltet und näher beleuchtet. Es hat sich jedoch ergeben, dass es bis dato kein umfassendes Modell gibt, welches den gewissen Grad an Abstrahierung aufweist, den es benötigt, um für jegliche Art von Software-Angeboten unabhängig ihres Einsatzbereiches anwendbar zu sein.

Der Schwerpunkt der Untersuchung lag daher in der Entwicklung eines metamodellbasierten Modells, welches Unternehmen ein klares Vorgehen aufzeigt, wie bei dem Konzept Product-led Growth vorgegangen werden muss. Das Modell soll nicht nur für Startups sondern auch für bereits etablierte Produkte genutzt werden können. Die Anwendung des Konzepts der Metamodellierung würde folglich eine einfachere Instanziierung von Product-led Growth für (metamodellbasierte) Software-Angebote ermöglichen.

Die im Rahmen dieser Arbeit aufgestellten Forschungsfragen sind in Tabelle 17 nochmals ersichtlich und konnten wie folglich beantwortet werden.

F1	Wie lässt sich Product-led Growth als ein metamodellbasiertes Konzept umsetzen?
F2	Welche bestehenden fachlichen und technischen Konzepte gibt es in der Literatur und Praxis, die mit Product-led Growth zusammenspielen?
F3	Inwiefern lässt sich das Metamodell eines Software-Produkts durch Product-led Growth als metamodellbasiertes Konzept erweitern?

Tabelle 17: Forschungsfragen

Die Frage 1 (F1) lässt sich anhand des in Kapitel 4 abgebildeten Modells beantworten. Zusammenfassend können dabei drei Ergebnisse genannt werden, die innerhalb der Arbeit erzielt wurden. Ergebnis 1 stellt das metamodellbasierte Vorgehensmodell auf Basis von Product-led Growth dar. Ergebnis 2 ergibt sich aus der Erweiterung des Modells um phasenspezifische Zielsetzungen, und Ergebnis 3 umfasst die Zuordnung

ausgewählter Konzepte zu den sich aus Ergebnis 1 ergebenden Vorgehensphasen. Weiters wurde festgestellt, dass sich aus dem Modell drei zentrale Validierungszyklen ableiten lassen – die angebotsorientierte Validierung, die umsetzungsorientierte Validierung und die betriebsorientierte Validierung.

Abschnitt 4.3 befasst sich im Speziellen mit der Beantwortung der zweiten Forschungsfrage (F2). So konnten in der Literatur und Praxis diverse fachliche und technische Konzepte gefunden werden, die mit Product-led Growth zusammenspielen und von Relevanz sind. Hervorzuheben sind beispielsweise das Value Proposition Canvas von Osterwaldner, das Business Model Freemium, Microservices oder auch das User Experience Design. Wie bereits erwähnt, gibt es eine Vielzahl von Konzepten, die von Unternehmen flexibel eingesetzt werden können.

Mit Hilfe von Kapitel 5 und der beinhalteten Szenarien-Evaluierung konnte ebenfalls eine Antwort auf F3 gefunden werden. So konnte das Metamodell des Software-Produkts ADOGRC um die Komponente Tracking sowie dessen Unterteilungen Content Tracking, User Tracking und Usage Tracking erweitert werden.

Zukünftige Forschung könnte an dem metamodellbasierten Vorgehensmodell anknüpfen, indem einzelne Phasen des Modells, wie beispielsweise Phase 6 (Understand & Measure), aufgegriffen und explizit ausgearbeitet werden. Des Weiteren wäre es eine Möglichkeit das erstellte generische Modell im Hinblick auf bestimmte Produkte eines Unternehmens zu kontextualisieren. Dies würde bedeuten, dass das komplette Modell zu diesem Zweck nochmals instanziiert werden müsste.

Abschließend kann gesagt werden, dass das metamodellbasierte Vorgehensmodell auf Basis von Product-led Growth für unterschiedliche Anwendungsszenarien geeignet ist aber auch einige Weiterentwicklungsmöglichkeiten beinhaltet.

