

Vorgehensmodelle bei Festpreisprojekten

Bachelorarbeit

eingereicht von: **Ing. Christoph Kerschenbauer**
Matrikelnummer: 5190 5467

im Fachhochschul-Bachelorstudiengang Wirtschaftsinformatik (0470)
der Ferdinand Porsche FernFH

zur Erlangung des akademischen Grades eines
Bachelor of Arts in Business

Betreuung und Beurteilung: Andreas Ertler, BA MA

Wiener Neustadt, Mai, 2022

Ehrenwörtliche Erklärung

Ich versichere hiermit,

1. dass ich die vorliegende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Inhalte, die direkt oder indirekt aus fremden Quellen entnommen sind, sind durch entsprechende Quellenangaben gekennzeichnet.
2. dass ich diese Bachelorarbeit bisher weder im Inland noch im Ausland in irgendeiner Form als Prüfungsarbeit zur Beurteilung vorgelegt oder veröffentlicht habe.

Vorau, Mai 2022

Unterschrift

Creative Commons Lizenz

Das Urheberrecht der vorliegenden Arbeit liegt bei beim Autor. Sofern nicht anders angegeben, sind die Inhalte unter einer Creative Commons <„Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz“ (CC BY-NC-SA 4.0)> lizenziert.

Die Rechte an zitierten Abbildungen liegen bei den in der jeweiligen Quellenangabe genannten Urheber*innen.

Die Kapitel 2 bis 4 der vorliegenden Bachelorarbeit wurden im Rahmen der Lehrveranstaltung „Bachelor Seminar 1“ eingereicht und am 19.01.2022 als Bachelorarbeit 1 angenommen.
--

Kurzzusammenfassung: Vorgehensmodelle bei Festpreisprojekten

Die vorliegende Arbeit befasst sich mit verschiedenen Vorgehensmodellen in der Softwareentwicklung bei Festpreisprojekten und deren Einfluss auf die Kundenzufriedenheit bei der Bedienung der Software.

Grundsätzlich bestehen Vorgehensmodelle aus Aktivitäten und den daraus resultierenden Daten sowie wiederum einer festgelegten Ausführungsreihenfolge der Aktivitäten. Durch diese Vorgaben wird eigentlich auch die zugrundeliegende Abarbeitung der Aktivitäten bestimmt. Obwohl die positiven Auswirkungen agiler Vorgehensmodelle bei der Entwicklung von Software bereits gut dokumentiert sind, so gibt es nur wenige Studien über die Auswirkungen auf die tatsächliche Kundenzufriedenheit des Endprodukts. Sind Kund:innen von Software-Festpreisprojekten, welche mit agilen Vorgehensmodellen umgesetzt werden, zufriedener als Kund:innen von Software-Festpreisprojekten, die mit linearen Vorgehensmodellen umgesetzt werden?

In dieser Studie wurde eine Umfrage durchgeführt, welche auf die Kundenzufriedenheit von Softwareprogrammen eingeht und in Form eines Customer Satisfaction Scores miteinander vergleicht. Die Fragen beziehungsweise die daraus resultierenden Daten wurden so gewählt, dass Endbenutzer:innen angeben können, wie zufrieden sie mit Softwareprodukten sind.

Die vorliegende Arbeit zeigt, dass die Kundenzufriedenheit nicht wesentlich durch das Vorgehensmodell beeinflusst wird, mit welcher die Software entwickelt wurde. Die Kundenzufriedenheit liegt bei sämtlichen Vorgehensmodellen auf einem ähnlichen Niveau.

Schlagwörter:

Kundenzufriedenheit, Agile Vorgehensmodelle, Lineare Vorgehensmodelle, Hybride Vorgehensmodelle, Customer Satisfaction Score, Scrum, Wasserfallmodell

Abstract: Software methodologies for fixed-price projects

This paper consists of different methodologies in software development such as agile software development and the waterfall approach for projects with fixed prices and their impact on the customer satisfaction for the finished software product.

Such methodologies describe how projects are planned as well as developed and in which order activities within the projects are implemented. Although the positive effects of agile methodologies in the development phase are already well investigated, there are few studies on the effects on the actual customer satisfaction of the finished product. Are customers of fixed-price software projects implemented with agile process models more satisfied than customers of fixed-price software projects implemented with linear process models?

The empirical method for this study was a survey that aims to get the satisfaction rate of customers in form of a Customer Satisfaction Score. With this approach, a statement of whether the project methodology has an impact or not can be stated.

Data has shown that the impact of the software development approach is negligible and has no proven effect on the satisfaction of user experience with the software. Customer satisfaction maintains a similar level regardless of the chosen software development approach.

Keywords:

Customer satisfaction, Agile methodologies, Waterfall approach, Hybrid method, Customer satisfaction score, Scrum

Inhaltsverzeichnis

1. EINFÜHRUNG	1
1.1 Ausgangssituation	1
1.2 Zielsetzung der Arbeit	2
1.3 Aufbau und methodisches Vorgehen	2
2. THEORETISCHE GRUNDLAGEN	4
2.1 Vorgehensmodelle	4
2.1.1 Lineare Vorgehensmodelle	4
2.1.2 Agile Vorgehensmodelle	8
2.1.3 Mischformen	11
2.2 Softwareentwicklung	12
2.2.1 Aktuelle Standards in der Softwareentwicklung	12
2.2.2 Continuous Integration	14
2.2.3 Softwarequalität	16
2.3 Kundenzufriedenheit	17
2.3.1 Definition	17
2.3.2 Stand der Wissenschaft und Technik zur Kundenzufriedenheit und dem eingesetzten Vorgehensmodell	18
3. KONZEPTIONELLES VORGEHEN UND LÖSUNGSANSATZ	23
3.1 Literaturstudie der Softwareprojekte	23
3.2 Kriterien und Themen für Fragebogen	24
3.3 Evaluationskriterien	24
3.3.1 Begriffsdefinition Evaluationskriterien	24
3.3.2 Kundenzufriedenheit messen mit Key-Performance-Indicator	24
4. DISKUSSION	26
4.1 Bewertung Lineare Vorgehensmodelle	26
4.2 Bewertung agile Vorgehensmodelle	26
4.3 Bewertung Hybride Vorgehensmodelle	27
4.4 Ausblick	28

5. METHODIK DER EMPIRISCHEN UNTERSUCHUNG	29
5.1 Untersuchungsdesign	29
5.2 Stichprobe	29
5.3 Fragebogendesign	31
5.3.1 Persönliche Angaben	31
5.3.2 Vorgehensmodelle	32
5.3.3 Softwarelösungen	32
5.3.4 Abschluss	33
6. ERGEBNISSE	34
6.1 Ergebnisse der Literaturstudie	34
6.2 Ergebnisse der Befragung	34
6.2.1 Persönliche Angaben	34
6.2.2 Vorgehensmodelle	37
6.2.3 Softwarelösungen	39
7. ANALYSE DER ERGEBNISSE	43
7.1 Verifikation der Hypothese	43
7.1.1 Hypothese	43
7.1.2 Daten zur Hypothese	43
7.1.3 Evaluierung Hypothese	47
7.2 Weitere Ergebnisse	48
7.2.1 Festpreisprojekte und variable Projekte	48
7.2.2 Kundenzufriedenheit und Branche	49
7.2.3 Vorgehensmodelle unbekannt nach Branche	50
8. SCHLUSSFOLGERUNGEN	51
8.1 Einordnung der Ergebnisse	51
8.2 Stärken und Schwächen der Studie	52
8.3 Conclusio	53
9. LITERATURVERZEICHNIS	53
10. ABBILDUNGSVERZEICHNIS	59
11. APPENDIX	60

1. Einführung

1.1 Ausgangssituation

Softwareprojekte sind aus meiner Sicht Teil des Alltags und fest in unserer Konsumgesellschaft etabliert. Nahezu jede Person ist direkt oder indirekt mit Softwareprojekten beziehungsweise den daraus resultierenden Ergebnissen in Form von Softwareprogrammen in Kontakt. Im Jahr 2021 waren zum Beispiel 92,5 % aller Personen in Österreich online und haben das Internet, und verbundene notwendige Software dafür, genutzt (Statistik Austria 2021).

Für Endbenutzer ist es in der Regel nicht relevant, wie eine Software entwickelt wird, sondern lediglich, ob das Ergebnis den Erwartungen hinsichtlich des Anwendungsfalles entspricht. Eine Handy Applikation sollte in etwa schnell und einfach ohne Anleitungen mit Touch zu bedienen sein (Nielsen 2020). Ein CAD-Programm hat hingegen umfangreichere Funktionalitäten, die teilweise Tastaturbedienung erfordern bzw. bieten, um damit effizient arbeiten zu können. User:innen die ihre Tätigkeiten rein am Keyboard erledigen können sind bei der Abarbeitung ihrer Aufgaben nämlich schneller (Jardina et al. 2009).

Genau so vielschichtig wie das Ergebnis beziehungsweise das Produkt selbst ist auch der Weg dorthin. Es gibt verschiedenste Wege bzw. Vorgehensmodelle, um solche Softwareprojekte zu realisieren. Viele Softwareprojekte scheitern, weil sie auf das falsche Vorgehensmodell gesetzt haben (The Standish Group International 2015).

Die Fragen, welche ich mit meiner Arbeit beantworten möchte, sind daher: Sind die Kunden bzw. Kundinnen von Softwareprojekten auch zufrieden? Gibt es einen Zusammenhang zwischen der Kundenzufriedenheit und dem Weg, der gewählt wurde, um das Projekt abzuschließen? Wie wird Kundenzufriedenheit definiert und wie kann man diese messen? Bei agilen Vorgehensmethoden wird besonders viel Wert auf die Einbindung und Mitarbeit des Kunden oder der Kundin während aller Projektphasen gelegt. Ein Projekt in welchem der Kunde oder die Kundin aktiv mitgearbeitet hat, sollte mehr den erwarteten Funktionen entsprechen, als ein Projekt das lange entwickelt wurde und erst am Ende dem Kunden präsentiert wird. Notwendige Änderungen sollten so frühzeitig vom Kunden erkannt und auch rasch im Programm realisierbar sein (Bleek und Wolf 2008).

Viele IT-Projekte werden bereits mit agilen Vorgehensmodellen entwickelt, obwohl der Projektumfang und der Endtermin des Projektes bereits fixiert sind. Viele Vorteile der agilen Entwicklung, wie etwa Planung in zyklischen Abständen, werden dadurch gar nicht bzw. nicht voll ausgeschöpft (Moksony 2014).

Ist es in Softwareprojekten mit fixiertem Endtermin und Budget sinnvoll auf agile Vorgehensmodelle zu setzen, oder birgt es keine Vorteile, agile Prinzipien zu verfolgen, wenn diese nicht den Projektumfang in irgendeiner Weise verändern?

1.2 Zielsetzung der Arbeit

Viele Softwareprojekte sind in der Vergangenheit gescheitert bzw. scheitern noch immer (The Standish Group International 2015). Aus diesem Grund hat sich die agile Softwareentwicklung als Ansatz in der Softwareprogrammierung entwickelt. Ein agiles Vorgehensmodell bei der Projektrealisierung soll besser auf geänderte Anforderungen während der Umsetzung eines Projektes reagieren können (Cohen, Lindvall und Costa 2003).

Als langjähriger Entwickler und Projektleiter im agilen Umfeld habe ich gesehen, dass das volle Potential von agilen Vorgehensmodellen nur in den seltensten Fällen ausgeschöpft wird. Meist steht vor dem Projektstart das Pflichtenheft, also der Projektumfang, bereits fest. Ebenso gibt es dazu meist einen definierten Endtermin. Im Laufe des Projektes stellt sich jedoch meist heraus, dass die initialen Anforderungen vom Pflichtenheft nicht mehr mit den tatsächlichen Anforderungen der Software übereinstimmen. Agile Vorgehensmodelle bauen genau auf diesen Umstand auf.

Die angesprochenen Umstände hinsichtlich des Projektumfangs in Verbindung mit dem gewählten Vorgehensmodell und den damit verbundenen Tätigkeiten wirft folgende Frage auf:

Sind Kunden von Software-Festpreisprojekten, welche mit agilen Vorgehensmodellen umgesetzt werden, zufriedener als Kunden von Software-Festpreisprojekten, die mit linearen Vorgehensmodellen umgesetzt werden?

Diese Arbeit und die damit verbundenen Analysen sollen zeigen, dass Kunden in der Tat von agilen Vorgehensmodellen profitieren, indem die Kundenzufriedenheit bei solchen Projekten tatsächlich höher ist als bei jenen Projekten, die mit rein linearen Vorgehensmethoden entwickelt wurden.

1.3 Aufbau und methodisches Vorgehen

Zunächst wird anhand vorhandener internationaler Studienergebnisse die Kundenzufriedenheit bei Softwareprojekten analysiert. Welche Softwareprojekte wurden abgeschlossen und welche sind gescheitert. Beeinflusst das gewählte Vorgehensmodell bei der Entwicklung des Programmes auch tatsächlich die Projektabschlussquote?

Welche Vorgehensmodelle wurden bzw. werden bei der Entwicklung des Softwareprojektes verwendet? Des Weiteren werden Kunden und Kundinnen von Softwareprojekten befragt, wie zufrieden sie mit ihrer Software sind, und welche Vorgehensmodelle bei der Entwicklung der Software zur Anwendung kamen, sofern sie in der Lage sind dies zu beantworten.

2. Theoretische Grundlagen

2.1 Vorgehensmodelle

Im Laufe der Zeit wurden Projekte, auch Softwareprojekte, in ihrem Umfang immer größer und komplexer. Die Abarbeitung eines Projektes gestaltete sich zunehmend schwieriger. Um den oft schwer abzuschätzenden Umfang erfolgreich abwickeln zu können, wurden sogenannte Vorgehensmodelle erarbeitet, um Projekte möglichst erfolgreich umzusetzen. Unter einem Vorgehensmodell versteht man vereinheitlichte strukturierte Methoden und Regeln für den bestmöglichen Ablauf von Entwicklungsprojekten (Gnatz, 2005). Vorgehensmodelle versprechen, den Erfahrungsschatz früherer Projekte zu sammeln und dadurch zu helfen, die gegenläufigen Kräfte, Kosten und Qualität geeignet auszubalancieren (Gnatz 2005).

Ein Vorgehensmodell muss jedoch dem aktuellen Stand der Wissenschaft und Technik entsprechen, sowie geeignet für das in Auftrag gegebene Projekt sein. Ständige Abänderungen und Anpassungen von Vorgehensmodellen sind daher unausweichlich (Broy und Rausch 2005).

Grundsätzlich bestehen Vorgehensmodelle aus Aktivitäten und den daraus resultierenden Daten sowie wiederum einer festgelegten Ausführungsreihenfolge der Aktivitäten. Durch diese Vorgaben wird eigentlich auch die zugrundeliegende Strategie von Projekten vorgegeben (Vivenzio und Vivenzio 2013).

In der Literatur wird generell zwischen linearen und agilen Vorgehensmodellen sowie Mischformen davon unterschieden (Engstler et al. 2015).

2.1.1 Lineare Vorgehensmodelle

Lineare Vorgehensmodelle, auch. sequenzielle Vorgehensmodelle genannt, zeichnen sich vor allem dadurch aus, dass der Durchlauf der notwendigen Schritte (Phasen) in aufeinanderfolgenden, abgeschlossenen Phasen passiert. Das Wasserfallmodell dient im Folgenden als Referenz für ein Lineares Vorgehensmodell.

Wasserfallmodell

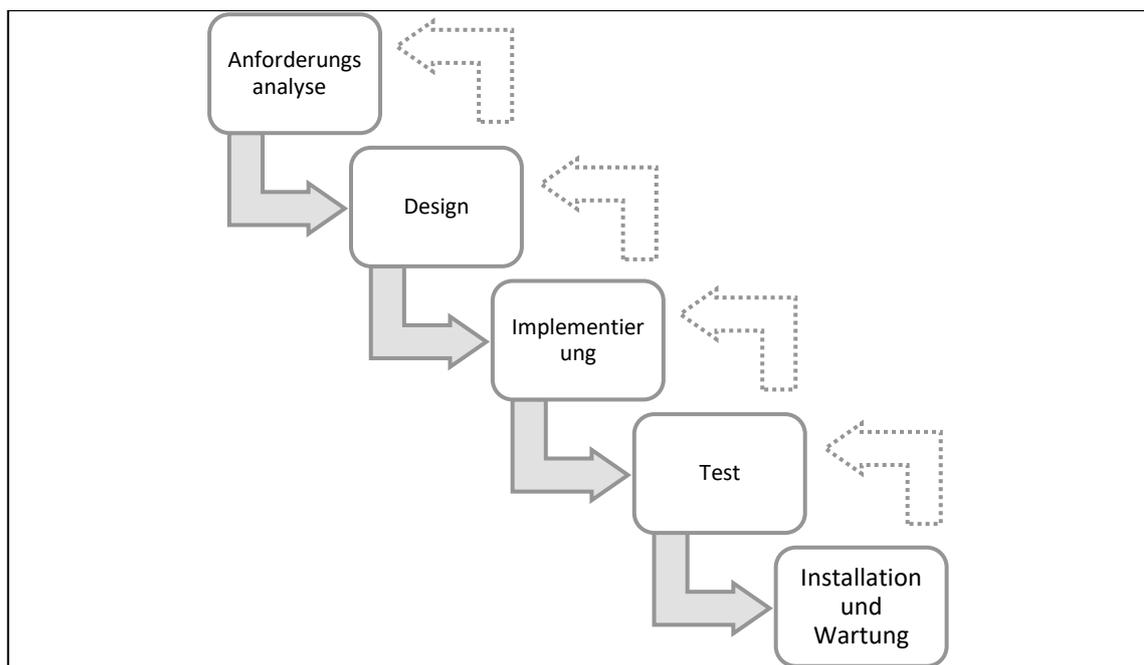
Das Wasserfallmodell ist ein sequenzielles Vorgehensmodell in der Softwareentwicklung, welches den Prozessfortschritt als fließend (ähnlich zu einem Wasserfall) durch eine definierte Anzahl von Phasen durchläuft, um erfolgreich eine Software planen und erstellen zu können (Royce 1970).

In diesem Modell werden nun verschiedene Phasen definiert durchlaufen. Jede Phase muss abgeschlossen sein, bevor man zur nächsten Phase übergehen kann. Aus diesem

Grund ist das Wasserfallmodell rekursiv, da jede Phase endlos wiederholt werden kann, bis die Phase perfektioniert ist (Bassil 2012).

Im Wesentlichen besteht das Wasserfallmodell (siehe Abbildung 1) aus fünf Phasen: Anforderungsanalyse, Design, Implementierung, Test sowie Installation und Wartung (Royce 1970).

Abbildung 1 -Wasserfallmodell



Quelle: Eigene Darstellung in Anlehnung an Royce, 1970 329

Anforderungsanalyse

Nach dem Institute of Electrical and Electronic Engineers (IEEE) wird die Anforderungsanalyse in verschiedene Bereiche geteilt (Fairley 2009):

- Anforderungserhebung
- Anforderungsanalyse
- Anforderungsspezifikation
- Anforderungsbewertung

Zunächst wird mit der Anforderungserhebung erfasst, welche Anforderungen an die Software gegeben sein müssen. Sie ist der erste Schritt, um ein Verständnis zu bekommen, welches Problem die Software überhaupt lösen soll. Eines der Grundprinzipien einer guten Anforderungsanalyseprozesses ist, die effektive

Kommunikation zwischen den verschiedenen Beteiligten. Im Idealfall zieht sich diese Kommunikation über den gesamten Entwicklungsprozess (Abran 2005).

Die Anforderungsanalyse befasst sich mit der Analyse der Anforderungen, um einerseits Konflikte zwischen den einzelnen Anforderungen zu erkennen und diese aufzulösen, andererseits die Grenzen der Software auszuloten. Was ist notwendig? Wie muss ich mit der Umgebung interagieren? Diese Fragen sollten mit der Anforderungsanalyse beantwortet werden. Ebenso sollten aus den Systemanforderungen die Anforderungen an die Software abgeleitet werden (Abran 2005).

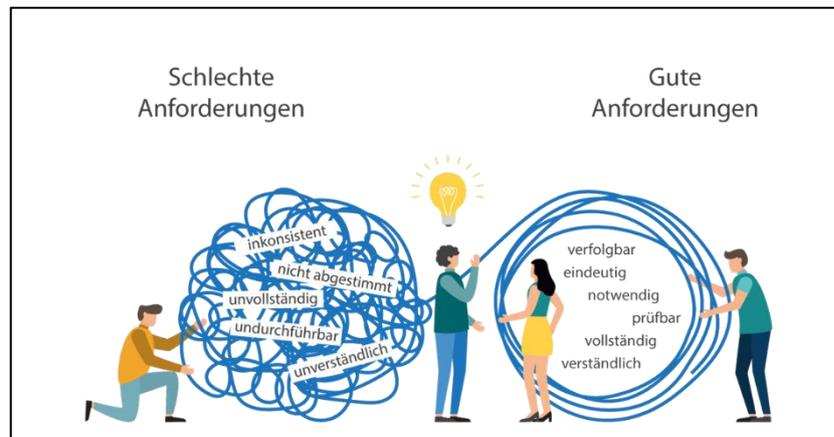
Die Software-Anforderungsspezifikation bildet die Grundlage für eine Vereinbarung zwischen Kunden und Auftragnehmern bzw. Lieferanten. Bei marktorientierten Projekten können diese Rollen von Marketing- und Entwicklungsabteilungen übernommen werden. Darüber hinaus wird vereinbart, was die Software leisten muss (Funktionalitätsumfang) und was nicht. (O A)

Die IEEE Kapitel 4.3 definiert acht Charakteristika einer guten Anforderungsanalyse (O A):

- korrekt
- unzweideutig (eindeutig)
- vollständig
- widerspruchsfrei
- bewertet nach Wichtigkeit und/oder Stabilität
- verifizierbar
- modifizierbar
- verfolgbar (traceable)

Beim letzten Punkt der Anforderungsanalyse werden mit der Anforderungsbewertung die Dokumente, Anforderungen etc. nochmal einer Prüfung unterzogen. Es soll sichergestellt werden, dass Softwareentwickler:innen die Anforderungen verstanden haben. Ebenso wichtig ist, dass die Anforderungsdokumente den Unternehmensstandards gehorchen und dass sie einerseits verständlich, konsistent und vollständig, als auch formal richtig sind. Die verschiedenen Interessensgruppen wie Kunden, Entwickler oder Projektleiter (siehe Abbildung 2) innerhalb des Projektes sollten die Dokumente überprüfen, und so sicherstellen, dass die Anforderungsdokumente, auf denen sich das Projekt stützt, ihre Richtigkeit haben (Abran 2005).

Abbildung 2 - Gute und Schlechte Anforderungen analysieren



Quelle: Krumbholz 2020

Design

Nachdem die Anforderungen abgeklärt sind, folgt in nächsten Schritt das Design, oder anders gesagt, die Architektur der Software. Softwarearchitektur ist eine strukturierte oder hierarchische Anordnung der Systemkomponenten sowie Beschreibung ihrer Beziehungen (Balzert 2001).

Im Design erarbeiten Softwareentwickler:innen und Softwaredesigner:innen die Softwarearchitektur, Datenbankkonzepte, Algorithmen, Diagramme sowie das Aussehen des User Interfaces (Bassil 2012).

Implementierung

In diesem Schritt werden nun die Anforderungen mit dem Design verknüpft und ein lauffähiges Programm, Datenbanken, Websites oder einzelnen Softwarekomponenten mittels Programmieren erzeugt. Am Ende der Phase hat man ein ausführbares Produkt in Form eines Programms (Bassil 2012).

Test

In der Testphase werden nun die in den vorherigen Phasen umgesetzten Komponenten bzw. das ganze System getestet. Dies kann intern durch eigens geschulte Tester oder auch durch den Auftraggeber bzw. der Auftraggeberin passieren (Vivenzio und Vivenzio 2013).

Auslieferung und Wartung

In der letzten Phase des Wasserfallmodells wird die Softwarelösung an den Kunden und Kundinnen übergeben und in Betrieb genommen. Zusätzliche Wartungsarbeiten für die Softwarequalität oder der Einführung neuer Features für den Kunden bzw. der Kundin werden hier ebenso getätigt (Bassil 2012).

2.1.2 Agile Vorgehensmodelle

Ein weiterer Teilbereich von Vorgehensmodellen sind agile Vorgehensmodelle. Agile Vorgehensmodelle hegen in der Softwareentwicklung aktuell großer Beliebtheit (Aston 2021).

Anforderungen und Software entwickeln sich stetig weiter. Eine Vorhersage bzw. einen detaillierten Plan über 3 Jahren aufzustellen, ist daher unmöglich. Sogar wenn es möglich wäre, hat es keinen Mehrwert auf die Zeit dazwischen, weil man sehr viele verschiedene Dinge benötigen würde. Ein Plan muss also laufend adaptiert bzw. agil abgeändert werden können (Beck und Fowler 2001).

Aus diesem Grund hat sich das Agile Manifest entwickelt.

Agiles Manifest

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

- **Individuen und Interaktionen** sind wichtiger als Prozesse und Werkzeuge
- **Funktionierende Software** ist wichtiger als umfassende Dokumentationen
- **Zusammenarbeit mit dem Kunden** ist wichtiger als Vertragsverhandlungen
- **Reagieren auf Veränderung** ist wichtiger als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein“ (Beck et al. 2001).

Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge

Die Menschen innerhalb eines Projektteams sowie deren Kommunikation und Interaktionen sind wichtiger anzusehen als die Prozesse und Tools, die eingesetzt werden. Dies beinhaltet auch den agilen Prozess. Das Wichtigste in einem Projekt sind also die Individuen, die das Projekt abarbeiten (Bleek und Wolf 2008).

Funktionierende Software sind wichtiger als umfassende Dokumentationen

Mit diesem Punkt wird beschrieben, dass es wichtiger ist, eine laufende Software an den Kunden bzw. der Kundin zu liefern als dessen Dokumentation. Dieser Punkt wird jedoch oft missverstanden und die Dokumentation als irrelevant eingestuft. Dies ist jedoch nicht der Fall. Dokumentation ist ebenfalls wichtig, man sollte sich initial jedoch nur auf jenes konzentrieren, welches auch einen Mehrwert für den Kunden bzw. der Kundin bietet (LeMay und Langenau 2019).

Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlungen

Der dritte Punkt des Manifests beschreibt die Beziehung zwischen den Personen, die eine Software haben möchten, und denjenigen die sie schreiben. Wichtig ist, dass es im agilen Umfeld keine „wir“ und „sie“ Partner:innen gibt, sondern ein „uns“. Diese Zusammenarbeit ist wichtig, da nur mit guter Kommunikation und Kollaboration ein gutes Produkt entsteht. Verträge haben ihren Nutzen, jedoch stärkt gute Zusammenarbeit das Zusammengehörigkeitsgefühl. Dies kann so weit gehen, dass ein Vertrag gar nicht mehr von Nöten ist (Cockburn 2002).

Reagieren auf Veränderung ist wichtiger als das Befolgen eines Plans

Wenn all die Prinzipien zuvor eingehalten werden, so muss man sich bewusst sein, dass sich Dinge verändern. Erst bei der Entwicklung von Software und im laufenden Zustand von Software wird deutlich, wo noch Veränderungen oder Verbesserungen an das laufende Programm notwendig sind. Mit fortschreitender Dauer des Projekts lernen alle beteiligten Personen dazu. So ist zu erwarten, dass man während der Durchführung eines Projektes Dinge abändern muss (Bleek und Wolf 2008).

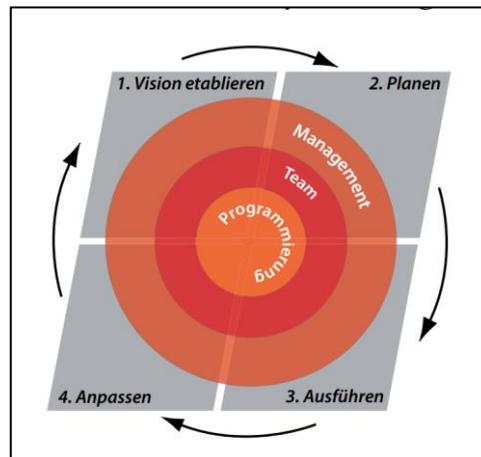
Methoden der agilen Softwareentwicklung

Die Prinzipien sind als theoretische Grundlage zu verstehen. Mit folgenden Methoden werden die agilen Prinzipien angewandt.

Iterationen

Die Hauptmethodik der agilen Softwareentwicklung ist iteratives, zyklisches Vorgehen. Dies sollte auf allen Ebenen z.B. Management, Team und Programmierung durchgezogen werden. Im Gegensatz zu klassischen Vorgehensmethoden werden nicht alle Einzelheiten im Voraus mit allen Details geplant. In kurzen Zyklen (oft auch Sprints genannt) wechseln sich Detailplanung und Programmierung miteinander ab. Aus diesem Grund kann man sehr rasch auf Änderungen hinsichtlich der Kunden und Kundinnen oder des Managements reagieren. Eine Iteration (siehe Abbildung 3) besteht immer aus einer Vision, einem Plan, der Ausführung des Plans sowie der Anpassung dessen für die nächste Iteration. (Bleek und Wolf 2008).

Abbildung 3 - Iterationen



Quelle: Bleek und Wolf, 2008 3

Kurze Releasezyklen

Das System kann schnell in Produktion gebracht werden, indem die Releasezyklen klein gehalten werden. Die Version sollte aber sinnvoll geschnitten werden und nicht halb implementiert ausgeliefert werden, nur um den Releasezyklus zu verkürzen (Beck 2005).

Pair Programming

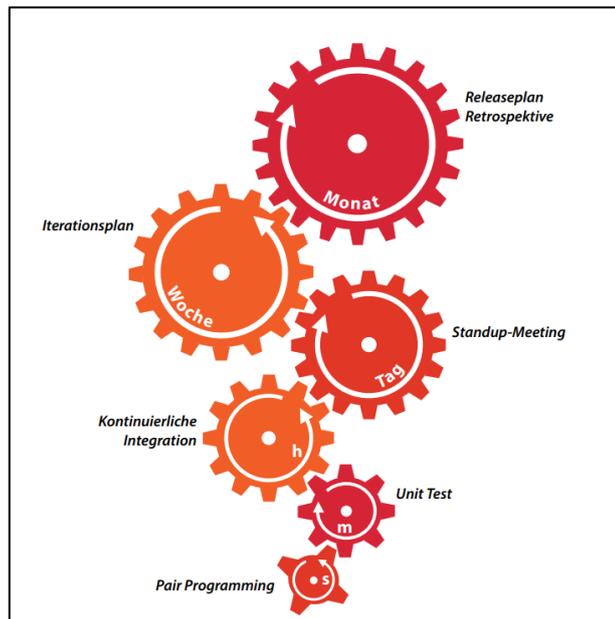
Pair Programming (deutsch Paarprogrammierung) bedeutet, dass der Applikationscode nicht nur von einer Person, sondern von zwei Personen geschrieben wird. Der besondere Umstand daran ist jedoch, dass beide mit nur einer Tastatur und einer Maus am selben PC arbeiten. Jede Person hat dabei eine eigene Rolle, die im Laufe der Zeit getauscht wird. Jene Person an der Tastatur programmiert, der Partner denkt weiter und überlegt, ob der gewählte Ansatz richtig ist, und ob dieser auch gut testbar ist (Beck 2005).

Zyklische Kommunikation

Zusätzlich zu den Iterativen Planungsmeetings gibt es eine Reihe von weiteren möglichen Meetings, um die Kommunikation zu erhöhen und zu verbessern. Neben dem Sprintzyklus gibt es auch tägliche Feedbackschleifen. In Scrum werden diese Meetings Daily Scrum genannt. Sie dienen dem Erfahrungsaustausch im Team und sollten höchstens 15 Minuten dauern. Auf diese Weise bleibt jedes Teammitglied über die Aufgaben und Tätigkeiten der Kollegen im Bilde. Probleme und Verzögerungen können so frühzeitig erkannt werden (Bleek und Wolf 2008).

Die einzelnen Bausteine können als Zahnräder (siehe Abbildung 4), die ineinandergreifen, verstanden werden. Die einzelnen Mechanismen sind in unterschiedlicher Dauer dargestellt (Bleek und Wolf 2008).

Abbildung 4 - Bausteine agiler Softwareentwicklung



Quelle: Bleek und Wolf, 2008 10

2.1.3 Mischformen

Wagile

Im Laufe der Zeit wurden klassische Wasserfallmodelle durch den Konkurrenzdruck obsolet und agile Vorgehensmodelle wurden als Standardvorgehensmodell in Unternehmen etabliert. Aber auch 20 Jahre nach dem Agilen Manifest stellt die Umstellung auf agile Vorgehensmethoden ein gewisses Risiko dar. Oft kam es dabei zu einem Hybridmodell namens „Wagile“. Wagile setzt sich aus dem sequenziellen Wasserfallmodell sowie dem agilen Konzept zusammen (Van Schaik 2021).

Wagile wird eher als negativ betrachtet und ist ein Produkt aus mehreren möglichen Problemen (Van Schaik 2020):

- Ein fehlgeschlagener Übergang zu agilen Vorgehensmethoden. Dies passiert, wenn Teile des Teams bzw. des Managements den agilen Prozess nicht voll unterstützen bzw. Teile davon falsch verstehen.
- Das Management kann bei der Umstellung Steine in den Weg legen. Diese können bewusst aber auch durch unbedachte Vorgehensweisen zu Stande kommen.
- Alle sind gewillt auf agile Methoden umzustellen, jedoch fehlt das Wissen oder die notwendige Infrastruktur, um eine vollständige Umstellung zu gewährleisten.

Hybride Vorgehensmodelle

Im weiteren Sinne werden hybride Ansätze in der Softwareentwicklung für verschiedenste Typen von Vorgehensmodellen verwendet (z.B. Scrumban, ist eine Mischform von Scrum und Kanban), jedoch ist damit meist die Verbindung von agilen Modellen mit traditionellen linearen Vorgehensmodellen gemeint (Kneuper 2018).

Das hybride Modell (auch Agiles-Wasserfall-Modell) zielt darauf hin, die Abhängigkeitsverfolgung und die Klarheit von Wasserfall beizubehalten, sowie auch die Stärken der agilen Methodik zu verwenden. Vom agilen Ansatz werden die notwendige Flexibilität und Transparenz, damit schnell ändernde Anforderungen der Interessengruppen abgedeckt werden können, hinzugezogen. Das Modell (oft auch Agile-Waterfall-Hybrid) sollte in folgenden Szenarien angewandt werden (Moksony 2014):

- Das Produkt besteht aus Hardware als auch aus Softwarekomponenten mit gleicher Priorität.
- Das Produkt ist eine Software, welche komplexe Back-End und Front-End Technologie mit sich bringt.

Der dritte und möglicherweise häufigste Punkt ist nicht speziell auf die Charakteristik des Projekts ausgelegt. In den meisten Fällen versucht man den Kundenwünschen gerecht zu werden. Da sie das Produkt in der Regel auch bezahlen, will man die Bedürfnisse des Kunden stets befriedigen. Natürlich wollen Kunden und Kundinnen auch Planungssicherheit haben, daher werden die Gesamtkosten in einer initialen Planungsphase abgeschätzt. Der Wasserfall Ansatz könnte also für die Architekturplanung und initiale Anforderungsdefinition, und die agile Methode für die Entwicklung und das Testing verwendet werden (Moksony 2014).

2.2 Softwareentwicklung

Die Entwicklung einer Software wird gerne mit einem Hausbau verglichen, und es gibt hier durchaus einige Gemeinsamkeiten. Wie im Kapitel 2.1 Vorgehensmodelle ersichtlich, konzipiert man aufgrund von Anforderungen einen Plan, um diesen anschließend umzusetzen. Die Pläne werden dann nach und nach umgesetzt, bis ein bewohnbares Haus bzw. eine fertige Software entsteht (Brandt-Pook und Kollmeier 2008).

2.2.1 Aktuelle Standards in der Softwareentwicklung

Architektur

Die Software-Architektur einer Anwendung bzw. eines Computerprogramms beschreibt die Struktur des Systems. Diese besteht aus einzelnen Elementen, den nach außen

sichtbaren Eigenschaften dieser Elemente und deren Beziehungen zueinander (Bass, Clements und Kazman 2003).

Für diese Architektur gibt es nun verschiedene Architekturmuster.

Microservices

Für Microservices gibt es keine allgemein anerkannte Definition, jedoch sind Microservices unabhängig deploybare (deployen = ein Artefakt auf einem System einspielen) Module. Beispielsweise kann eine E-Commerce Software in verschiedene Bereiche wie Bestellprozess, Registrierung oder auch Artikelsuche aufgeteilt werden. In einem großen Programm wären alle Dinge gemeinsam definiert, jedoch müsste das gesamte System bei einem Update heruntergefahren werden. Mit der Microservice-Architektur ist nun jeder Bereich ein eigenes Modul und kann unabhängig von den anderen Modulen neu auf ein System eingespielt werden, ohne die anderen Funktionen zu beeinträchtigen (Wolff 2018).

Vor allem bei modernen Webanwendungen könne so vermieden werden, dass die komplette Anwendung während Updates nicht erreichbar wäre. Dies wäre für YouTube bzw. Netflix nicht vorstellbar (Oliver 2016).

Cloud Computing

Cloud Computing (oft auch nur als Cloud bezeichnet) ist eine Methode, bei der die Daten welche verarbeitet bzw. gespeichert werden, nicht mehr lokal gespeichert, sondern auf Servern, die mit dem Internet verbunden sind, ausgelagert werden (Singer 2010).

„Grundprinzip ist das Ausgliedern von Software- oder sogar Hardwarefunktionen der Anwender, so dass in vielen Fällen gar nicht mehr genau feststellbar ist, wo sich die ausgelagerten Informationen oder Anwendungen „in der Wolke“ befinden. Die zugrundeliegende Technologie ist nicht neu, doch die Konsequenzen für die Geschäftsmodelle von IT-Anwendern und -Anbietern sind kaum zu überschätzen: IT-Leistungen werden in Echtzeit als Service über das Internet bereitgestellt und nach Nutzung abgerechnet. Der Zugriff selbst erfolgt in der Regel über eine allgemeine verfügbare Standardanwendung, zumeist einen Webbrowser“ (Singer 2010).

Durch diesen Vorgang können bei Bedarf, zum Beispiel bei vielen Anfragen an einen Server, zusätzliche Ressource freigegeben werden. Dies muss natürlich bei der Programmierung von Anwendungen berücksichtigt werden (Singer 2010).

Single Page Applications

Single Page Applications (oft auch Single-Page-Webanwendung oder SPA) sind wie der Name schon verrät, eine JavaScript getriebene Web-Applikation, welche genau aus einer einzigen HTML Seite besteht. Weitere Seiten bzw. Daten werden im Nachgang dynamisch asynchron nachgeladen (Flanagan 2011).

Dies ermöglicht kurze Ladezeiten für Webanwendungen, da nur jene Daten geladen werden, die gerade benötigt werden. Um dies zu bewerkstelligen, benötigt es eine Client-Server Architektur, wobei die Single Page Applications auf der Userseite im Webbrowser ausgeführt werden und die Daten vom Server nachgeladen werden (Flanagan 2011).

Versionskontrollsysteme

Bei der Erstellung von Software, schreiben Softwareentwickler:innen Quellcode. Dieser kann geändert, erweitert und rückgängig gemacht werden. Wenn nun Entwickler:innen auf eine Codedatei zugreifen wollen, können gleichzeitige Zugriffe auf den Code zu einem Problem werden. Versionskontrollsysteme (engl. Version Control Systems, VCS) erlauben mit speziellen Workflows solch einen Zugriff (Otte 2009).

VCS verarbeiten und erkennen Änderungen als sogenannte Revisionen. Jeder Softwareentwickler:in kann pro Tag beliebig viele Revisionen mit einem Commit erzeugen. Mithilfe dieser Revisionen hat man Zugriff zu vergangenen und geänderten Codeständen in einem Log (Loeliger und McCullough 2012).