7 Literaturverzeichnis

- Alvarez, J., Léger, P.-M., Fredette, M., Chen, S.-L., Maunier, B. & Senecal, S. (2020). An Enriched Customer Journey Map: How to Construct and Visualize a Global Portrait of Both Lived and Perceived Users' Experiences? *Designs*, 4(3), 1–21. <https://doi.org/10.3390/designs4030029>
- Appcues. (2021). *What is product-led growth? Sales-led and marketing-led had their time. The future is product-led growth*. Appcues. <https://www.productled.org/foundations/what-is-product-led-growth>
- Badran, O. & Al-Haddad, S. (2018). The Impact of Software User Experience on Customer Satisfaction. *Journal of Management Information and Decision Sciences*, 21, 1–20.
- Belleflamme, P. & Neysen, N. (2021). A Multisided Value Proposition Canvas for Online Platforms. *Journal of Business Ecosystems*, 2(1), 1–14. <https://doi.org/10.4018/JBE.2021010101>
- Bergener, K., Clever, N. & Stein, A. (2019). *Wissenschaftliches Arbeiten im Wirtschaftsinformatik-Studium*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-57949-7>
- BOC Group. (2022, 6. Juni). *BOC Group: Enterprise Management-Systeme, die Ihre Entscheidungen leichter machen*. <https://www.boc-group.com/de/about/>
- Bork, D., Karagiannis, D. & Pittl, B. (2020). A survey of modeling language specification techniques. *Information Systems*, 87, 1–20. <https://doi.org/10.1016/j.is.2019.101425>
- Boudreau, K. J., Jeppesen, L. B. & Miric, M. (2021). Competing on Freemium: Digital Competition with Network Effects. *Strategic Management Journal*, Artikel smj.3366. Vorab-Onlinepublikation. <https://doi.org/10.1002/smj.3366>
- Brink, A. (2013). *Anfertigung wissenschaftlicher Arbeiten*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-8349-4397-2>
- Burzynski, P. & Karagiannis, D. (2020). *CoChaCo Manual v0.8*.
- Carolla, M. (2015). *Ein Referenz-Datenmodell für Campus-Management-Systeme in deutschsprachigen Hochschulen*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-09347-1>
- Carvalho, L., Galina, S. & Sánchez-Hernández, M. I. (2020). An international perspective of the business incubators' perception about business model canvas for startups. *Thunderbird International Business Review*, 62(5), 503–513. <https://doi.org/10.1002/tie.22112>
- Elberzhager, F., Holl, K., Karn, B. & Immich, T. (2017). Rapid Lean UX Development Through User Feedback Revelation. In M. Felderer, D. Méndez Fernández, B. Turhan, M. Kalinowski, F. Sarro & D. Winkler (Hrsg.), *Lecture Notes in Computer Science. Product-Focused Software Process Improvement* (Bd.