2.2.2 Continuous Integration

Der Prozess einer Softwareintegration ist kein neues Problem. Software zu integrieren ist bei Ein-Mann-Projekten oft keine Schwierigkeit, aber mit steigender Größe und Komplexität von Projekten wird dies zunehmend ein Problem. Dies kann auch schon dadurch passieren, dass eine weitere Person im Projekt mitarbeitet. In diesem Fall muss sichergestellt werden, dass die einzelnen Softwarekomponenten miteinander funktionieren, zeitnah und öfters passieren. Wenn man bis zum Schluss eines Projektes mit der Integration der einzelnen Komponenten wartet, kann dies zu Qualitätsproblemen führen (Duvall, Matyas und Glover 2007).

Continuous Integration ist nun eine Methodik mit deren Mitglieder eines Teams ihre Arbeit permanent bzw. kontinuierlich, im Regelfall zumindest täglich, in ein Integrationssystem bringen. Jede Integration in ein solches System wird mit einem automatischen Bild auf Fehler und Integrationsfehler so schnell wie möglich überprüft. Viele Teams finden, dass solch ein Ansatz zu signifikant reduzierten Integrationsproblemen führt und so dem Team erlaubt, eine zusammenhängende Software schneller zu entwickeln (Fowler 2006).

Ein Build ist ein Prozess, der Entwicklungscode zu Maschinencode kompiliert und dabei weitere Schritte wie Softwaretests, Codeinspektion automatisiert ausführt. Dies bringt einige Vorteile mit sich (Duvall, Matyas und Glover 2007):

- Alle Entwickler können lokal auf ihrem PC einen Build starten und so ihren Code verifizieren, bevor sie ihre Arbeit auf das Integrationssystem übertragen.
- Entwickler müssen ihren Code zumindest einmal täglich übertragen
- Integrationsbuilds werden täglich automatisiert an einem eigens dafür vorgesehenen Build-System gestartet.

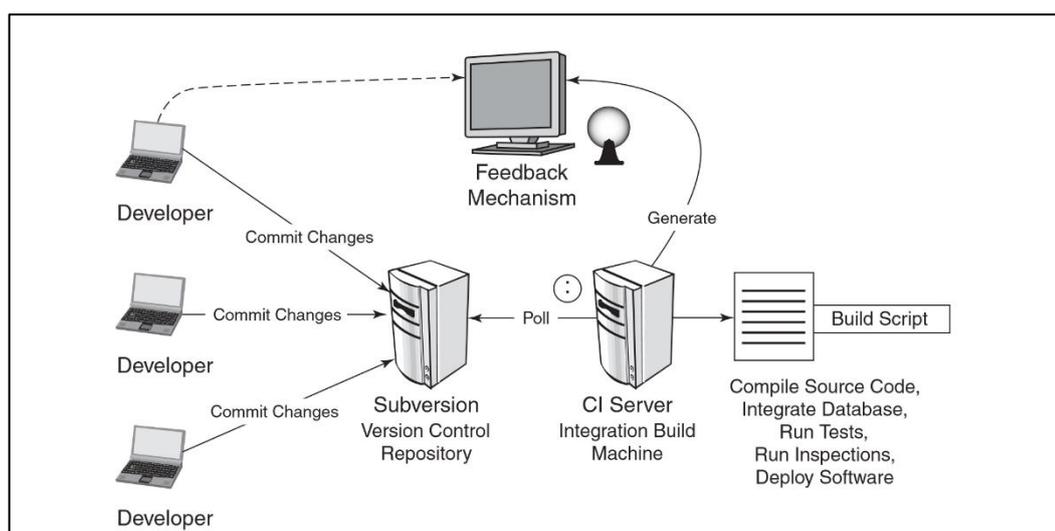
- Alle Tests müssen bei jedem Build durchlaufen.
- Ein Artefakt (z. B. Ausführbares Programm) wird gebaut und kann funktional getestet werden.
- Fehlgeschlagene Builds werden mit höchster Priorität wieder zum Laufen gebracht.
- Automatisiert generierte Reports können von Entwicklern verwendet werden, um Analysen über den Code zu machen. Diese kann für Verbesserungen am Code verwendet werden.

Grundsätzlich braucht Continuous Integration, um zu funktionieren, kein bestimmtes Toolset. Es hat sich jedoch von Vorteil erwiesen, einen eigenen Build-Server zu verwenden (Fowler 2006).

Die notwendigen Schritte in einer CI Umgebung sehen typischerweise (siehe Abbildung 5) folgendermaßen aus (Duvall, Matyas und Glover 2007):

- Zuerst committed ein Entwickler Code in ein Versionskontrollsystem. Der CI-Server prüft in der Zwischenzeit in vorgegebenen Zeitabständen Änderungen am Versionskontrollsystem.
- Nachdem ein Commit erfolgt ist, erkennt der Build-Server Änderungen am Versionskontrollsystem und holt sich die letzte Code-Version davon und startet ein Build-Skript, welches das System integriert und baut.
- Der CI-Server generiert Feedback für die Entwickler, indem er die Ergebnisse des Builds über Email an die Projektteilnehmer:innen bzw. Teammitglieder:innen schickt.
- Der CI-Server wartet nun weiterhin auf Änderungen im Versionskontrollsystem.

Abbildung 5 - Ablauf CI



Quelle: Duvall, Matyas und Glover, 2007 5

2.2.3 Softwarequalität

Es gibt keine allgemeine Definition von Softwarequalität. In der Literatur wird es als ein mehrdimensionales Konstrukt aus verschiedenen Eigenschaften beschrieben.

Hewlett Packard (HP) entwickelte dazu das FURPS als Beispiel für ein Softwarequalitätsmodell, welches verschiedene Punkte berücksichtigt (Grady und Caswell 1987):

- Functionality (Funktionalität)
- Usability (Gebrauchstauglichkeit)
- Reliability (Zuverlässigkeit)
- Performance (Leistung)
- Supportability (Wartbarkeit)

Zusätzlich zu den definierten Werten ist jedoch zu beachten, dass nicht für jede Software die gleichen Qualitätsanforderungen gelten.

„Jedes Unternehmen, das Software entwickelt, bemüht sich, die beste Qualität auszuliefern. Man kann ein Ziel aber nur dann nachweisbar erreichen, wenn es präzise definiert ist, und das gilt für den Begriff „beste Qualität“ nicht. Software-Qualität ist facettenreich. Viele Eigenschaften einer Software ergeben gemeinsam die Software-Qualität. Nicht alle diese Eigenschaften sind gleichermaßen für den Benutzer und den Hersteller einer Software wichtig. Einige Eigenschaften sind für ein bestimmtes Software-Produkt besonders wichtig, andere für das gleiche Produkt vollständig irrelevant“ (Liggesmeyer 2009).

Generell kann man bei Softwarequalität zwischen einer externen und internen Qualität unterscheiden (Bergmann und Pribsch 2010).

Externe Qualität

Externe Qualität beschreibt den Nutzen der Software für den Endkunden, und daher nur für jene Qualitätsaspekte, die für ihn relevant sind. Diese sind unter anderem (Bergmann und Pribsch 2010):

- **Funktionalität:** Die Fähigkeit eines Programms, die Aufgaben entsprechend den Erwartungen bzw. der Programmanforderungen zu erfüllen.
- **Gebrauchstauglichkeit:** Das Programm sollte einfach und effizient sowie benutzerfreundlich und barrierefrei bedienbar sein.
- **Reaktionsfreudigkeit:** Die Anwendung sollte in einer entsprechenden Antwortzeit reagieren. Dies ist oft ein ausschlaggebender Grund für Erfolg und Misserfolg eines Produktes.
- **Sicherheit:** Ein wesentlicher Bestandteil von externer Qualität ist die Sicherheit der Benutzerdaten und der Eingaben.

- **Verfügbarkeit und Zuverlässigkeit:** Dieser Punkt beschreibt die Fähigkeit, dass Programme auch unter großer Last erreichbar sind, und auch unter diesen Umständen wie erwartet funktionieren.

Interne Qualität

User von Programmen verwenden diese, ohne die interne Aspekte zu kennen. Softwareprogrammierer:innen sind jedoch für die Programmierung und Herstellung der Software zuständig und daher an der internen Softwarequalität interessiert (Liggesmeyer 2009).

Im Gegensatz zur externen Qualität beschreibt die interne Qualität die erforderlichen Merkmale, die für Entwickler:innen während der Programmierung relevant sind. Für Softwareprogrammierer:innen ist es wichtig, einen einfachen und strukturierten Code zu schreiben, sodass dieser Code auch von anderen Personen gut lesbar, veränderbar und erweiterbar ist. Die interne Qualität ist zwar nicht direkt für Endbenutzer:innen wahrnehmbar, jedoch wirkt sich diese auf längere Sicht negativ auf die externe Qualität aus, wenn diese nicht gegeben ist (Bergmann und Pribsch 2010).

Manager:innen in der Softwareentwicklung stehen eigentlich als Bindeglied zwischen Kunden und Kundinnen und Entwickler:innen und haben so ein Interesse auf die interne als auch auf die externe Qualität eines Produktes (Liggesmeyer 2009).

Das bereits angesprochene „Continuous Integration“ in Kapitel 2.2.2 ist ein mögliches Tool, um die interne Qualität zu gewährleisten.

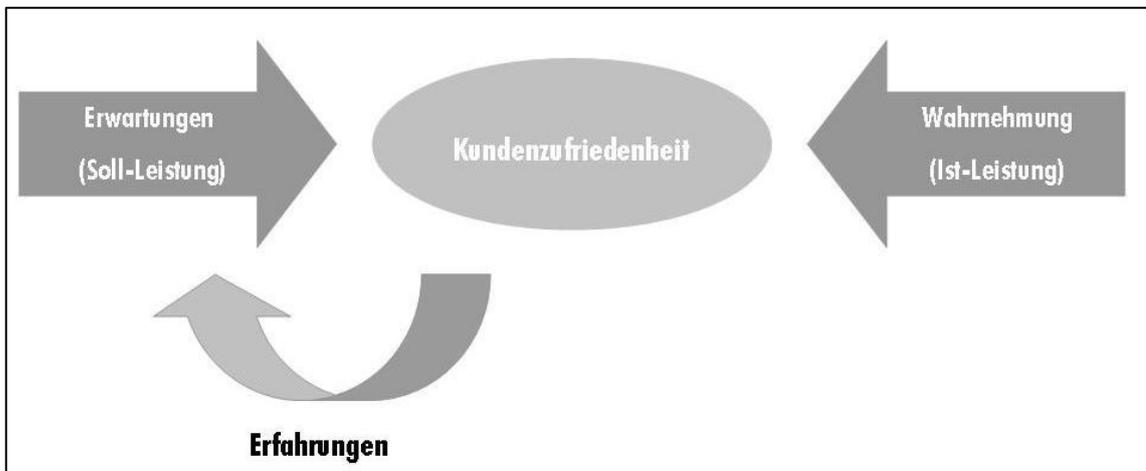
2.3 Kundenzufriedenheit

2.3.1 Definition

Kundenzufriedenheit ist schon seit Jahrzehnten Thema von wissenschaftlichen Forschungen und Arbeiten, jedoch hat sich bis heute keine einheitliche Definition durchgesetzt. Die meisten Autoren und Autorinnen stimmen jedoch überein, dass Kundenzufriedenheit ein Nachkaufphänomen ist, welches widerspiegelt, wie Kunden und Kundinnen ein gekauftes Produkt oder eine in Anspruch genommene Dienstleistung beurteilen (Faullant 2007).

Auf der anderen Seite können unzufriedene Kund:innen aufgrund von Abwanderung oder negativer Mundpropaganda Firmen bzw. Produkthersteller:innen vor erhebliche Probleme stellen (Festge 2007).

Abbildung 6 - Kundenzufriedenheit



Quelle: Foerster, 2015

Um Kundenzufriedenheit messen zu können, benötigen wir Kennzahlen wie zum Beispiel den Customer Satisfaction Score, welcher zu einem späteren Zeitpunkt noch behandelt werden (Natrapei 2019). Siehe Kapitel 3.3 Evaluationskriterien.

2.3.2 Stand der Wissenschaft und Technik zur Kundenzufriedenheit und dem eingesetzten Vorgehensmodell

Wie bereits erwähnt, kann das Vorgehensmodell einen direkten Einfluss auf die externe Qualität bedeuten. Um zu verstehen, welchen Einfluss das Vorgehensmodell auf die Kundenzufriedenheit hat, muss man die theoretischen Grundlagen bezüglich den verschiedenen Vorgehensmodellen festhalten. Damit Kundenzufriedenheit für die Bewertung relevant sein kann, muss das Softwareprojekt erfolgreich abgeschlossen und im Einsatz sein.

Erfolgreiche Softwareprojekte

Um zu bestimmen welche Softwareprojekte erfolgreich sind, müssen messbare Faktoren für ein erfolgreiches Projekt definiert werden (Huber 2011).

Ein Softwareprojekt gilt im Konsens dann als erfolgreich, wenn die im Vorfeld vereinbarten Ziele erreicht wurden. Je nach Definition sind in der Regel die Anforderungen in der verfügbaren Zeit und im verfügbaren Budgetrahmen zu erledigen. Diese können sich im Laufe des Projektes verändern oder präzisiert werden bzw. sogar zum Teil drastisch vom Ursprungsplan abweichen (Huber 2011).

Neben diesen Grundprinzipien gibt es jedoch weitere Parameter, die sich an den Erfolg eines Projektes knüpfen, jedoch hängt dies meist vom Standpunkt der jeweilig involvierten Parteien ab. So könnte zum Beispiel erst dann ein Projekt als erfolgreich deklariert werden, wenn auch Auftraggeber:innen mit dem Ergebnis zufrieden sind (Angermeier 2005).

Ob zum Beispiel Verkaufszahlen eines Produktes in einigen Jahren ein Kriterium für dessen Erfolg sind, sollte nicht Teil des Projekterfolges sein. Es kann sein, dass sich der Markt oder Kundenwünsche in der Zwischenzeit erheblich verändert haben (Huber 2011).

Laut dem Chaos Report 2015 (siehe Abbildung 7) wurden 2015 nur 29 % aller IT Projekte erfolgreich abgeschlossen. 52 % wurden teilweise erfolgreich, und 19 % gar nicht abgeschlossen (The Standish Group International 2015).

Abbildung 7 - Erfolgreiche IT Projekte

MODERN RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011-2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Quelle: The Standish Group International, 2015 2

Auch im Chaos Report wird ein Projekt dann als erfolgreich deklariert, wenn das Projekt rechtzeitig, innerhalb des Budgets mit dem initial definierten Umfang abgeschlossen wurde. Wenn es zu Kostenüberschreitungen und/oder Terminüberschreitungen bzw. zu Änderungen im Funktionsumfang kam, so wird das Projekt als teilweise erfolgreich deklariert. Ein Projekt, welches nie in einem Produktiveinsatz war, gilt als nicht erfolgreich (The Standish Group International 2015).

Dies umfasst alle Projektgrößen (Siehe Abbildung 8), jedoch ist deutlich zu erkennen, dass mit steigender Komplexität und Projektgröße auch weniger Projekte als erfolgreich gelten.

Abbildung 8 - Erfolgreiche IT Projekte nach Größe

CHAOS RESOLUTION BY PROJECT SIZE			
	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	100%	100%	100%

The resolution of all software projects by size from FY2011-2015 within the new CHAOS database.

Quelle: The Standish Group International, 2015 3

Vorgehensmodelle und Kundenzufriedenheit

Um die Kundenzufriedenheit messen zu können, muss ein Projekt als erfolgreich bzw. teilweise erfolgreich abgeschlossen worden sein. Erst in einem Produktiveinsatz kann evaluiert werden, ob die Software den Vorstellungen entspricht. Dieser Umstand beruht darauf, dass die Kundenzufriedenheit als externer Qualitätsfaktor erst beurteilt werden kann, wenn das Projekt fertiggestellt wurde (Bergmann und Priebisch 2010).

Basierend auf den verschiedenen Vorgehensmodellen und deren Zusammenarbeit mit den Kunden und Kundinnen ergeben sich verschiedene Faktoren, die man berücksichtigen kann (Andrei et al. 2019).

Bei linearen Vorgehensmodellen können verschiedene Phasen in einem Projekt reevaluiert werden. Dies kann zu zusätzlichen Kosten sowie zu einer Verschiebung der initial geplanten Ziele führen. Ebenso könnte sich dieser Umstand negativ auf das Endprodukt und die Kundenzufriedenheit auswirken. Damit dies nicht eintritt, können kleine Softwareprojekte, bei welchen der Termin, Ziele und Budget vorgegeben sind (Fixpreisprojekte) durchaus von Vorteil für Kunden und Kundinnen sein. Es könnte dazu führen, dass das Team das Projekt sehr gut geplant hat und daher sehr organisiert ist. Ebenso besteht die Möglichkeit einen Ausfall eines Teammitglieds besser kompensieren zu können, da alles sehr gut dokumentiert ist und somit direkt an der Arbeit des ausgefallenen Mitglieds angesetzt werden kann. Das Design steht von Anfang an fest, und Kunden und Kundinnen sind zufrieden, weil sie genau das bekommen, was definiert worden ist (Andrei et al. 2019).

Dies gilt allerdings nicht als gegeben. Eine Studie aus Südafrika (Iyawa, Herselman und Coleman 2016) kommt zu dem Schluss, dass mit dem Wasserfallmodell die Kundenzufriedenheit niedriger ist als mit agilen Vorgehensmodellen. Kunden und Kundinnen wissen oft nicht initial welche Anforderungen an die Software gegeben sein müssen.

Nach dem aktuellen Stand der Wissenschaft und Technik sind also agile Methoden klar im Vorteil. Mit folgenden Methoden kann die Kundenzufriedenheit durch den Einsatz von agilen Vorgehensmethodiken positiv beeinflusst werden (Amirova et al. 2019):

- Klare Ein- und Abgrenzung der Rolle des Kunden
- Aktive Beteiligung des Kunden und Kundin während des Projektes. Besonders wichtig ist hier die aktive Kommunikation zwischen Kunde bzw. Kundinnen und Teammitgliedern. Dies erhöht die Zufriedenheit und stärkt das Zusammengehörigkeitsgefühl.
- Möglichkeit von frühem und stetigem Kundenfeedback.
- Die Fähigkeit dem Kunden oder der Kundin ein Feature zu liefern, welches sie benötigen.

- Langfristige Zusammenarbeit von Kunden bzw. Kundinnen und dem Entwicklungsteam, um auch nach einem erfolgreichen Projektabschluss weiterhin neue Funktionen einzubauen.

3. Konzeptionelles Vorgehen und Lösungsansatz

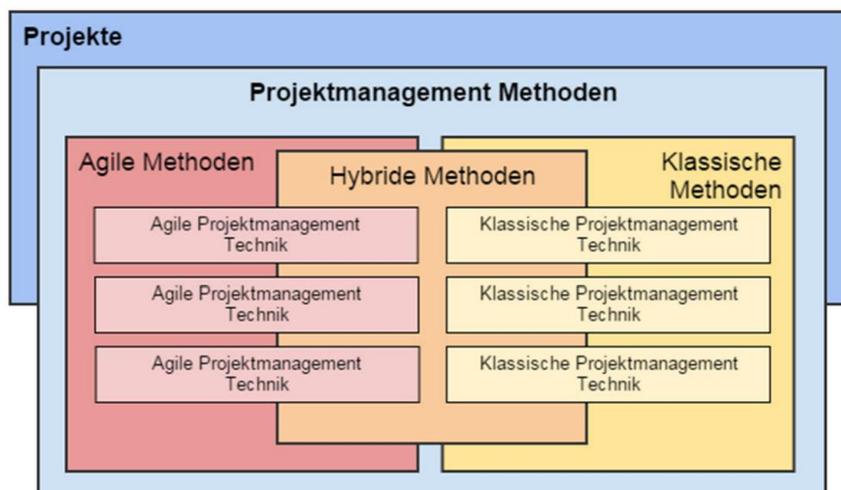
Damit die Hypothese bestätigt oder widerlegt werden kann, wird eine Marktforschungsmethode in Form einer Umfrage angewandt. Zusätzlich zum Fragebogen gibt es eine weitreichende Literaturstudie über abgeschlossene Softwareprojekte und dem darin verwendeten Vorgehensmodell. Im Speziellen wird untersucht, welchen Einfluss hybride Vorgehensmodelle, also eine Mischung aus linearen und agilen Vorgehensmodellen, auf das Ergebnis haben.