- 10611, S. 535–542). Springer International Publishing.
https://doi.org/10.1007/978-3-319-69926-4_43
- Fink, A. (2005). Conducting Research Literature Reviews: From the Internet to Paper. *Evaluation Journal of Australasia*.
- Fitzgerald, B. & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 123, 176–189.
<https://doi.org/10.1016/j.jss.2015.06.063>
- Foss, N. J. & Saebi, T. (2017). Fifteen Years of Research on Business Model Innovation. *Journal of Management*, 43(1), 200–227.
<https://doi.org/10.1177/0149206316675927>
- Friedrich, T., Schlauderer, S. & Overhage, S. (2019). The impact of social commerce feature richness on website stickiness through cognitive and affective factors: An experimental study. *Electronic Commerce Research and Applications*, 36, 1–19. <https://doi.org/10.1016/j.elerap.2019.100861>
- Gheorghe, A.-M., Gheorghe, I. D. & IATAN, I. L. (2020). Agile Software Development. *Informatica Economica*, 24(2/2020), 90–100.
<https://doi.org/10.24818/issn14531305/24.2.2020.08>
- Gu, X., Kannan, P. K. & Ma, L. (2018). Selling the Premium in Freemium. *Journal of Marketing*, 82(6), 10–27. <https://doi.org/10.1177/0022242918807170>
- Hinderks, A., Schrepp, M., Domínguez Mayo, F. J., Escalona, M. J. & Thomaschewski, J. (2019). Developing a UX KPI based on the user experience questionnaire. *Computer Standards & Interfaces*, 65, 38–44.
<https://doi.org/10.1016/j.csi.2019.01.007>
- Hrusto, A., Runeson, P. & Engström, E. (2021). Closing the Feedback Loop in DevOps Through Autonomous Monitors in Operations. *SN Computer Science*, 2(6), 1–14. <https://doi.org/10.1007/s42979-021-00826-y>
- Jain, P., Djasasbi, S. & Wyatt, J. (2019). Creating Value with Proto-Research Persona Development. In F. F.-H. Nah & K. Siau (Hrsg.), *Lecture Notes in Computer Science. HCI in Business, Government and Organizations. Information Systems and Analytics* (Bd. 11589, S. 72–82). Springer International Publishing. https://doi.org/10.1007/978-3-030-22338-0_6
- Jamshidi, P., Pahl, C., Mendonca, N. C., Lewis, J. & Tilkov, S. (2018). Microservices: The Journey So Far and Challenges Ahead. *IEEE Software*, 35(3), 24–35.
<https://doi.org/10.1109/MS.2018.2141039>
- Jokinen, J. P. P., Silvennoinen, J. & Kujala, T. (2018). Relating Experience Goals With Visual User Interface Design. *Interacting with Computers*, 30(5), 378–395. <https://doi.org/10.1093/iwc/iwy016>
- Karagiannis, D., Burzynski, P., Utz, W. & Buchmann, R. A. (2019). A Metamodeling Approach to Support the Engineering of Modeling Method Requirements. In *2019 IEEE 27th International Requirements Engineering Conference (RE)* (S. 199–210). IEEE. <https://doi.org/10.1109/RE.2019.00030>

- Karagiannis, D., Döller, V. & Kunnummel, V. *Metamodeling*. Universität Wien.
- Karagiannis, D. & Höfferer, P. (2006). Metamodels in action: An overview. *ICSOFT*, 27–36.
- Karagiannis, D. & Kühn, H. (2002). Metamodelling Platforms.
- Klimanov, D. & Tretyak, O. (2019). Linking business model research and marketing: new network-based approach to business model analysis. *Journal of Business & Industrial Marketing*, 34(1), 117–136. <https://doi.org/10.1108/JBIM-12-2017-0330>
- Kuester, S., Konya-Baumbach, E. & Schuhmacher, M. C. (2018). Get the show on the road: Go-to-market strategies for e-innovations of start-ups. *Journal of Business Research*, 83, 65–81. <https://doi.org/10.1016/j.jbusres.2017.09.037>
- Lemon, K. N. & Verhoef, P. C. (2016). Understanding Customer Experience Throughout the Customer Journey. *Journal of Marketing*, 80(6), 69–96. <https://doi.org/10.1509/jm.15.0420>
- Lindič, J. & Da Marques Silva, C. (2011). Value proposition as a catalyst for a customer focused innovation. *Management Decision*, 49(10), 1694–1708. <https://doi.org/10.1108/00251741111183834>
- Luther, L., Tiberius, V. & Brem, A. (2020). User Experience (UX) in Business, Management, and Psychology: A Bibliometric Mapping of the Current State of Research. *Multimodal Technologies and Interaction*, 4(2), 1–19. <https://doi.org/10.3390/mti4020018>
- Ma, Z., He, X. & Liu, C. (2013). Assessing the quality of metamodels. *Frontiers of Computer Science*, 7(4), 558–570. <https://doi.org/10.1007/s11704-013-1151-5>
- Melander, L. (2020). Customer involvement in product development. *Benchmarking: An International Journal*, 27(1), 215–231. <https://doi.org/10.1108/BIJ-04-2018-0112>
- Metelskaia, I., Ignatyeva, O., Deneff, S. & Samsonowa, T. (2018). A business model template for AI solutions. In L. Moutinho & X.-S. Yang (Hrsg.), *Proceedings of the International Conference on Intelligent Science and Technology - ICIST '18* (S. 35–41). ACM Press. <https://doi.org/10.1145/3233740.3233750>
- NewsRX LLC (24. Juli 2019). Appcues Embraces Product-Led Growth as the Future of Software. *Computer Weekly News*, S. 27. <https://link.gale.com/apps/doc/A594008066/ITOF?u=43wien&sid=bookmark-ITOF&xid=881fac5f>
- Osterwalder, A. & Pigneur, Y. (2011). *Business Model Generation: Ein Handbuch für Visionäre, Spielveränderer und Herausforderer* (J. T. A. Wegberg, Übers.). Campus Verlag. <https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=832895>
- Osterwalder, A., Pigneur, Y. & Tucci, C. L. (2005). Clarifying Business Models: Origins, Present, and Future of the Concept. *Communications of the*

- Association for Information Systems*, 16.
<https://doi.org/10.17705/1CAIS.01601>
- Pastushenko, O., Hynek, J. & Hruška, T. (2021). Evaluation of user interface design metrics by generating realistic-looking dashboard samples. *Expert Systems*, 38(5), 1–16. <https://doi.org/10.1111/exsy.12434>
- Ruiz, J., Serral, E. & Snoeck, M. (2021). Unifying Functional User Interface Design Principles. *International Journal of Human–Computer Interaction*, 37(1), 47–67. <https://doi.org/10.1080/10447318.2020.1805876>
- Schuhmacher, M. C., Kuester, S. & Hultink, E. J. (2018). Appetizer or Main Course: Early Market vs. Majority Market Go-to-Market Strategies for Radical Innovations. *Journal of Product Innovation Management*, 35(1), 106–124. <https://doi.org/10.1111/jpim.12379>
- Shein, E. (2022). *SaaS companies are doubling down on product-led growth*. <https://www.techrepublic.com/article/report-saas-companies-product-led-growth/>
- Shepherd, D. A. & Gruber, M. (2021a). The Lean Startup Framework: Closing the Academic–Practitioner Divide. *Entrepreneurship Theory and Practice*, 45(5), 967–998. <https://doi.org/10.1177/1042258719899415>
- Shepherd, D. A. & Gruber, M. (2021b). The Lean Startup Framework: Closing the Academic–Practitioner Divide. *Entrepreneurship Theory and Practice*, 45(5), 967–998. <https://doi.org/10.1177/1042258719899415>
- Sinzig, C. (2017). *Nichtmarktstrategien multinationaler Unternehmen*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-16845-2>
- Souza, R., Malta, K., Silva, R., Masiero, P., Almeida, E. & Machado, I. (2019). A Case Study about Startups' Software Development Practices. In A. B. Albuquerque & A. L. B. de Paula Barros (Hrsg.), *Proceedings of the XVIII Brazilian Symposium on Software Quality* (S. 198–203). ACM. <https://doi.org/10.1145/3364641.3364663>
- Teece, D. J. (2018). Business models and dynamic capabilities. *Long Range Planning*, 51(1), 40–49. <https://doi.org/10.1016/j.lrp.2017.06.007>
- Teegene, E. W., Seppänen, P. & Ahmad, M. O. (2019). Software development methodologies and practices in start-ups. *IET Software*, 13(6), 497–509. <https://doi.org/10.1049/iet-sen.2018.5270>
- Thüring, M. & Mahlke, S. (2007). Usability, aesthetics and emotions in human–technology interaction. *International Journal of Psychology*, 42(4), 253–264. <https://doi.org/10.1080/00207590701396674>
- Toffetti, G., Brunner, S., Blöchliger, M., Spillner, J. & Bohnert, T. M. (2017). Self-managing cloud-native applications: Design, implementation, and experience. *Future Generation Computer Systems*, 72, 165–179. <https://doi.org/10.1016/j.future.2016.09.002>