3.1 Literaturstudie der Softwareprojekte

Im Kapitel 2.3.2 wurde bereits untersucht, welcher Zusammenhang zwischen erfolgreichen und gescheiterten Softwareprojekten hinsichtlich verwendeten Vorgehensmodells besteht. Um ein Softwareprojekt erfolgreich abzuschließen, sind wie bereits oben erwähnt, auf jeden Fall agile Methodiken von Vorteil (The Standish Group International 2015).

Um jedoch die Kundenzufriedenheit messen zu können, muss man einen Schritt weiter gehen als bisher. Ein erfolgreich abgeschlossenes Softwareprojekt, also ein im Produktiveinsatz befindliches, ist die Grundbedingung, um Kundenzufriedenheit messen zu können (Bergmann und Pribsch 2010). Untersucht wird, ob Programme, welche mit agilen Softwareprojekten abgeschlossen wurden, die Kundenzufriedenheit erhöhen können. Ebenso wird geprüft, ob der Grundaufbau des Projekts linear war, und nur agile Ansätze in der Softwareentwicklung verwendet wurden. So kann zum Beispiel das Projektmanagement und das Pflichtenheft linear sein, in weiterer Folge die Entwicklung jedoch agil gestaltet sein (siehe Abbildung 9).

Abbildung 9 - Vorgehensmodelle - Agil, Klassisch (Linear), Hybrid



Quelle: Mikusz et al., 2018 74

3.2 Kriterien und Themen für Fragebogen

Zusätzlich zu der Literaturstudie wird ein Fragebogen verwendet, um empirische Ergebnisse hinsichtlich der Zufriedenheit mit einem Softwareprodukt und dem verwendeten Vorgehensmodell aufzuzeigen.

Wichtig bei einem Fragebogen ist, worüber eine Aussage getroffen werden soll. Wer wird den Fragebogen ausfüllen? Um ein repräsentatives Ergebnis zu erhalten, müssen nicht alle Kunden und Kundinnen den Fragebogen ausfüllen. Es reicht eine repräsentative Gruppe, welche die Software verwendet und ihre Zufriedenheit mit der Software kundtut (Kirchhoff et al. 2010).

Ein wichtiger Faktor ist, dass die befragten Personen nicht zwingend das Vorgehensmodell, mit welchem das Produkt der Kunden und Kundinnen entwickelt wurde, kennen müssen. Es reicht, wenn der Fragesteller dies weiß.

Ein gutes Umfragedesign ist die Basis für die Auswertung der Fragen. Ein gut ausgewähltes Design, stellt sicher, dass die benötigten Themen effizient ausgewählt werden. Solch ein Forschungsdesign umfasst in der Regel die folgenden Komponenten bzw. Aufgaben (Malhotra 2010):

- Die benötigten Informationen definieren
- Den Stichprobenumfang festlegen
- Bestimmung der Mess- und Evaluationskriterien
- Einen Plan für die Datenanalyse entwerfen

3.3 Evaluationskriterien

3.3.1 Begriffsdefinition Evaluationskriterien

Ziel der Umfrage ist es, eine wissenschaftlich fundierte Bewertung des Evaluationsgegenstandes zu erlangen. Um dies zu bewerkstelligen, müssen klare und messbare Bewertungskriterien festgelegt werden. Erst danach können die Daten erhoben bzw. gesammelt und interpretiert werden. Die Auswahl dieser Kriterien bestimmt in der Regel das Evaluationsergebnis (Döring und Bortz 2016).

3.3.2 Kundenzufriedenheit messen mit Key-Performance-Indicator

Der Begriff Key-Performance-Indicator (KPI) bzw. Leistungskennzahl ist ein Wert, der zur Überwachung und Messung der Effektivität verwendet wird. Die meisten Branchen haben ihre eigenen Kennzahlen, die für ihren Bereich jeweils relevant sind, damit innerhalb der Branche diese Effektivität gemessen werden kann (John Reh 2020).

Für Kundenzufriedenheit allgemein, aber generell auch in der Softwarebranche oft genutzt, wird unter anderem oft der Customer-Satisfaction-Score (CSAT) verwendet. Dieser ist eine einfache Methode, um die Kundenzufriedenheit zu quantifizieren und in

Zahlen auszudrücken. In Umfragen haben Kund:innen die Möglichkeit konkrete Leistungen oder Erfahrungen eines Produktes oder Unternehmens anhand einer vordefinierten Skala zu bewerten. Das Ergebnis ist eine vergleichbare Kennzahl (KPI) die den Grad der Kundenzufriedenheit in Prozent ausdrückt (Natrapei 2019).

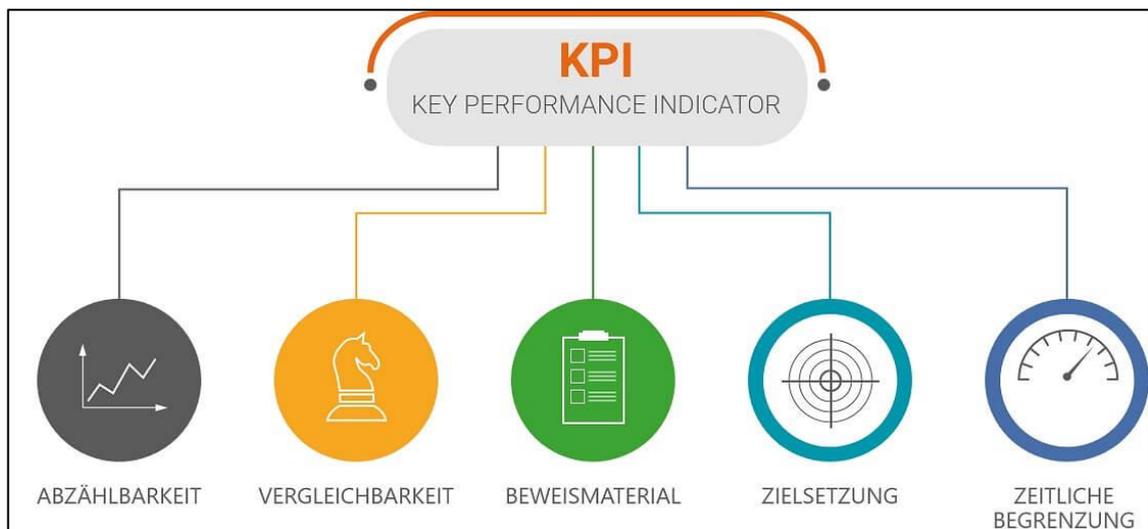
Die Bewertung erfolgt auf einer einzigen Frage, die messen soll, inwiefern das Produkt die Erwartungen der Kunden und Kundinnen erfüllen. Die Antwortmöglichkeiten sind in der Regel in Abstufungen von „Sehr unzufrieden“ bis „Sehr zufrieden“ zusammengefasst, und können mittels einer Single-Choice-Antwortmöglichkeit ausgewählt werden. Die meist verbreitetste CSAT-Berechnung zielt darauf ab, einen Prozentsatz zu kalkulieren. Je höher der Prozentwert ausfällt, umso zufriedener sind die Kunden und Kundinnen mit ihrem Produkt (Pecoraro 2021).

Der Wert wird anhand von Bewertungen anderer Produkte und Branchen in Relation gesetzt. Aufgrund des verwendeten Vorgehensmodells werden die CSATs daraufhin verglichen.

Zusätzlich zu der allgemeinen Kundenzufriedenheit werden noch andere Faktoren in der Fragestellung eine Rolle spielen. Wie wird die Usability wahrgenommen? Wie ist die Qualität der Software? Deckt die Software die notwendigen Anforderungen an benötigte Userinterfaces ab?

Ebenfalls relevant ist, in welcher Branche die Software eingesetzt wird (siehe Abbildung 10), um die KPIs auch miteinander vergleichbar zu machen.

Abbildung 10 - Key Performance Indicator



Quelle: Key Performance Indicator, 2020

4. Diskussion

Vorgehensmodelle und Software sind seit jeher eng miteinander verbunden. Mit der wachsenden Komplexität von Software und den daraus resultierenden Projekten entstanden mit der Zeit immer mehr Vorgehensmodelle und Ende der 90er Jahre auch komplett neue Denkmuster, die agilen Vorgehensmodelle.

4.1 Bewertung Lineare Vorgehensmodelle

Beide Ansätze haben noch immer ihre Daseinsberechtigung. Lineare Projekte haben viele Vorteile, wenn der Projektumfang und die Anforderungen ganz genau bekannt sind. Eine gute Dokumentation des Entwicklungsprozesses, der Kosten und des Aufwands lassen sich bereits von Beginn an abschätzen, und die Struktur lässt sich durch abgegrenzte Projektphase klar planen, um hier einige Vorteile zu nennen.

In diesem Fall können und werden auch die Software-Requirements ganz genau spezifiziert und geplant. Dies ist zum Beispiel bei sehr großen Projekten in der Luftfahrt oder auch Automobilbranche der Fall. Das Projektvolumen und die Anzahl der Stakeholder und Zulieferer sind zu groß, als dass man hier spontane Änderungen an Hard- oder auch der darauf laufenden Software machen kann. Jene Personen, welche die Anforderungen dafür festlegen, sind jedoch erfahrene Mitarbeiter:innen oder Ingenieure und Ingenieurinnen, die genau wissen, welche Anforderungen an das Projekt gegeben sind. Ist dies der Fall, so sind agile Vorgehensmethoden bei der Planung und Durchführung wohl nicht wirklich geeignet, sodass dies nur Kommunikationsoverhead bedeuten würde.

4.2 Bewertung agile Vorgehensmodelle

Im Gegensatz dazu sind jedoch die meisten Softwareprojekte keine Projekte, wo die Anforderungen bis ins kleinste Detail bereits bekannt und vorgegeben sind. Ein Kunde oder eine Kundin, welche eine Software beauftragen, haben zwar eine grobe Vorstellung an die Anforderungen, jedoch wirft die genaue Umsetzung derer oft Fragen auf. Aus Kundensicht ist dies auch nicht notwendig, da die Umsetzung durch professionelle Softwareentwickler:innen und Manager:innen erfolgt.

Das Problem liegt wie so oft im Detail. Aus meiner eigenen Erfahrung in der IT-Branche kann ich sagen, dass das größte Problem nicht das „Was“, sondern das „Wie“ ist. Auch wenn die Funktionalität im Programm grundsätzlich gegeben ist, so hat sich der Auftraggeber oder die Auftraggeberin die Funktionalität oder ein Feature meistens anders vorgestellt, als es im Endergebnis der Fall ist. Oft wissen die Kunden und Kundinnen zu Beginn nicht, was sie benötigen. Diese Erkenntnis bekommen sie erst, wenn sie das „fertige“ Produkt bedienen können. Dies ist ein großes Problem, da man eigentlich den Kunden bzw. die Kundin zufrieden stellen und nicht nur die vertraglichen

Vereinbarungen erfüllen möchte. Mit dem agilen Ansatz können hier durchaus Vorteile für den Kunde bzw. der Kundin entstehen. Diese Vorteile gehen jedoch auch mit erhöhter Verfügbarkeit und Aufwand auf Kundenseite einher. So sollten Kunden und Kundinnen für wiederholend stattfindende Abstimmungsmeetings zur Verfügung stehen. Auf Kundenseite darf hier jedoch nicht nur der Aufwand im Fokus sein, sondern sollten auch die Vorteile, die mit diesem Ansatz einher gehen, erkannt werden.

Dies wird von Kunde zu Kunde unterschiedlich gehandhabt. So war ich bereits in Projekten beteiligt, die zur Gänze auf agile Vorgehensmethoden gesetzt haben. Die Features für die Software wurden im Zwei-Wochen-Rhythmus geplant und umgesetzt. Kunden und Kundinnen haben alle zwei Wochen eine auslieferbare Software, welche diese auch nach einer kurzen Testphase in die Produktivumgebung eingliedern können. Es gab im Vorfeld keine Verträge über die finalen Funktionalitäten, sondern lediglich jene, die in einem Zwei-Wochen-Rhythmus geplant wurden. In einem sogenannten Backlog (Aufgaben bzw. Features die zu erledigen sind) wurden zwar weitere Anforderungen verwaltet, jedoch wurde immer nur für die nächsten beiden Wochen geplant.

Ein solches Vorgehen ist in der IT Industrie sicher die Ausnahme. Die meisten Kunden und Kundinnen wollen Planungssicherheit für ihre Ausgaben haben, und dadurch auch ein zugesichertes, vorab definiertes Feature-Set für ihr Geld bekommen.

4.3 Bewertung Hybride Vorgehensmodelle

Hybride Vorgehensmodelle (ob gewollt oder nicht) haben sich mit der Zeit in Softwareprojekten etabliert. So wird vorab eine gründliche Planung mit Pflichtenheft erstellt. Im Nachgang wird daraus ein Angebot abgeleitet und ein Termin mit Features dem Kunden bzw. der Kundin zugesagt. Die Entwicklung erfolgt jedoch in den meisten Fällen nicht nach der Wasserfallmethodik, sondern agil. Wie bereits in 2.2.1 erwähnt, basieren die aktuellen Standards in der Softwareentwicklung auf agilen Ansätzen. Will man eine bestimmte Softwarequalität gewährleisten, so kommt man um gewisse agile Grundprinzipien wie schnelles Feedback oder CI gar nicht herum. Änderungen der Anforderungen können in den meisten Fällen besprochen werden, und durch andere Features ausgetauscht werden. Die Detailplanung der Software erfolgt dann ähnlich dem vollagilen Konzept. Die meisten Softwareunternehmen sind sich, aus meiner Erfahrung, dessen gar nicht bewusst, da man oft auch mit „Wir arbeiten agil“ aktiv wirbt. Dies muss aber nicht schlecht sein, so kann man die Vorteile eines linearen Vorgehensmodells wie Planungssicherheit, Kostenkontrolle für den Auftraggeber bzw. der Auftraggeberin mit den Vorteilen von agilen verbinden.

Es bleibt, wie so oft, situationsabhängig, wann ein Vorgehensmodell sich als die richtige Entscheidung erweist. Wichtig ist jedoch, dass man nicht starr auf ein Modell vertraut, sondern für sich selbst die richtigen Methodiken und Vorteile rauspickt. Dies kann auch

innerhalb des Unternehmens von Projekt zu Projekt unterschiedlich sein, jedoch würde ich empfehlen, nicht alle Erfahrung nach jedem Projekt über Bord zu werfen.

4.4 Ausblick

Viele Softwareprojekte werden aktuell mit hybriden Vorgehensmodellen entwickelt. Ob sich dies positiv auf die Kundenzufriedenheit auswirkt, ist bis dato oft nicht Teil der Studien und damit auch nicht final geklärt. Viel mehr werden in Studien Vorgehensmodelle nur so weit in Betracht gezogen, bis ein Projekt abgeschlossen wurde, oder nicht, und ob diese einen Einfluss auf den Projekterfolg hatte. Die meisten Studien zielen also auf die interne Qualität ab.

Kundenzufriedenheit ist aus meiner Sicht ein externer Qualitätsstandard. Kunden und Kundinnen haben das Interesse, dass die Software ihren Erwartungen entspricht. Wie das Produkt entwickelt wurde, ist für Auftraggeber:innen von sekundärem Interesse.

Mit neuen empirischen Daten soll nun die Kundenzufriedenheit von Softwareprojekten gemessen werden. Ebenso werden Studien, die sich vor allem mit hybriden Vorgehensmodellen beschäftigen, untersucht. Dies wird im praktischen Teil der Bachelorarbeit (Bachelorarbeit II) evaluiert.

5. Methodik der empirischen Untersuchung

Um die Nachvollziehbarkeit der empirischen Untersuchung zu gewährleisten, wird in diesem Abschnitt der Arbeit die angewandte wissenschaftliche empirische Methode erläutert. Um dies zu bewerkstelligen, werden nachfolgend das Untersuchungsdesign sowie der Aufbau des Fragebogens beschrieben.

5.1 Untersuchungsdesign

Das Untersuchungsdesign bzw. Forschungsdesign dient als Vorlage für die Durchführung eines Marktforschungsprojekts. Es legt die Einzelheiten fest, wie eine Umfrage durchgeführt werden soll. Als Ausgangspunkt für eine Untersuchung muss eine Problemstellung definiert werden. Im Anschluss werden Forschungsfragen und Hypothesen aufgestellt, die von der Umfrage beantwortet werden (Malhotra 2010).

Jene angesprochenen Problemdefinitionen, die bereits in Kapitel 1 erwähnt wurden, dienen als Grundlage für das Forschungsdesign.

Die verwendete Art der Untersuchung ist Primärforschung. Als Primärforschung werden Daten, die für ein anstehendes Problem bzw. Untersuchungsproblem neu erhoben werden müssen, bezeichnet. Dies wird im sozialwissenschaftlichen Bereich oft auch als Feldforschung („Field Research“) bezeichnet (Kuß 2007).

Primärdaten können in qualitativer, sowie in quantitativer Form erhoben werden. Qualitative Marktforschung ist eine unstrukturierte Forschungsmethodik, die auf einer kleinen Stichprobe basiert. Quantitative Marktforschung ist eine Methode, die darauf abzielt, die gesammelten Daten zu quantifizieren und wendet in der Regel eine Form der statistischen Analyse an (Malhotra 2010).

Für die Problemstellung und Hypothese wurde in dieser Arbeit auf eine quantitative Forschungsmethode gesetzt. Das Ziel war, mit einer Umfrage möglichst viele Personen zu erreichen, die Kenntnisse über das eingesetzte Vorgehensmodell bei der Bedienung einer von ihnen eingesetzten Software haben. Ebenso sollte die befragte Person Kenntnisse darüber haben, ob zum Zeitpunkt des Projektstarts der Umfang und das Budget des Projektes bereits bekannt waren. Mit diesem Vorgehen soll die Hypothese beantwortet werden.

5.2 Stichprobe

Eine Schwierigkeit von Umfragen liegt in deren Repräsentativität. Inwieweit sind die befragten Personen für die Grundgesamtheit auf Basis der Stichprobe repräsentativ (Kuß 2007)?

Hierbei geht es um die Auswahl von Untersuchungseinheiten (z.B. Personen) bei denen eine Befragung durchgeführt wird. Im Regelfall werden kleine Teilgruppen von einer

Gesamtmenge befragt. Anstelle einer Totalerhebung von zum Teil Millionen Proband:innen wird eine Stichprobe derer verwendet (Kuß 2007).

Grundsätzlich ist bei einer Stichprobe darauf zu achten, dass diese ein möglichst gutes Abbild der Grundgesamtheit ist. Dies wird insbesondere damit erreicht, indem man auf zufällige Stichproben setzt (Kirchhoff et al. 2010).

Grundsätzlich kann man jedoch zwischen zufallsgesteuerten Stichproben und nicht zufallsgesteuerten Stichproben unterscheiden. Bei zufallsgesteuerten Stichproben haben Proband:innen aus einer Grundgesamtheit dieselbe Wahrscheinlichkeit, Teilnehmer:in einer Umfrage zu werden. Bei nicht zufallsgesteuerten Stichproben werden die Teilnehmer:innen auf Basis persönlicher Erfahrung oder Wissen der forschenden Person ausgewählt (Malhotra 2010).

Bei der Nichtwahrscheinlichkeitsstichprobe gibt es mehrere Subformen. Beim Konzentrationsverfahren („judgemental sampling“) werden Elemente der Grundgesamtheit nach dem Urteil der forschenden Person absichtlich ausgewählt. Die forschende Person wählt nach eigenem Ermessen Elemente aus, die in die Stichprobe aufgenommen werden, weil die Person glaubt, dass diese repräsentativ seien. Eine weitere Spezialform dessen ist die Quotenstichprobe („quota sampling“). Diese kann als zweistufige Stichprobe gesehen werden. Die erste Stufe besteht aus einer Kontrollkategorie aus der Grundgesamtheit, die zweite Stufe aus den Stichproben eines Konzentrationsverfahrens. So kann sichergestellt werden, dass die Zusammensetzung der Stichprobe mit der Zusammensetzung der Grundgesamtheit übereinstimmt (Malhotra 2010).

Im Fall von meiner Problemstellung war für die Beantwortung der Fragen IT-Wissen gefragt. Die Grundgesamtheit aller Personen in Österreich ist jedoch nicht in der IT tätig (Verband Österreichischer Software Industrie 2021).

Für die Auswahl der Proband:innen wurde eine Quotenstichprobe angewandt. Als erste Stufe wurde die Umfrage an Personen, unabhängig von ihrem IT-Fachwissen, auf mehreren Kanälen geschickt. Für die zweite Stufe war die Auswahl der Proband:innen nicht rein zufällig ausgewählt, sondern die Umfrage wurde zusätzlich an von mir spezifisch ausgewählte Kanäle und Expert:innen im IT-Bereich geschickt. Diese sollten Kenntnis von einer Software und dem eingesetzten Vorgehensmodell haben, sowie über dessen Entwicklung Bescheid wissen. Die Überlegung war, dass die Grundgesamtheit aller Personen die Fragen nicht oder nur in sehr begrenztem Maße beantworten können, jedoch Expert:innen im IT-Bereich sehr wohl. Die Anzahl der Teilnehmer:innen, welche für die Hypothese wichtige Fragen beantworten kann, ist generell sehr eingeschränkt, da diese auf arbeitende Personen im IT-Sektor basiert.

5.3 Fragebogendesign

Für die Formulierung von Fragen ist es wichtig, welche Meinungen, Sachverhalte etc. grundsätzlich ermittelt werden sollen. Diese sollten durch Problemdefinitionen und Untersuchungshypothesen bereits bestimmt sein. Wenn man beispielsweise die Hypothese überprüfen will, ob eine mangelnde Kundenzufriedenheit zu einer geringeren Kundenbindung führt, so muss man erheben, wie gut die Zufriedenheit ist (Kuß 2007).

Als Grundlage für die gestellten Fragen im Fragebogen, diente die Problemstellung und Hypothese dieser Arbeit. Für den Fragebogen wurden ausschließlich geschlossene Fragen verwendet.

Geschlossene Fragen bieten eine einfache Beantwortung und wenig Probleme bei der Verarbeitung bzw. Auswertung der Antworten. Ebenso werden Antworten gewählt, auf welche die Auskunftsperson ohne die Vorgaben nicht gekommen wäre (Kuß 2007).

Der Fragebogen beinhaltete zehn Fragen und war in drei Hauptbereiche aufgeteilt. Nach einem kurzen Einleitungstext über die Umfrage hatte die befragte Person eine Übersicht über die einzelnen Frageblöcke. Diese gliederten sich in persönliche Angaben, Vorgehensmodelle und Softwarelösungen.

5.3.1 Persönliche Angaben

Die ersten Fragen im Bogen widmeten sich den persönlichen Angaben der Person. Ziel der persönlichen Angaben war es nicht, eine Fülle von demographischen Eigenschaften herauszufinden, da dies für die Beantwortung der Hypothese auch nicht notwendig ist. Lediglich die Berufsbranche der befragten Person ist relevant, da dies wesentliche Informationen zur Stichprobe preisgibt. Die Frage war mittels Multiple Choice auf die Branchen der WKO eingegrenzt (WKO 2022). Zusätzlich wurde die Auswahl um eine sonstige Branche sowie der Auswahlmöglichkeit nicht beruflich tätig zu sein ergänzt, um keine Personen auszuschließen.

Des Weiteren wurde gefragt, ob die befragte Person beruflich mit einem Softwareprogramm arbeitet. Hierfür wurde ebenso auf Multiple Choice Antworten gesetzt. Die Auswahlmöglichkeiten waren dichotom, und wurden um eine Möglichkeit, dies nicht zu wissen, ergänzt. Wurde die Frage positiv beantwortet, so ergab sich die Zusatzfrage, ob die Software speziell für das Unternehmen entwickelt worden ist. Also, ob es sich beim eingesetzten Programm um eine Individualsoftware handelt. Die Auswahlmöglichkeiten waren dieselben wie in der Frage zuvor.

Unabhängig der Antworten zuvor, wurde die nächste Frage immer gestellt. Diese beinhaltete dieselbe Fragestellung wie Frage zwei, nur mit dem Ziel, ob privat eine Softwarelösung verwendet wird.

Mit diesen Informationen soll sichergestellt werden, welche Personen die Umfrage ausfüllen und welchen Bezug die befragten Personen zu Softwareprogrammen haben.

5.3.2 Vorgehensmodelle

Die Fragen im zweiten Bereich dienten zum Ableiten des Kenntnisstandes hinsichtlich Vorgehensmodelle der befragten Personen. Die Eingangsfrage in diesem Bereich zielte darauf ab, welche generellen Vorgehensmodelle der befragten Person bekannt sind. Die Optionen beinhalteten Lineare Vorgehensmodelle, Agile Vorgehensmodelle, und Hybride Vorgehensmodelle und waren mittels Multiple Choice Mehrfachauswahl möglich. Ebenso war eine Auswahl vorhanden, wenn keines der Vorgehensmodelle bekannt war.

Die nächste Frage beinhaltete den Detailgrad der Kenntnisse über die gängigsten Vorgehensmodelle (Shiklo 2019). Zu diesem Zweck wurde eine Matrix-Skala verwendet, um die verschiedenen Bewertungsobjekte zuordnen zu können. Die Objekte auf der Vertikalachse waren aktuell eingesetzte Vorgehensmodelle, in der Horizontalachse konnte der Kenntnisstand über dieses Vorgehensmodell ausgewählt werden. Die Auswahloptionen dienten anzugeben, ob ausgeprägtes Wissen und Erfahrung, Basiswissen vorhanden war, oder ob diese noch unbekannt sind. Die Antwort war verpflichtend. Zusätzlich gab es in der Vorgehensmodellachse noch die Option, ob noch weitere bzw. noch andere Vorgehensmodelle bekannt gewesen wären. Dies war jedoch nur als Option verfügbar und beinhaltete kein Freitextfeld.

5.3.3 Softwarelösungen

Im finalen Teilbereich der Umfrage ging es nun um die Verknüpfung zwischen benutzer Software und eingesetztem Vorgehensmodell. Dies ist für die Hypothese von entscheidender Bedeutung. Eine Bestätigung oder Widerlegung der Hypothese ist nur möglich, sofern befragte Personen Kenntnis von Software, eingesetztem Vorgehensmodell zur Entwicklung derer, sowie Kenntnis über das Entwicklungsprojekt haben. Nur wenn diese Kriterien erfüllt sind, hat die Kundenzufriedenheit für die Hypothese eine Relevanz.

Zunächst wurde gefragt, ob die befragte Person eine Software in Verwendung hat, von derer diese das Vorgehensmodell kennt, mit welchem das Programm entwickelt wurde. Die Frage wurde als Entscheidungsfrage gestellt. Im Falle von einer positiven Antwort, wurde auf die nächste Frage verwiesen. Wurde die Frage verneint, so wurde die Umfrage beendet und auf die Abschlussfolie verlinkt, da die nächsten Fragen für die Auswertung der Daten, die für die Hypothese relevant sind, leider nicht von Bedeutung waren.

War das Vorgehensmodell bekannt, so gelangte man zur nächsten Frage. Diese sollte herausfinden, mit welchem Vorgehensmodell die von der befragten Person verwendete Software entwickelt worden ist. Zur Auswahl stand Agiles, Lineares sowie Hybrides Vorgehensmodell. Die Frage war eine Multiple Choice Frage mit einer einzigen Antwortmöglichkeit.

Darauf aufbauend wurde ebenso die Frage gestellt, ob zum Projektstart der Software bereits Umfang und Preis feststanden. Die Antwortmöglichkeiten waren wiederum Multiple-Choice mit einer Ja-Nein-Frage sowie einer weiteren Option dies nicht zu wissen. Diese Frage war von großer Bedeutung, da nur anhand derer zugeordnet werden kann, ob es sich um ein Fixpreisprojekt handelt oder nicht.

Zu guter Letzt wurde noch eine Frage über die Kundenzufriedenheit gestellt. Wie bereits in 3.3.2 erwähnt, soll für die Messung derer ein CSAT verwendet werden. Die Frage setzte sich aus dem Fragetext „wie zufrieden man mit dem gewählten Softwareprodukt ist“, sowie aus einer Optionsskala mit Antwortmöglichkeiten zusammen. Die Antwortmöglichkeiten gliederten sich von 1 – 10. Die Zahl 1 stand hierfür, dass man gar nicht mit der Software zufrieden sei. Die Zahl 10 stand dafür, mit dem gewählten Produkt extrem zufrieden zu sein. In Einserschritten waren Möglichkeiten der Auswahl zur Abstufung der beiden Antwortmöglichkeiten vorhanden. Es war nur eine Antwortmöglichkeit zugelassen. Der Wertebereich 1-3 wird als negativ, 4-6 neutral, und alles über 7 als positives Ergebnis gewertet.

Basierend auf dieser Methode kann eine Customer Satisfaction Score (CATS) für die Kundenzufriedenheit gemessen werden (Natrapei 2019).

Ursprünglich waren weitere Fragen zur Software in Bezug auf deren Usability bzw. Softwarequalität geplant, jedoch wurden diese aufgrund der zusätzlich subjektiven Wahrnehmung zur generellen Kundenzufriedenheit wieder entfernt. Der CATS sollte als einziger Wert zum Vergleich dienen.

5.3.4 Abschluss

Der Fragebogen schloss für alle befragten Personen, unabhängig ihrer Auswahl und damit verbundenen Antworten, mit einer Dankesfolie ab.

6. Ergebnisse

6.1 Ergebnisse der Literaturstudie

Wie bereits im Kapitel 3.1. erläutert, hat das Vorgehensmodell bei der Entwicklung von Software einen Einfluss, ob ein Softwareprojekt erfolgreich abgeschlossen werden kann oder nicht. Analysen, ob das Vorgehensmodell nach dem Abschluss eines Projektes auch Einfluss auf die Kundenzufriedenheit hat, gibt es wenige. Die Studien zum Thema Kundenzufriedenheit und Agile Vorgehensmethoden sind daher sehr limitiert.

Eine Sammelstudie (Dybå und Dingsøy 2008) weist darauf hin, dass Kunden:innen bei großen und komplexen Projekten von Agilen Methoden profitieren. Dies wirke sich einerseits auf die Arbeitszufriedenheit und die Produktivität des Auftragnehmers aus, sowie auch auf die Kundenzufriedenheit des Auftraggebers. Über die Kundenzufriedenheit der Anwender wird jedoch nichts erwähnt.

In einer anderen Studie (Cartaxo et al. 2013) wurden verschiedene Aspekte wie Kundenzufriedenheit, Termintreue, Qualität, Agilität usw. berücksichtigt. Die Studie beruhte auf Umfragen, welche nach einer Querschnittserhebung mit einer realen Projektstichprobe durchgeführt wurde. Die Stichprobe wurde in zwei Gruppen aufgeteilt. Eine Gruppe wurde zu Scrum-Projekten befragt, die andere zu traditionellen Projekten. Die Kundenzufriedenheit zielt hier nicht unbedingt nur auf die Bedienung der Software ab, sondern mehr, wie gut die Zusammenarbeit als Auftraggeber mit einem Auftragnehmer von einem externen Standpunkt her gewesen ist. Das Ergebnis der Umfrage war, dass es keine Hinweise darauf gibt, dass die Scrum-Gruppe eine bessere Kundenzufriedenheit erzielte als die Nicht-Scrum-Gruppe. Die Forschenden konnten durch die Einführung von Scrum nicht ableiten, dass die Kundenzufriedenheit gestiegen wäre.

6.2 Ergebnisse der Befragung

In diesem Kapitel finden sich die Antworten auf den in Kapitel 5.3 beschriebenen Fragebogen. Bei der Umfrage gaben insgesamt 87 befragte Personen eine Antwort ab. 94 Personen starteten die Umfrage, sodass sich eine Abschlussquote von 92,6 % ergibt.

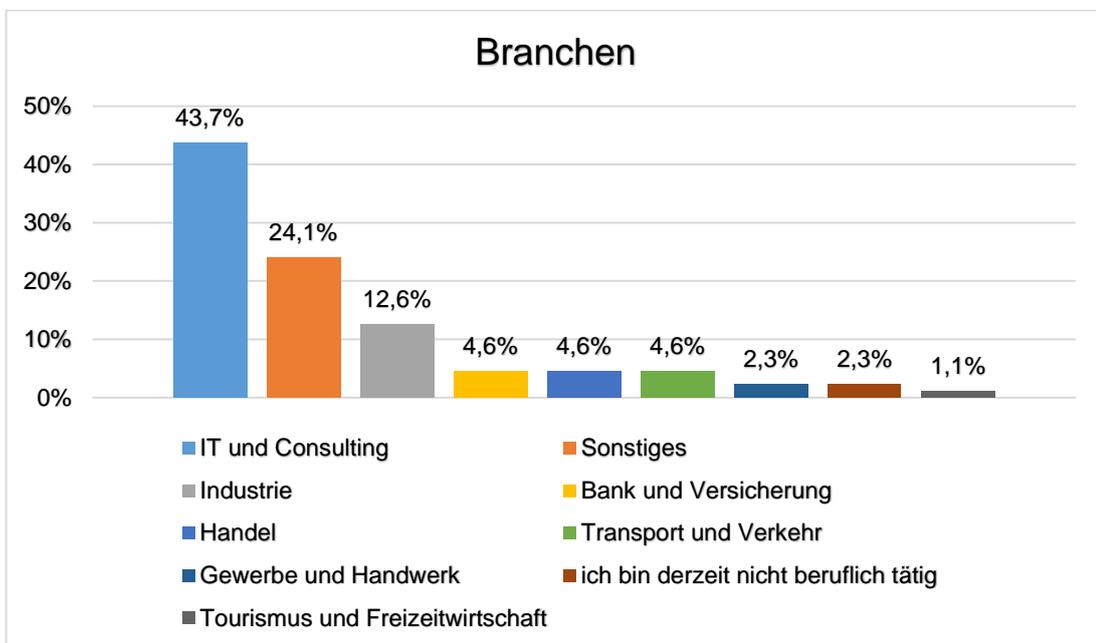
6.2.1 Persönliche Angaben

Um die Daten in Relation zu den gegebenen Antworten zu stellen, wurde zunächst erhoben, welche demographischen Merkmale und Erfahrung im Softwarebereich bereits vorhanden sind. Mithilfe dieser, konnte die Quotenstichprobe verifiziert werden.

Frage 1: In welcher Branche sind Sie beruflich tätig?

Alle an der Umfrage erfolgreich teilgenommenen Personen haben diese Frage beantwortet. Die meisten befragten Personen (siehe Abbildung 11) arbeiten in der Branche IT und Consulting. Dies entspricht 43,7 % aller Proband:innen. Jene Branche, die mit 24,6 % aller befragten Personen die zweitgrößte Gruppe darstellt, ist Sonstiges, gefolgt von Industrie mit 12,6 %. Die Branchen Bank und Versicherung, Handel sowie Transport und Verkehr konnten jeweils ein Ergebnis von 4,6 % aller befragten Personen erzielen. Gewerbe und Handwerk konnte 2,3 % verbuchen, Tourismus und Freizeitwirtschaft 1,1 %. 2,3 % aller Befragten gaben an, derzeit nicht beruflich tätig zu sein.

Abbildung 11 - Branchen



Frage 2: Arbeiten Sie beruflich mit Softwareprogrammen bzw. Softwarelösungen?

Die große Mehrheit (siehe Abbildung 12) der beruflich aktiven Personen gab an, derzeit mit einem Softwareprogramm auch beruflich zu arbeiten. Dies entspricht 87,1 % aller Befragten. Von der Gesamtmenge der abgegebenen Antworten entsprachen 10,6 % jener Option, dass sie derzeit nicht beruflich mit einer Software arbeiten. Der Rest von 2,4 % aller Proband:innen weiß nicht, ob dies gegenwärtig der Fall ist.

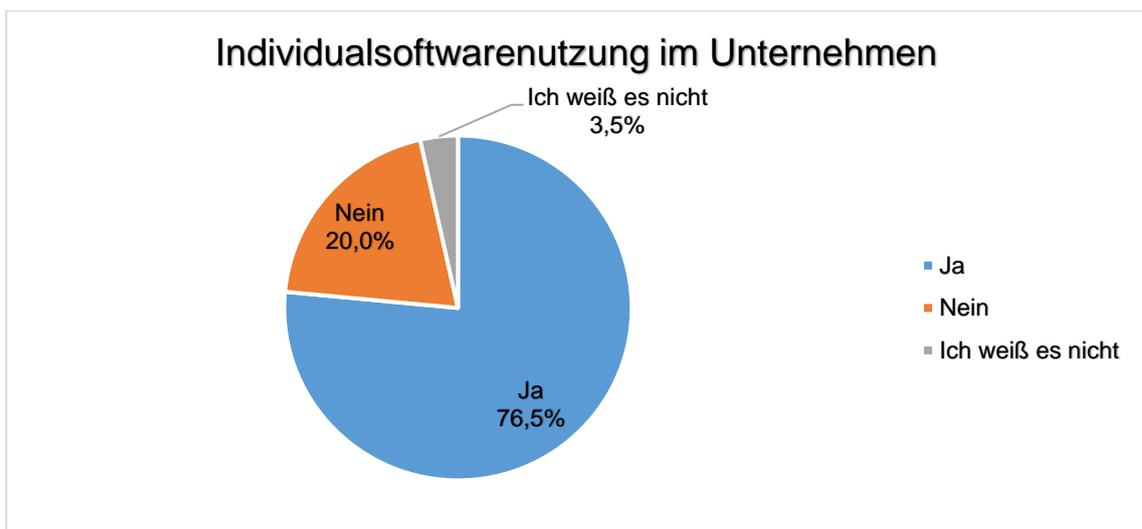
Abbildung 12 - Softwarenutzung beruflich



Frage 3: Arbeiten Sie in einem Unternehmen, das Software bzw. Programme einsetzt, die speziell für Ihr Unternehmen entwickelt worden sind?

Von allen befragten Personen gaben 76,5 % an, dass sie beruflich mit einer Software arbeiten, welche speziell für ihr Unternehmen entwickelt worden ist (siehe Abbildung 13). Damit arbeiten mehr als Dreiviertel aller Teilnehmer:innen, welche beruflich Software einsetzen, mit Individualsoftware. 20 % der befragten Teilnehmer:innen gaben an, dass sie mit keinem speziell für ihr Unternehmen entwickelten Programm arbeiten. 3,5 % aller Personen wissen dies nicht.