- van der Linden, J., Amadieu, F., Vayre, E. & van de Leemput, C. (2019). User Experience and Social Influence: A New Perspective for UX Theory. In A. Marcus & W. Wang (Hrsg.), *Lecture Notes in Computer Science. Design, User Experience, and Usability. Design Philosophy and Theory* (Bd. 11583, S. 98–112). Springer International Publishing. https://doi.org/10.1007/978-3-030-23570-3_9
- Vial, G. (2019). Understanding digital transformation: A review and a research agenda. *The Journal of Strategic Information Systems*, 28(2), 118–144. <https://doi.org/10.1016/j.jsis.2019.01.003>
- Waseem, M., Liang, P., Shahin, M., Di Salle, A. & Márquez, G. (2021). Design, monitoring, and testing of microservices systems: The practitioners' perspective. *Journal of Systems and Software*, 182, 1–44. <https://doi.org/10.1016/j.jss.2021.111061>
- Xiao, Y. & Watson, M. (2019). Guidance on Conducting a Systematic Literature Review. *Journal of Planning Education and Research*, 39(1), 93–112. <https://doi.org/10.1177/0739456X17723971>
- Yang, B., Liu, Y., Liang, Y. & Tang, M. (2019). Exploiting user experience from online customer reviews for product design. *International Journal of Information Management*, 46, 173–186. <https://doi.org/10.1016/j.ijinfomgt.2018.12.006>

8 Abbildungsverzeichnis

Abbildung 1: Vorgehen bei einer systematischen Literaturanalyse	4
Abbildung 2: Strukturbild der vorliegenden Arbeit.....	10
Abbildung 3: Berücksichtigung des Marketing-Mix bei der Go-to-Market Strategie ..	12
Abbildung 4: Das Produkt als hauptsächlicher Wachstumstreiber.....	13
Abbildung 5: Modellierungs-Rahmenwerk	17
Abbildung 6: Metamodellierungs-Hierarchie	18
Abbildung 7: Metaelemente	19
Abbildung 8: Vereinfachtes Beispiel eines Metamodells	19
Abbildung 9: Von CoChaCo empfohlene Schritte zur Erstellung eines Metamodells	22
Abbildung 10: Hauptkonstrukte von CoChaCo	23
Abbildung 11: Vorgehensweise bei der Erstellung des PLG-Modells	24
Abbildung 12: Zuordnung einer Anforderung zu einem der fünf Konstrukte	25
Abbildung 13: Nutzung des Hierarchiemodells als Mind Mapping	26
Abbildung 14: Elemente des PLG-Modells	27
Abbildung 15: Vorgehen Szenarien-Evaluierung	28
Abbildung 16: Metamodellbasiertes Vorgehensmodell „PLG-oriented cycle for software-offerings”	29
Abbildung 17: Angebotsorientierte Validierungszyklus	32
Abbildung 18: Umsetzungsorientierte Validierung	33
Abbildung 19: Betriebsorientierte Validierung.....	33
Abbildung 20: Erweiterung des Vorgehensmodells um phasenspezifische Zielsetzungen	34
Abbildung 21: Phase 1 (Define) erweitert um Zielsetzungen	35
Abbildung 22: Phase 2 (Design) erweitert um konkrete Zielsetzungen	36
Abbildung 23: Phase 3 (Build) erweitert um konkrete Zielsetzungen.....	37
Abbildung 24: Phase 4 (Deploy & Scale) erweitert um konkrete Zielsetzungen	38
Abbildung 25: Phase 5 (Engage & Extend) erweitert um konkrete Zielsetzungen	39
Abbildung 26: Phase 6 (Understand & Measure) erweitert um konkrete Zielsetzungen	40
Abbildung 27: Phase 7 (Learn & Adopt) erweitert um konkrete Zielsetzungen.....	41
Abbildung 28: Metamodellbasiertes Vorgehensmodell erweitert um ausgewählte Konzepte	42
Abbildung 29: Build-Measure-Learn Feedback Loop.....	43
Abbildung 30: Bausteine des Lean-Startup Frameworks.....	44
Abbildung 31: Business Model Canvas	46
Abbildung 32: Value Proposition Canvas.....	48
Abbildung 33: Phasen der Customer Journey	50
Abbildung 34: Komponenten der User Experience.....	52
Abbildung 35: Minimum-Viable-Product-Validierungsschleife.....	54