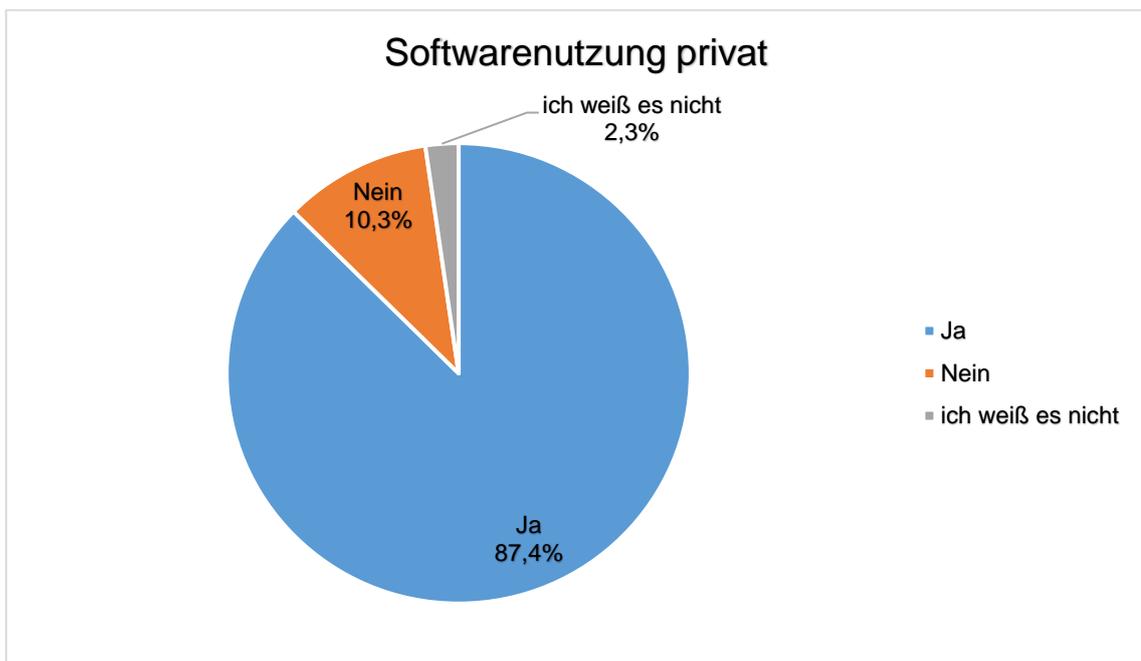
Abbildung 13 - Individualsoftwarenutzung im Unternehmen



Frage 4: Arbeiten Sie privat mit Softwareprogrammen bzw. Softwarelösungen?

Ebenso wie bei Frage 2 gab die große Mehrheit an, auch privat mit Software zu arbeiten (siehe Abbildung 14). Dies entspricht einem Prozentsatz von 87,4 % aller beteiligten Personen der Umfrage. 10,3 % aller befragten Teilnehmer:innen gaben an, dies nicht zu tun. Wie bei den Fragen zuvor gaben auch einige Proband:innen an, dies nicht zu wissen. Dieser Anteil beträgt 2,4 %.

Abbildung 14 - Softwarenutzung privat



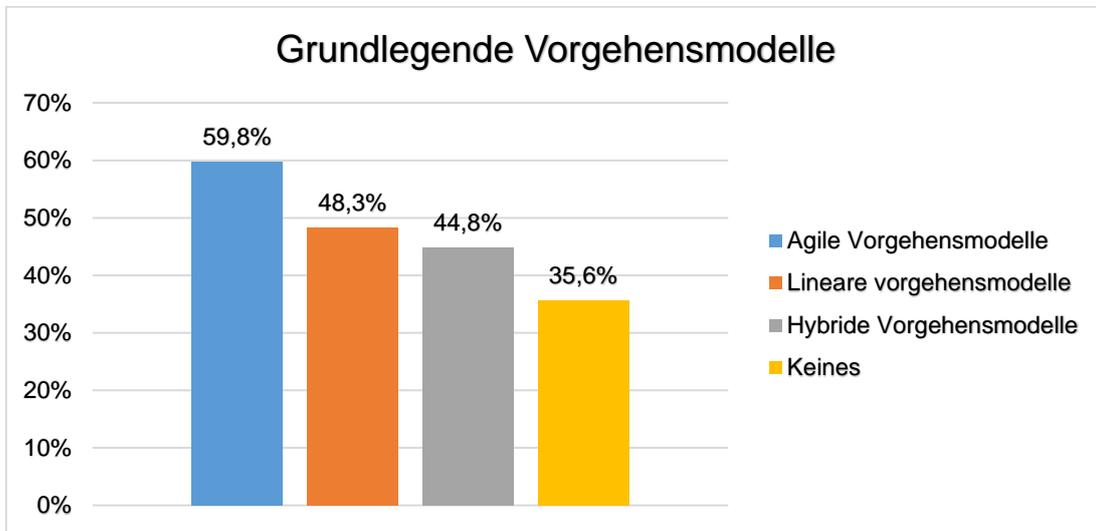
6.2.2 Vorgehensmodelle

Die folgenden Fragen brachten einen Überblick über den generellen Wissenstand der befragten Personen hinsichtlich Vorgehensmodelle bei Softwareentwicklung.

Frage 5: Welche grundlegenden Vorgehensmodelle zur Softwareentwicklung sind Ihnen bekannt?

Vorgehensmodelle in der Softwareindustrie sind den meisten Proband:innen bekannt (siehe Abbildung 15). Den größten Anteil machen dabei Agile Vorgehensmodelle aus. Demnach ist 59,8 % aller befragten Proband:innen diese Art der Softwareentwicklung bekannt. Lineare Vorgehensmodelle sind für 48,3 % der Teilnehmenden ein Begriff. Hybride Vorgehensmodelle sind einem Anteil von 44,8 % der befragten Kund:innen ebenso nicht unbekannt. 35,6 % aller an der Umfrage teilnehmenden Personen war kein Vorgehensmodell bekannt.

Abbildung 15 - Kenntnis über grundlegende Vorgehensmodelle



Frage 6: Wie schätzen Sie Ihren persönlichen Wissensstand zu den folgenden Vorgehensmodellen ein?

Die Ergebnisse der Umfrage legen dar, dass das Wasserfallmodell mit 34,5 % jenes Vorgehensmodell ist, bei denen die befragten Teilnehmer:innen das ausgeprägteste Wissen bzw. Erfahrung damit haben. 28,7 % der befragten Personen gaben an, Basiswissen über das Vorgehensmodell zu verfügen. Beträchtliche 36,8 % haben vom Wasserfallmodell noch nie etwas gehört.

Die Umfrage zeigt ebenso, dass 32,2 % aller an der Umfrage teilnehmenden Personen ausgeprägtes Wissen oder Erfahrung mit Scrum haben. Über ein Basiswissen über Scrum, verfügen demnach 34,5 % aller Beteiligten. Dies ist der höchste Wert in dieser Kategorie. Laut den Ergebnissen haben 33,3 % noch nie etwas von Scrum gehört, was den geringsten Wert darstellt.

Des Weiteren herrscht Erfahrung oder Wissen über Kanban bei 23 % aller Proband:innen. Zudem haben 34,5 % Basiswissen über dieses Vorgehensmodell. Dies ist der geteilte Spitzenwert mit Scrum in dieser Kategorie. Für 42,5 % aller beteiligten Personen ist dieses Vorgehensmodell unbekannt.

Das V-Modell, ein weiteres Lineares Vorgehensmodell, ist laut den Daten jenes Vorgehensmodell, mit denen zumindest 13,8 % aller befragten Proband:innen ausgeprägtes Wissen oder Erfahrung haben. Basiswissen in diesem Vorgehensmodell ist mit 31 % im Vergleich zur Erfahrung relativ hoch.

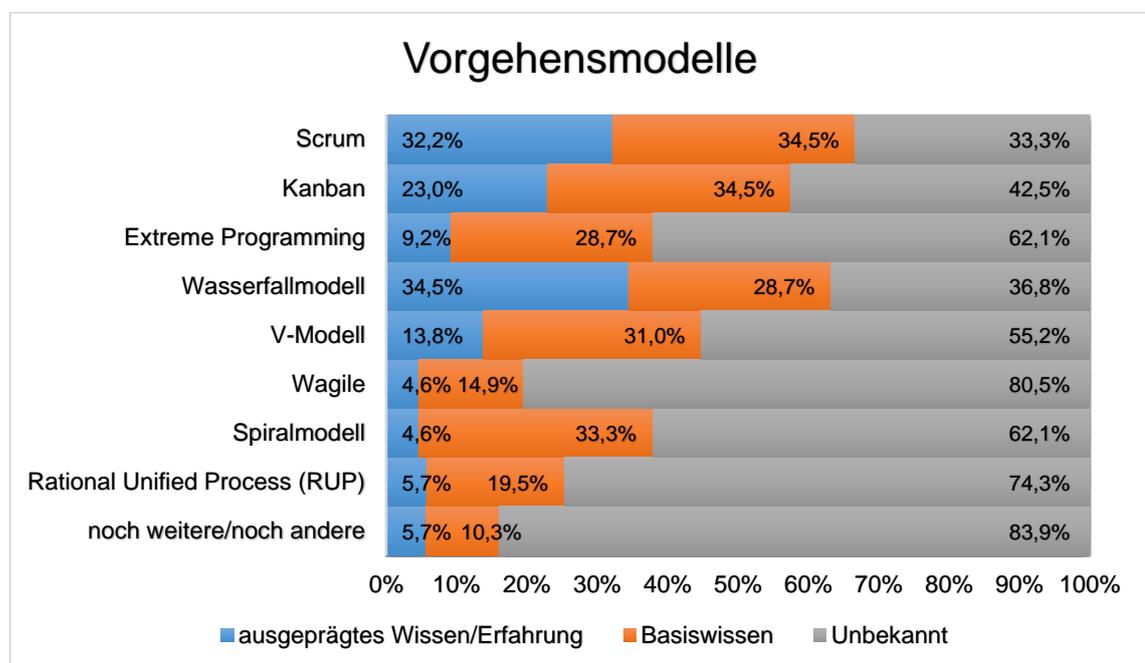
Erwähnenswert ist auch noch, dass zumindest 9,2 % ausgeprägtes Wissen oder Erfahrung mit Extreme Programming haben, jedoch ist dieses Vorgehensmodell für 62,1 % aller teilnehmenden Personen unbekannt.

Die restlichen Vorgehensmodelle haben im ausgeprägten Wissen einen geringen Anteil. Weder Wagile, Spiralmodell, Rational Unified Process (RUP) oder sonstige Vorgehensmodelle kommen hier auf einen Wert von über 6 %. Das Spiralmodell ist jedoch zumindest einem Drittel der befragten Teilnehmer:innen ein Begriff.

Wagile ist für die meisten Proband:innen unbekannt, sodass 80,5 % von diesem Vorgehensmodell bzw. dem verbinden verschiedener Vorgehensmodelle noch nie etwas gehört haben.

83,9 % haben kein Wissen über weitere Vorgehensmodelle. Damit wurden die gängigsten Vorgehensmodelle abgedeckt (siehe Abbildung 16).

Abbildung 16 - Vorgehensmodelle



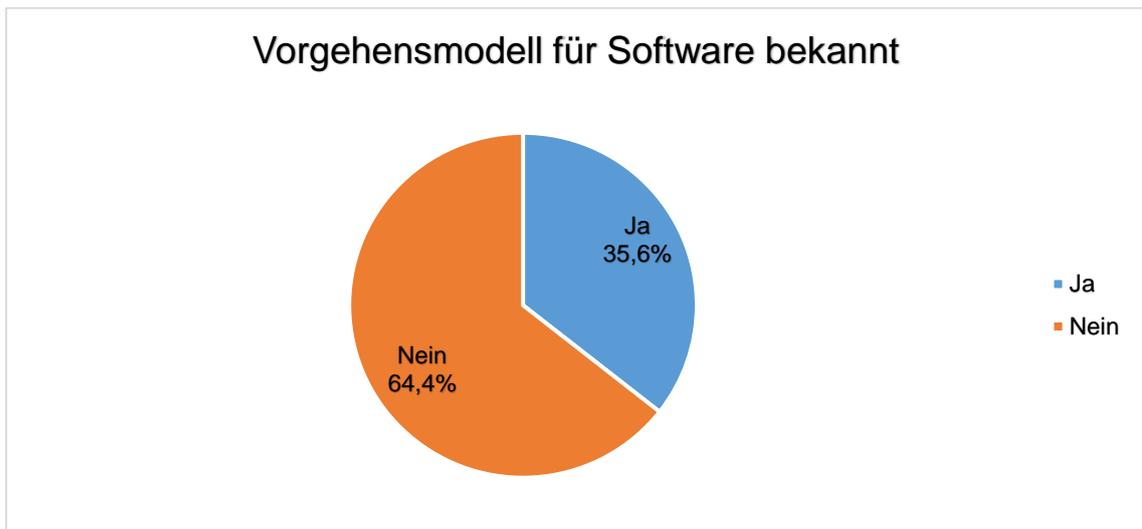
6.2.3 Softwarelösungen

Die Fragen in diesem Kapitel hatten das Ziel, die benützte Software aus den Fragen zuvor mit dem Vorgehensmodell, mit welchem die Software entwickelt ist, zu verknüpfen. Anschließend wurden noch Fragen über die Kundenzufriedenheit beantwortet.

Frage 7: Benutzen Sie eine Software bei der Ihnen das Vorgehensmodell, mit dem die Software entwickelt wurde, bekannt ist?

Die größte Teilmenge aller befragten Personen verneinte diese Frage. 64,4 % aller befragten Personen verwenden keine Software, von der ihnen das Vorgehensmodell bekannt ist. 35,6 % beantworteten diese mit Ja (siehe Abbildung 17).

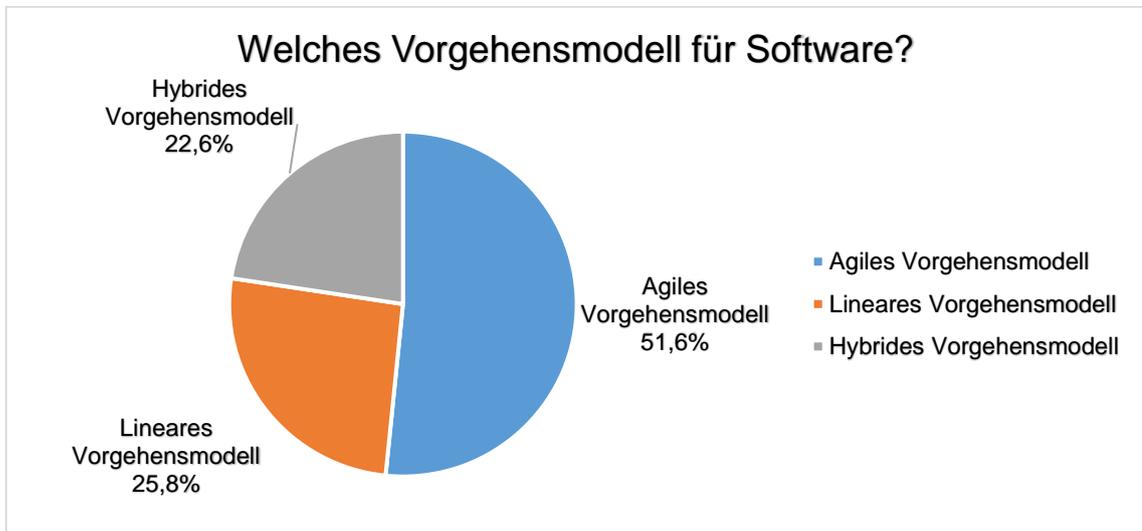
Abbildung 17 - Vorgehensmodell für Software bekannt



Frage 8: Mit welchem Vorgehensmodell wurde die Software entwickelt?

Die meisten Proband:innen gaben an (siehe Abbildung 18), dass die von ihnen bekannte Software mittels einem Agilen Vorgehensmodell entwickelt wurde. Das sind 51,6 % aller Befragten. Die nächstgrößte Gruppe innerhalb der Vorgehensmodelle bildet die Gruppe von Linearen Vorgehensmodellen. Diese liegen mit 25,8 % vor der Gruppe der Hybriden Vorgehensmodelle. Jene Hybride Vorgehensmodelle bilden mit 22,6 % aller Vorgehensmodelle das Schlusslicht der Frage.

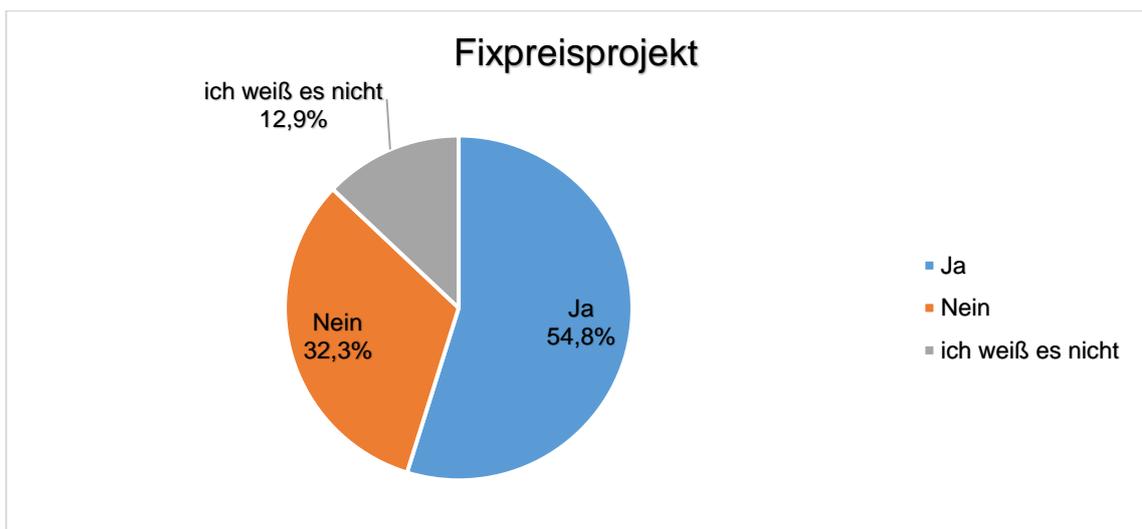
Abbildung 18 - Einsatz von Vorgehensmodellen



Frage 9: Waren beim Softwareprojekt zum Entwicklungsstart bereits Umfang und Preis bekannt? (Fixpreisprojekt)

Bei der Mehrheit der Befragten, nämlich 54,8 %, waren zum Entwicklungsstart der Software bereits Preis und Umfang vorgegeben. Diese Gruppe bildet die Grundlage für die Erkenntnis, ob es sich bei der Software um ein Fixpreisprojekt handelt. Die prozentuell gesehen nächstgrößere Antwortmöglichkeit war Nein mit 32,3 %. Zudem gaben 12,9 % aller Teilnehmenden der Befragung an, dies nicht zu wissen (siehe Abbildung 19).

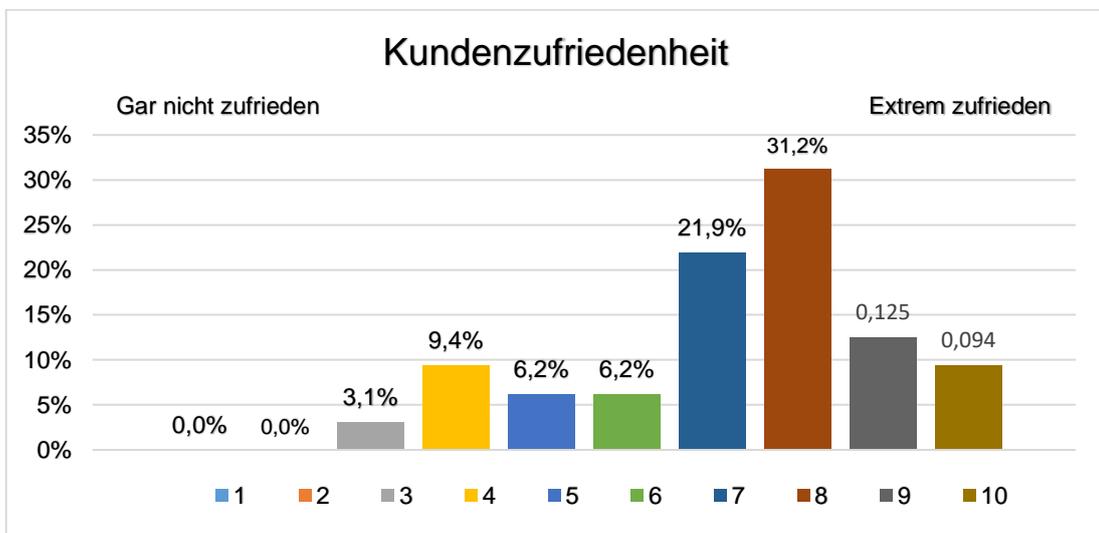
Abbildung 19 - Übersicht Fixpreisprojekt



Frage 10: Wie zufrieden sind Sie mit dem gewählten Softwareprodukt?

Die letzte Frage im Fragebogen gibt Aufschluss über die Kundenzufriedenheit der eingesetzten bzw. ausgewählten Software. Die Kund:innen gaben mehrheitlich an, mit der Software recht zufrieden zu sein. Dies spiegelt sich in einem Mehrheitsanteil bei der Zahl 8 wider (siehe Abbildung 20). Das sind 32,3 % aller Befragten. Knapp dahinter liegt der nächstgrößte Zahlenwert der Zufriedenheit mit 7, welcher einen Anteil von 22,6 %, macht. Durchschnittswert aller Antworten beträgt 7,4 Punkte.

Abbildung 20 - Kundenzufriedenheit



7. Analyse der Ergebnisse

Um die Umfrageergebnisse in Beziehung zu setzen, ist es wichtig zu wissen, wie die einzelnen Fragen in Verbindung zueinander stehen. Folgendermaßen kann die Auswirkung des gewählten Vorgehensmodells zur Entwicklung einer Software auf die Kundenzufriedenheit gezeigt werden.

7.1 Verifikation der Hypothese

Im Folgenden Kapitel soll die Hypothese mithilfe der gesammelten Daten verifiziert bzw. falsifiziert werden.

7.1.1 Hypothese

Kund:innen von Software-Festpreisprojekten, welche mit Agilen Vorgehensmodellen umgesetzt werden, sind zufriedener als Kund:innen von Software-Festpreisprojekten, die mit Linearen Vorgehensmodellen umgesetzt werden.

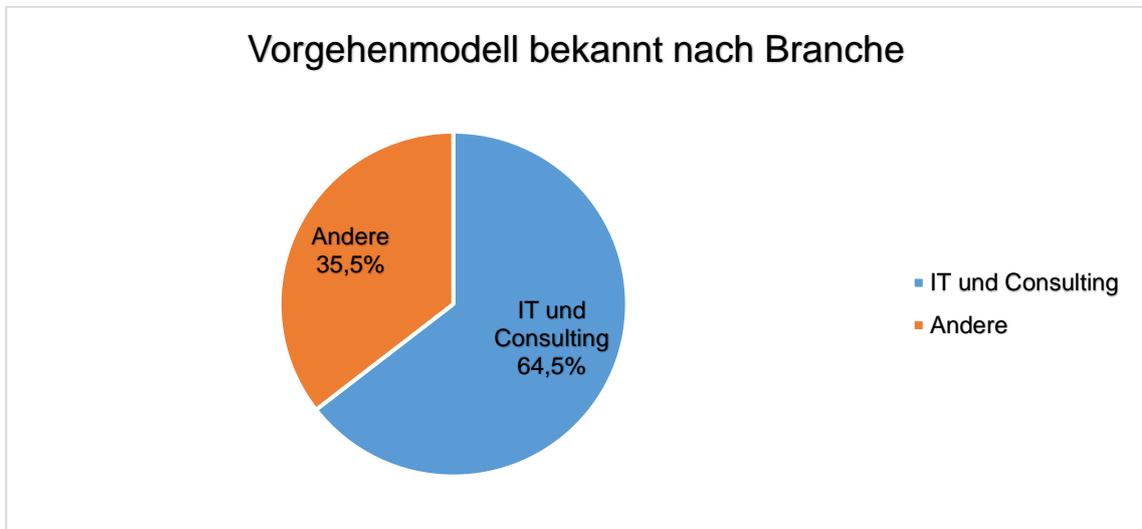
Um diese Hypothese beantworten zu können, benötigen man eine Kombination der Ergebnisse von mehreren Fragen, um zu einem Gesamtergebnis zu kommen.

7.1.2 Daten zur Hypothese

Vorgehensmodelle

Ausgangspunkt für die Einschränkung auf weitere Merkmale wie Kundenzufriedenheit und Festpreisprojekt ist das verwendete Vorgehensmodell, um eine Software zu entwickeln. Von allen Proband:innen haben 35,4 % diese Frage, ob ein Vorgehensmodell für ein eingesetztes Programm bekannt ist, mit Ja beantwortet. Der Großteil der Teilnehmer:innen, welche diese Frage mit einem positiven Ergebnis beantworten konnten, arbeitet im IT-Bereich. Fast zwei Drittel, genauer gesagt 64,5 %, sind in der Branche IT und Consulting, 35,5 % in anderen Sektoren beruflich tätig (siehe Abbildung 21).

Abbildung 21 - Vorgehensmodell bekannt nach Branche



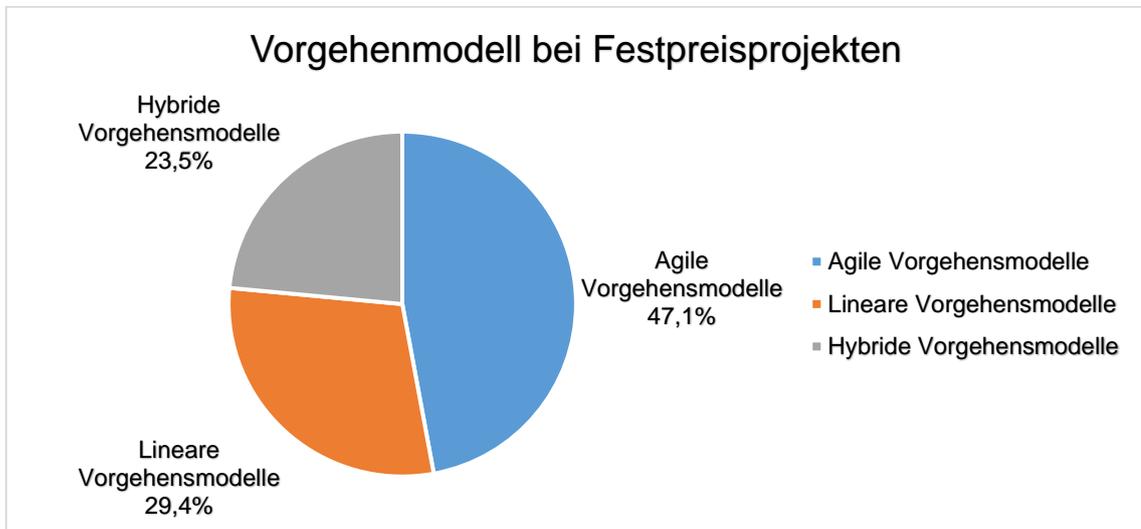
Festpreisprojekte

Wie bereits in Kapitel 6.2.3 ersichtlich, gaben 54,8 % aller befragten Mitwirkenden an, Software zu verwenden, von denen Preis und Umfang bereits zu Projektstart bekannt waren. Diese Daten können als Grundlage für Festpreisprojekte genommen werden. Die Frage wurde nicht allen Teilnehmer:innen gestellt, da ein bekanntes Vorgehensmodell für die Software eine Grundbedingung war, um diese Frage überhaupt beantworten zu können

Festpreisprojekte und verwendetes Vorgehensmodell

Von diesen 54,8 % der Antworten, die wir als Festpreis für die weiteren Analysen festsetzen können, wurde wiederum die Mehrheit mit Agilen Vorgehensmethoden entwickelt (siehe Abbildung 22). Diese Mehrheit, dargestellt in Form von 47,1 % der Gesamtmenge, liegt deutlich über Lineare Vorgehensmodelle, welche bei Festpreisprojekten einen Anteil von 29,4 % haben. 23,5 % fallen auf Hybride Vorgehensmodelle zurück. Die meisten Softwareprojekte werden aktuell demnach mit Agilen Vorgehensmethoden umgesetzt, auch wenn der Preis und Umfang bereits feststehen.

Abbildung 22 - Vorgehensmodelle bei Festpreisprojekten



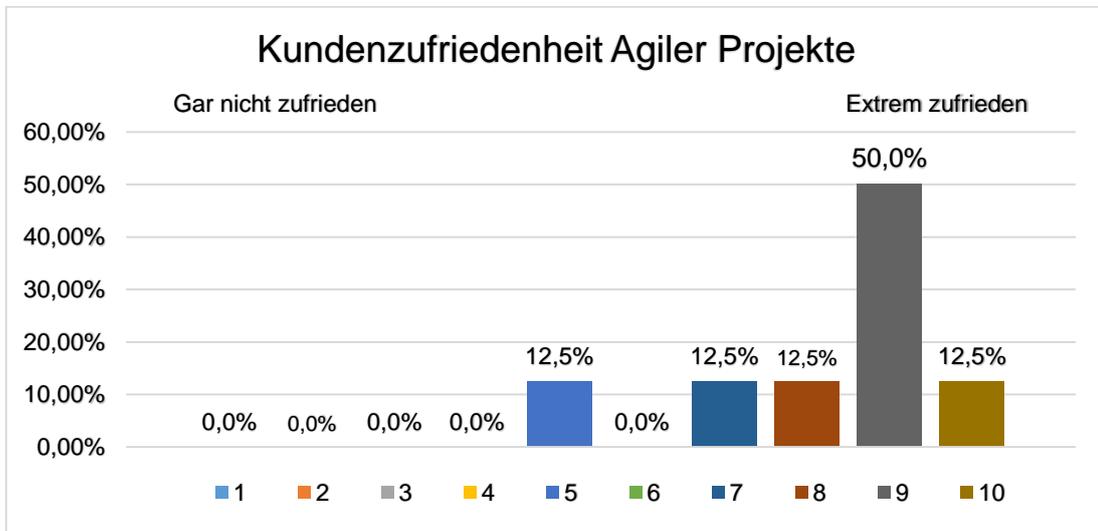
Kundenzufriedenheit nach Vorgehensmodell bei Festpreisprojekten

Schließlich sind die Daten so weit eingegrenzt, dass man aufgrund der zuvor gefilterten Daten, die durchschnittliche Kundenbewertung auf Festpreisprojektbasis und Vorgehensmodell eingrenzen kann.

Agile Vorgehensmodelle

Die meisten befragten Personen gaben an, mit der Software auf einer Punkteskala von 1-10 mit 9 Punkten zufrieden zu sein. Dies entspricht 50 % aller Befragten, die Festpreis bereits mit Ja beantwortet, und ein eine Software mit Agilem Vorgehensmodell gewählt haben. Die Anzahl an positiven Bewertungen (7 – 10) ist demnach in der deutlichen Überzahl. Nur 12,5 % aller Antworten liegen im neutralen Bereich, wobei keine Antworten im negativen Bereich sind. (siehe Abbildung 23).

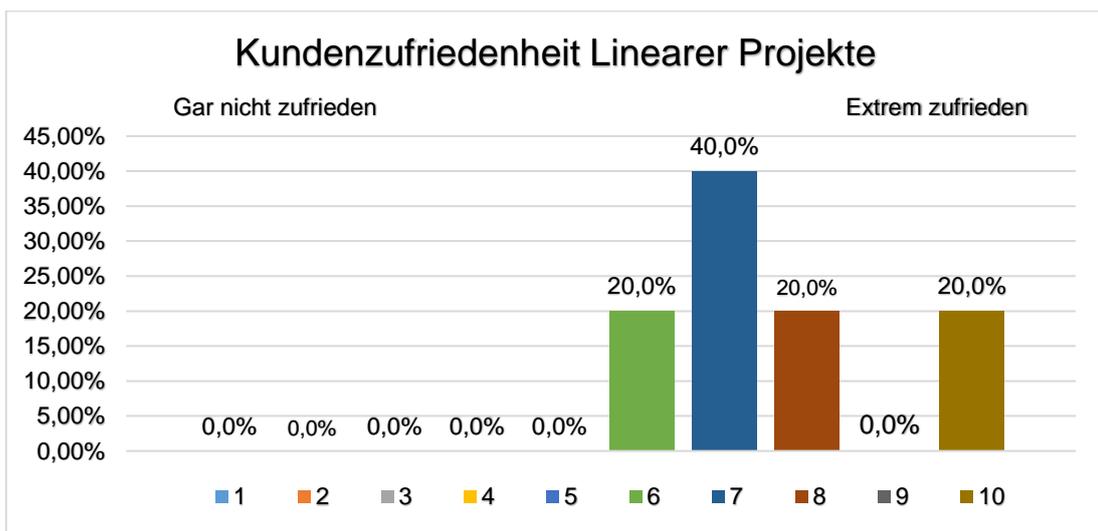
Abbildung 23 - Kundenzufriedenheit bei Agilen Projekten



Lineare Vorgehensmodelle

Bei den Linearen Vorgehensmodellen verhält es sich ähnlich wie bei den zuvor erwähnten Agilen Vorgehensmodellen, jedoch gibt es hier keine einzige Antwort unter 6 Punkten. Alle in der Umfrage angegebenen Bewertungen für Lineare Vorgehensmodelle liegen in Wertebereich 6 – 10 (siehe Abbildung 24).

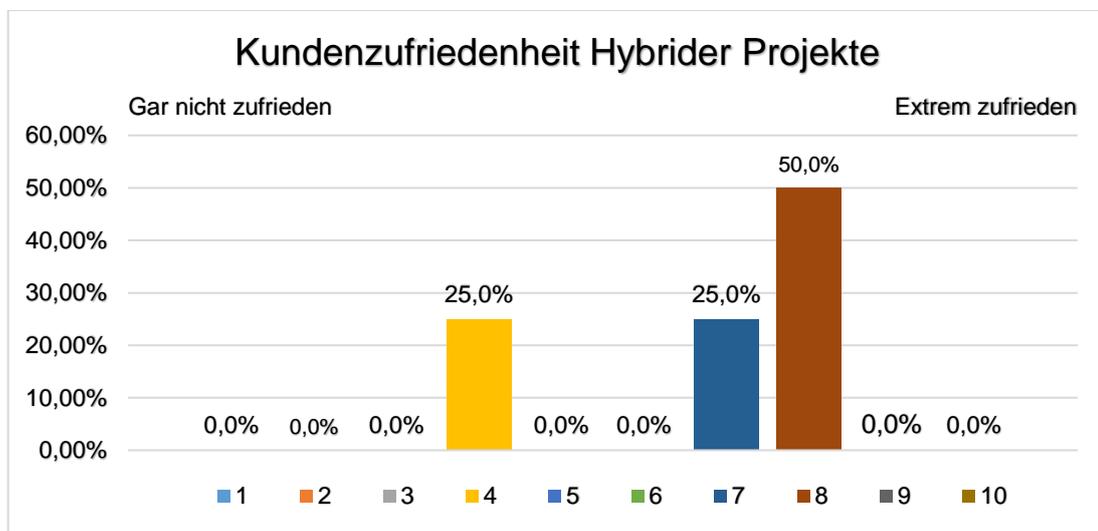
Abbildung 24 - Kundenzufriedenheit bei Linearen Vorgehensmodellen



Hybride Vorgehensmodelle

Die Kundenzufriedenheit bei Hybriden Vorgehensmodellen ist ähnlich positiv wie bei den beiden vorher genannten Modellen. Von den Proband:innen gaben ebenso die meisten an, dem Softwareprodukt eher positiv und zufrieden gegenüber zu stehen. Wie im Diagramm ersichtlich (siehe Abbildung 25), ist bei den meisten Teilnehmer:innen die Wahl in der vorhandenen Punkteskala auf 8 gefallen. Es gibt jedoch keine Antworten über 8, und mit 4 Punkten auch den geringsten Einzelwert bei der Zufriedenheit im Vergleich aller Vorgehensmodelle.

Abbildung 25 - Kundenzufriedenheit bei Hybriden Vorgehensmodellen



Quelle: Eigene Darstellung

7.1.3 Evaluierung Hypothese

Customer Satisfaction Score

Um die Kundenzufriedenheiten einzelner Vorgehensmodelle miteinander zu vergleichen, wird der Key Performance Indicator CSAT zur Evaluierung herangezogen (siehe Kapitel 3.3.2). Mithilfe dessen, kann man die Kundenzufriedenheit auf einen Prozentwert herunter brechen und die Vorgehensmodelle werden somit vergleichbar. Je höher der Prozentwert ist, desto zufriedener sind Kund:innen mit dem gewählten Produkt bzw. der von ihnen gewählten Software. Ebenso ist auch ein Durchschnittswert über alle Antworten möglich (Pecoraro 2021).

Der CSAT für die Berechnung nimmt alle positiven Bewertungen heran, was in unserem Fall ab 7 Punkte liegt. Die Punkteverteilung auf unsere Skala legt nahe, dass 1-3 eher negativ, 4-6 als neutral, und 7-10 als eher positiv angesehen wird.

Softwareprogramme, welche mit **Agilen Vorgehensmethoden** entwickelt wurden, weisen einen CATS-Prozentsatz von **87,5 %**, und einen CATS-Durchschnitt von **8,25** auf.

Bei den **Linearen Vorgehensmodellen** ist der CSAT-Prozentwert von **80 %** und CSAT-Durchschnitt von **7,6** jedoch etwas tiefer.

Im Vergleich schneidet das **Hybride Vorgehensmodell** in der Umfrage am schlechtesten ab. Der CSAT-Prozentwert liegt bei **75 %**, der CSAT-Durchschnitt bei **6,75** Punkten.

Grundsätzlich wurden jedoch alle Softwareprogramme eher positiv bewertet, was sich in einem CATS-Score von über 70 % spiegelt.

Beantwortung der Forschungsfrage

Wie zuvor berechnet, ist der Durchschnitt als auch der CSAT-Prozent bei Softwareprojekten, die mittels Agiler Methoden entwickelt wurden, höher als im Vergleich zu anderen Vorgehensmodellen. Um diese statistisch miteinander vergleichen zu können, benötigen wir den p-Wert, welcher das Ergebnis eines Signifikanztests zur Prüfung einer zuvor aufgestellten (Null) Hypothese ist: Ist der p-Wert kleiner als das gewählte Irrtumsniveau (Signifikanzniveau α), dann wird das Ergebnis als statistisch signifikant angesehen (Bender und Lange 2002).

Bei unseren Daten ergab sich ein **p-Wert** von **0,24**. Bei einem gewählten Signifikanzniveau von **$\alpha = 0,05$** liegt dieser p-Wert über dem festgelegten Signifikanzniveau. Der Unterschied bei den CSAT-Werten ist somit **nicht signifikant** und kann **nicht** als **systematisch höher** eingestuft werden.

Somit kann **nicht verifiziert** werden, dass Kund:innen von Software-Festpreisprojekten, welche mit Agilen Vorgehensmodellen umgesetzt werden, zufriedener sind als Kund:innen von Software-Festpreisprojekten, die mit Linearen Vorgehensmodellen umgesetzt worden sind. Dies gilt ebenso für Projekte mittels Hybrider Vorgehensmodelle.

Die Umfrage unterstreicht ebenso die Ergebnisse, die in der Literaturstudie gefunden wurden. Die Hypothese ist damit mit den vorliegenden Daten der Umfrage **falsifiziert**.

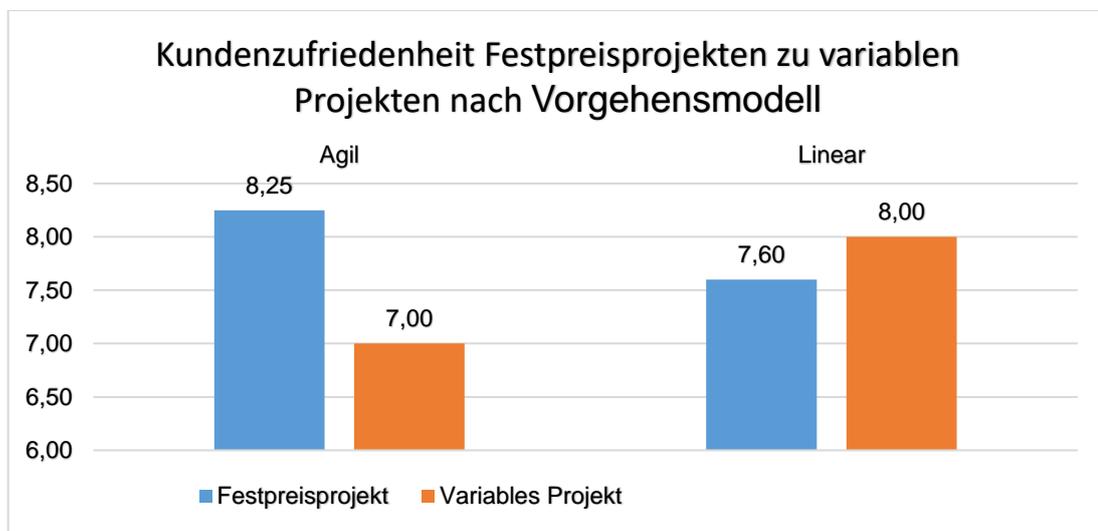
7.2 Weitere Ergebnisse

7.2.1 Festpreisprojekte und variable Projekte

Für die Hypothese waren zuvor zwingend Festpreisprojekte notwendig, damit diese überhaupt verifiziert werden könne. Zusätzlich gab es in der Umfrage jedoch einige Projekte, welche zum Projektstart noch keine Festpreisprojekte waren. Je nach

Vorgehensmodell gibt es hier einige Unterschiede. Demnach ist die Kundenzufriedenheit bei Festpreisprojekten bei Agilen Vorgehensmodellen mit einem Schnitt von 8,25 Punkten höher als bei variablen Projekten (siehe Abbildung 26), welche einen Schnitt von 7,00 Punkten aufweisen. Genau andersherum ist es bei Linearen Vorgehensmodellen. Während bei Festpreisprojekten ein Schnitt von 7,60 Punkten vorliegt, steigt der Durchschnitt für variable Projekte auf 8,00 Punkte. Für Hybride Vorgehensmodelle liegen keine Vergleichsdaten für variable Projekte zur Verfügung, deshalb kann hier nicht mit Festpreisprojektdaten verglichen werden.