Abbildung 36: Szenarien aus dem Bereich GRC des Produkts ADOGRC	61
Abbildung 37: Value Proposition Canvas mit Fokus IKS-Manager	62
Abbildung 38: Customer Journey	63
Abbildung 39: Abgedeckte Szenarien des Produkts ADOGRC	64
Abbildung 40: ADOGRC Produktorganisation	65
Abbildung 41: Auswirkung von Kontinuität auf die Phase Design.....	67
Abbildung 42: Zusammenhang zwischen der Phase Deploy & Scale und Phase Design	68
Abbildung 43: Key Performance Indicators.....	70

9 Tabellenverzeichnis

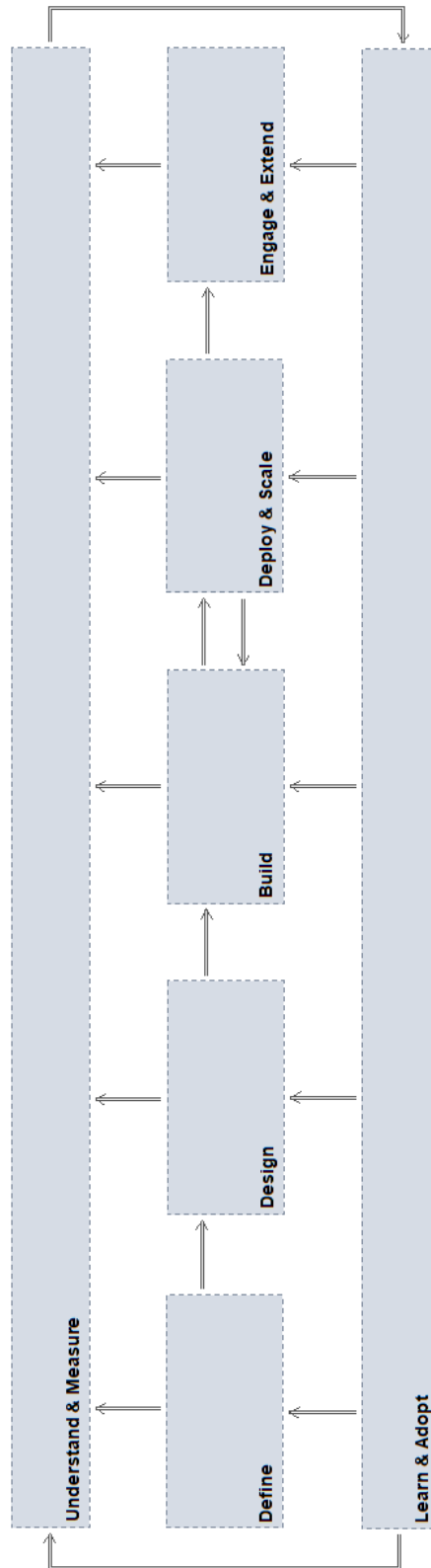
Tabelle 1: Im Rahmen dieser Arbeit angestrebte Ergebnisse.....	2
Tabelle 2: Aufgestellte Forschungsfragen	3
Tabelle 3: Für die Literaturrecherche gewählte Datenbanken	5
Tabelle 4: Für die Literaturrecherche definierte Suchbegriffe	5
Tabelle 5: Kombination der Suchbegriffe.....	6
Tabelle 6: Filter- und Auswahlkriterien der Literaturanalyse	7
Tabelle 7: Anzahl Suchergebnisse in Suchmaschinen	7
Tabelle 8: Anzahl der Ergebnisse nach Sichtung der Publikationen.....	7
Tabelle 9: Merkmale eines Modells	16
Tabelle 10: Eigenschaften eines Metamodells	20
Tabelle 11: Von CoChaCo unterstützte Modelltypen	23
Tabelle 12: Phasen des metamodellbasierten Vorgehensmodells	30
Tabelle 13: Übersicht ausgewählter Konzepte	42
Tabelle 14: Einordnung der neun Bereiche des BMC in vier Elemente	45
Tabelle 15: Prinzipien der Agilen Softwareentwicklung	56
Tabelle 16: Zuordnung ausgewählter Konzepte zu Phasen	58
Tabelle 17: Forschungsfragen	71

10 Abkürzungsverzeichnis

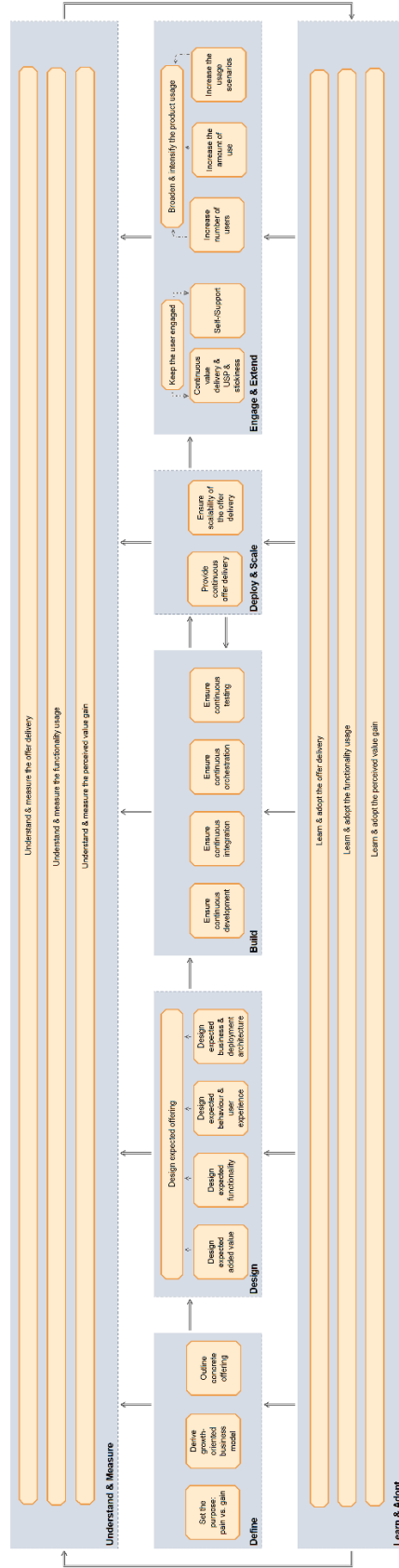
Abkürzung	
BMC	Business Model Canvas
VPC	Value Proposition Canvas
VOC	Voice of the Customer
IKS	Internes Kontrollsystem
RW	Rechnungswesen
CO	Controlling
HR	Human Resources
ERM	Enterprise Risk Management
ISMS	Information Security Management
BSI	Bundesamt für Sicherheit in der Informationstechnik
COBIT	Control Objectives for Information and Related Technology
MIFID	Market in Financial Instruments Directive
BWG	Bankwesengesetz
MaRisk	Mindestanforderungen an das Risikomanagement
QM	Qualitätsmanagement
ESG	Environmental, Social, Governance
CSRD	Corporate Sustainability Reporting Directive
OpRisk	Operationelles Risiko
SEO	Search Engine Optimization
RM	Risikomanagement
Etc	Et cetera
GRC	Governance, Risk and Compliance
USP	Unique selling Proposition

11 Anhang

PRODUCT-LED GROWTH - ORIENTED CYCLE FOR SOFTWARE-OFFERINGS



PRODUCT-LED GROWTH ORIENTED CYCLE FOR SOFTWARE OFFERINGS



PRODUCT-LED GROWTH-ORIENTED CYCLE FOR SOFTWARE-OFFERINGS