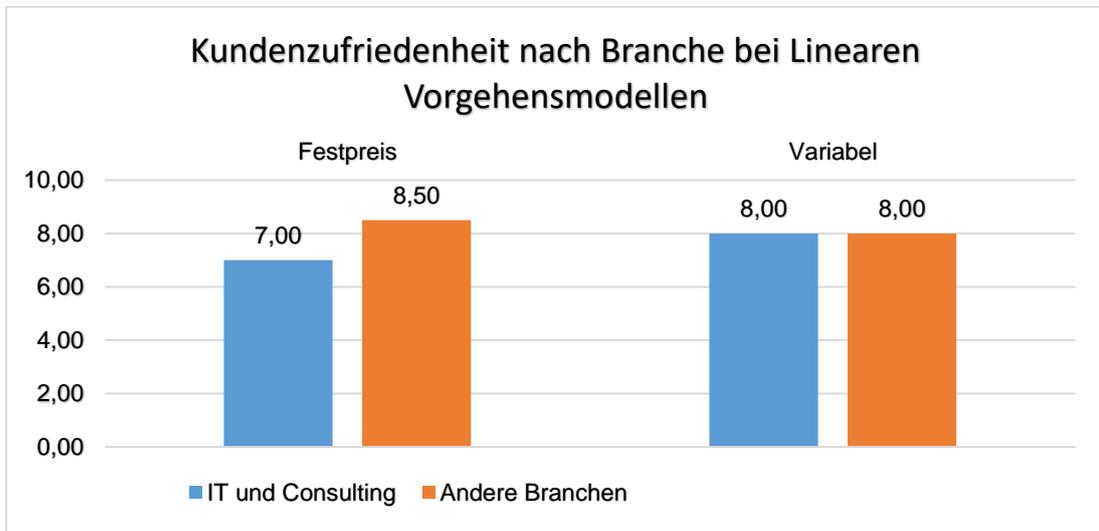
Abbildung 26 - Kundenzufriedenheit nach Projektart und Vorgehensmodell



7.2.2 Kundenzufriedenheit und Branche

Für den Vergleich zwischen Kundenzufriedenheit und Branche liegen nicht genügend Daten vor. Die Kundenzufriedenheit für Agile Softwarefestpreisprojekte wurde ausschließlich von Personen bewertet, die in der Branche IT und Consulting tätig sind. Ähnlich verhält es sich mit Hybriden Vorgehensmodellen. Für eine aussagekräftige Auflistung der Unterschiede liegen demnach nicht genügend Ergebnisse aus anderen Branchen vor. Bei den Linearen Vorgehensmodellen wurden jedoch gleich viele Bewertungen von Personen die in der IT Branche arbeitend abgegeben, wie von Personen, die in anderen Branchen tätig sind. So liegt der Durchschnitt aller Bewertungen bei variablen Projekten bei 8,00 und unterscheidet sich nicht zwischen den einzelnen Branchen. Bei Festpreisprojekten sieht dies jedoch anders aus. Arbeitende Personen in der IT-Branche bewerteten Festpreisprojektsoftware lediglich mit 7,00 Punkten. Im Gegensatz dazu bewerteten Proband:innen aus den anderen Branchen die Software mit 8,50 Punkten (siehe Abbildung 27).

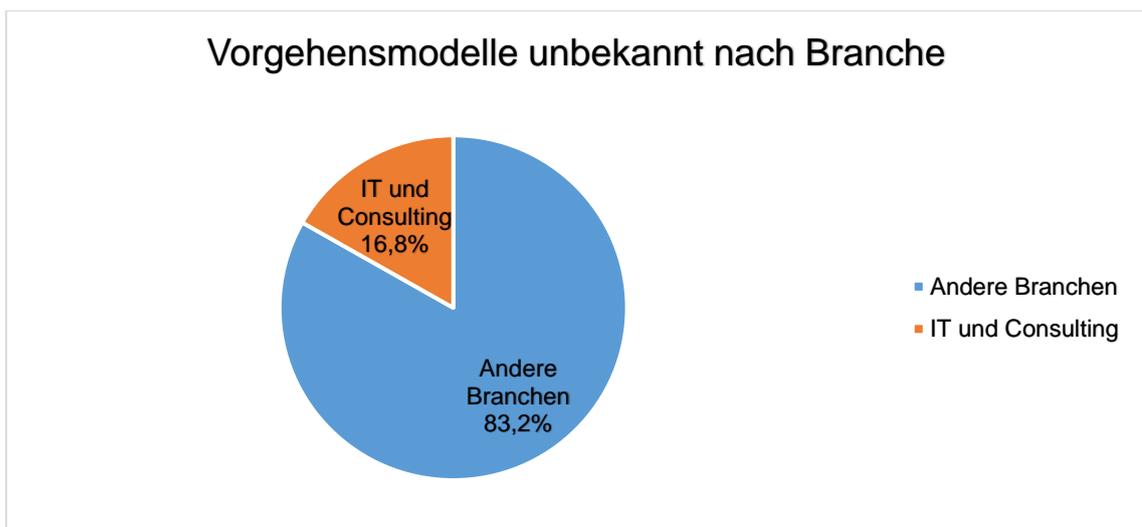
Abbildung 27 - Kundenzufriedenheit nach Branche



7.2.3 Vorgehensmodelle unbekannt nach Branche

Von allen befragten Personen gab es eine doch nicht so kleine Anzahl an Personen, die keines der genannten Vorgehensmodelle gekannt hat. Dies waren wie in Kapitel 6.6.2 erwähnt 35,6 % aller Befragten. Von diesem Anteil ausgehend, arbeiten 16,9 % in der IT und Consulting Branche, und die große Mehrheit von 83,9 % in anderen Branchen.

Abbildung 28 - Vorgehensmodelle unbekannt nach Branchen



8. Schlussfolgerungen

8.1 Einordnung der Ergebnisse

Vorgehensmodelle, egal ob linear, agil oder eine Hybridvariante davon, haben wesentlichen Einfluss, wie ein Projekt abgearbeitet und entwickelt wird. Ebenso bestimmen Faktoren wie Zeit, Budget und Projektkomplexität welches Vorgehensmodell, wann eingesetzt werden sollte.

Die Umfrage hat jedoch gezeigt, dass das Vorgehensmodell nach erfolgreicher Abarbeitung des Projektes keinen nennenswerten Einfluss auf die Kundenzufriedenheit hat. Zumindest kann dies im Rahmen dieser Studie nicht abgeleitet werden. Die Kundenzufriedenheit über den CSAT war zwar tendenziell bei Agilen Softwareprojekten höher, jedoch nicht in einem Rahmen, in dem ich dies erwartet hätte, und auch statistisch gesehen signifikant wäre. Dass sich der CSAT nicht bedeutsam von anderen Programmen, welche mit Linearen oder Hybriden Vorgehensmodellen entwickelt wurden, unterscheidet, war so nicht vorhersehbar und widerlegt die Hypothese dieser Arbeit.

Grundsätzlich kann gesagt werden, dass die Bewertung der Software gut bis sehr gut ausgefallen ist. Kaum eine befragte Person hat negativ über die von ihrer ausgewählten Software geurteilt. Die Auswahl der Software war frei, eventuell haben die Personen eher Software gewählt, die von ihrer Seite aus angenehm zu bedienen ist. Trotzdem ist wie bereits angesprochen kein Unterschied zwischen den Vorgehensmodellen erkennbar. Dies zeigt auch, dass man mit jedem Vorgehensmodell sehr gute Ergebnisse hinsichtlich Qualität und Kundenzufriedenheit erzielen kann.

Ein möglicher Grund dafür ist, dass heutzutage viele Agile Praktiken bei der Softwareentwicklung auch in Lineare Vorgehensmethoden einfließen. Versionskontrollsysteme und Continuous Integration werden nicht nur in Agilen Vorgehensmethoden verwendet. Dies trägt auch dazu bei, dass in traditionellen Projekten schneller ein Fehler behoben und ausgeliefert werden kann, was sich wiederum in der Kundenzufriedenheit widerspiegeln dürfte.

Interessant ist auch, dass Hybride Vorgehensmodelle vergleichsweise am schlechtesten abschneiden. Auch in Hinsicht auf den Umstand, weil dieses Vorgehensmodell eigentlich dafür prädestiniert wäre, Projekte mit fixem Umfang und Preis agil abarbeiten zu können.

Der Fokus der Befragung und somit der Studie hat sich vor allem auf Festpreisprojekte bezogen. Trotzdem konnte auch festgestellt werden, dass die Daten hinsichtlich Zufriedenheit ebenso für variable Projekte gelten.

Wenig überraschend war, dass das Wissen über Vorgehensmodelle für Personen, die im IT-Umfeld arbeiten, bedeutend höher war. Da die meisten Vorgehensmodelle sehr eng mit der IT verbunden sind, war dies anzunehmen.

Ein wesentlicher Faktor, der ebenfalls aus der Umfrage hervorgeht, ist, wie viele Personen in Unternehmen arbeiten, die Software verwenden, welche speziell für ihr Unternehmen entwickelt worden ist. Mehr als Dreiviertel der Proband:innen haben dies bejaht.

Zusammenfassend lässt sich sagen, dass sich die Umfrage in die sehr spärlich vorhandenen Studienergebnisse einordnen lässt. Die Daten decken sich mit den bereits bestehenden Erkenntnissen zu Kundenzufriedenheit und Vorgehensmodell.

8.2 Stärken und Schwächen der Studie

Mithilfe weniger ausgewählter Fragen konnte eine statistische Auswertung über die Kundenzufriedenheit und eingesetztem Vorgehensmodell erstellt werden. Der Zusammenhang von theoretischer Literatur zu Agilen Vorgehensmethoden und empirischen Daten aus den Umfragen konnte hergestellt werden. Der Fragebogen war konkret und kurz gestaltet, sodass die Rücklaufquote bei einem komplexen Thema hoch geblieben ist. Diese kann mit 92,4 % als durchaus sehr gut angesehen werden. Ebenso waren die Antwortmöglichkeiten vorgegeben und somit gut messbar. Die Aufteilung in Teilbereiche hat hierbei wohl ebenso beigetragen. Des Weiteren glaube ich, dass mithilfe dieser Umfrage die Aussagekraft höher ist, als dies durch Experteninterviews der Fall gewesen wäre. Die Anzahl der Antworten über die Kundenzufriedenheit ist in diesem Falle wohl deutlich höher. Ebenso können Expert:innen im IT-Bereich zwar oft die Entwicklungssicht darstellen, jedoch fehlen auch von User:innensicht die Primärdaten zur Zufriedenheit.

Wie zu erwarten, war die Kenntnis über Vorgehensmodelle über die Gesamtstichprobe mäßig. Das war für mich nicht überraschend, da es ein sehr spezielles Thema ist. Antworten für Agile Vorgehensmethoden und Hybride Vorgehensmethoden blieben für befragte Personen, welche nicht in der IT arbeiten, leider unter einem aussagekräftigen Wert. Immerhin konnte gezeigt werden, dass sich für Lineare Vorgehensmodelle die Unterschiede im Durchschnitt nicht wesentlich von IT-Branche und anderer Branchen unterscheiden. Die angewandte Quotenstichprobe hat sich aus meiner Sicht als ein gutes Mittel herausgestellt, die Stichprobe zu unterteilen und so zumindest teilweise die Antworten der Expertengruppe auf die Allgemeinheit umzulegen.

Trotzdem ist für die Beantwortung der Fragen zu den Vorgehensmodellen nur ein Teilbereich der Softwareanwender:innen fähig und somit ist das Ergebnis der Studie mit Vorsicht zu genießen. Auch wenn die Abweichungen mithilfe statistischer Auswertungsmethoden berücksichtigt wurden, so ist davon auszugehen, dass die Umfrage nur in begrenztem Maße für alle Softwareprogramme repräsentativ ist. In

gewisser Weise kann man sogar sagen, dass lediglich bei Personen mit IT-Fachwissen die Kundenzufriedenheit nicht wesentlich abweicht, egal welches Vorgehensmodell zur Programmierung dessen verwendet wurde. Ebenso ist die Aussage über eine Kundenzufriedenheit rein subjektiv und einzig von der befragten Person abhängig. Zudem gibt diese Angabe nur eine Aussage über die Zufriedenheit über einen definierten Zeitpunkt.

8.3 Conclusio

Agile Vorgehensmodelle beeinflussen nicht zwangsläufig die Kundenzufriedenheit in einem positiven Sinne. Je nach Projektgröße, Komplexität, Zeitplan und Budget kann das eine oder das andere Vorgehensmodell besser zur Umsetzung einer Software geeignet sein.

Die Umfrage sowie auch die Literaturstudie haben gezeigt, dass es zwar Indizien, dass Agile Vorgehensmethoden die Kundenzufriedenheit positiv beeinflussen, gibt, so kann dies nicht final geklärt bzw. verifiziert werden. Weder existierende Studien noch diese Erhebung haben einen signifikanten Unterschied in der Kundenzufriedenheit von Anwendersoftware gezeigt. Obwohl die Kundenzufriedenheit bei Agilen Projekten tendenziell höher scheint, so ist der Differenz in der Kundenzufriedenheit zu gering, um dies final zu bestätigen. Hybride Vorgehensmodelle fügen sich ebenso nahtlos in diese Thematik ein. Die Kundenzufriedenheit liegt auch bei Projekten, welche mit Hybriden Vorgehensmodellen entwickelt wurden, in einem ähnlichen Niveau.

Das Vorgehensmodell, welches verwendet wurde, um ein Programm zu entwickeln, bestimmt somit nicht die Kundenzufriedenheit der User:innen der Software. Dies trifft sowohl auf Festpreisprojekte als auch auf variable Projekte, zu. Beide Projektvarianten erreichten einen CSAT Durchschnittswert von mindestens 7 Punkte, was als positiv gewertet werden kann. Einen Unterschied gibt es jedoch bei der Abwicklung des Projektes. Projekte mit Agilen Vorgehensmodellen scheitern weniger oft als dies Lineare Projekte tun. Ebenso gibt es einige Aspekte, die sich positiv auf die Zufriedenheit des Projektteams auswirkt. Sozusagen beeinflusst das Vorgehensmodell zwar nicht die Kundenzufriedenheit des Endanwenders, jedoch jene des Auftraggebers. Somit kann verifiziert werden, dass Vorgehensmodelle zwar nicht die externe Qualität, jedoch die interne Qualität beeinflussen. Mehrere Studien weisen darauf hin, dass das Vorgehensmodell die Arbeitszufriedenheit und Produktivität der Mitarbeiter positiv oder negativ beeinflussen kann. Dies sollte bei der Evaluierung eines geeigneten Vorgehensmodells zum Projektstart eine zentrale Rolle spielen.

Trotzdem gilt, auf die Bedürfnisse der Kund:innen einzugehen, und diese mit höchster Sorgfalt umzusetzen. Damit ein Projekt als erfolgreich umgesetzt und somit als nicht gescheitert gezählt werden kann, so sind für dessen Erreichung Agile Vorgehensmethoden zu bevorzugen.

9. Literaturverzeichnis

- Abran, Alain, Hrsg. 2005. *Guide to the software engineering body of knowledge: 2004 version ; SWEBOOK*. Los Alamitos, Calif: IEEE Computer Soc.
- Amirova, Rozaliya, Ilya Khomyakov, Ruzilya Mirgalimova und Alberto Sillitti. 2019. Software Development and Customer Satisfaction: A Systematic Literature Review. In: *Software Technology: Methods and Tools*, hg. von Manuel Mazzara, Jean-Michel Bruel, Bertrand Meyer, und Alexander Petrenko, 11771:136–149. Lecture Notes in Computer Science. Cham: Springer International Publishing. doi:10.1007/978-3-030-29852-4_11, http://link.springer.com/10.1007/978-3-030-29852-4_11 (zugegriffen: 4. Dezember 2021).
- Andrei, Bogdan-Alexandru, Andrei-Cosmin Casu-Pop, Sorin-Catalin Gheorghe und Costin-Anton Boiangiu. 2019. A study on using waterfall and agile methods in software project management. *Journal Of Information Systems & Operations Management*. 125–135.
- Angermeier, Georg. 2005. Projekterfolg. *Glossar - Projekterfolg*. 17. August. <https://www.projektmagazin.de/glossarterm/projekterfolg> (zugegriffen: 4. Dezember 2021).
- Aston, Ben. 2021. 9 Of The Most Popular Project Management Methodologies Made Simple. 10. August. <https://thedigitalprojectmanager.com/project-management-methodologies-made-simple/>.
- Balzert, Helmut. 2001. *Lehrbuch der Software-Technik. 2. Auflage*. Heidelberg: Spektrum Akademischer Verlag.
- Bass, Len, Paul Clements und Rick Kazman. 2003. *Software architecture in practice*. 2nd ed. SEI series in software engineering. Boston: Addison-Wesley.
- Bassil, Youssef. 2012. A Simulation Model for the Waterfall Software Development Life Cycle. *ArXiv abs/1205.6904*.
- Beck, Kent. 2005. *Extreme Programming: die revolutionäre Methode für Softwareentwicklung in kleinen Teams ; [das Manifest]*. Studienausg., 2. Nachdr. Programmer's choice. München: Addison-Wesley.
- Beck, Kent und Martin Fowler. 2001. *Planning extreme programming*. The XP series. Boston: Addison-Wesley.
- Beck, Kent L., Michael A. Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, et al. 2001. Manifesto for Agile Software Development. In: .
- Bender, R und St Lange. 2002. Was ist der p -Wert? *DMW - Deutsche Medizinische Wochenschrift* 126, Nr. Suppl. Statistik (25. Februar): T 39-T 40. doi:10.1055/s-2001-12739, .
- Bergmann, Sebastian und Stefan Pribsch. 2010. *Softwarequalität in PHP-Projekten: Digg, eZ Components, studiVZ, swoodo, symfony, TYPO3, Zend Framework*. München: Hanser.
- Bleek, Wolf-Gideon und Henning Wolf. 2008. *Agile Softwareentwicklung: Werte, Konzepte und Methoden*. 1. Aufl. it-agile. Heidelberg: dpunkt.verlag.

- Brandt-Pook, Hans und Rainer Kollmeier. 2008. *Softwareentwicklung kompakt und verständlich: Wie Softwaresysteme entstehen*. Wiesbaden: Vieweg +Teubner / GWV Fachverlage GmbH, Wiesbaden. <https://doi.org/10.1007/978-3-8348-9507-3> (zugegriffen: 30. November 2021).
- Broy, Manfred und Andreas Rausch. 2005. Das neue V-Modell® XT. *Informatik-Spektrum* 28, Nr. 3 (1. Juni): 220–229. doi:10.1007/s00287-005-0488-z, .
- Cartaxo, Bruno, Allan Araujo, Antonio Sa Barreto und Sergio Soares. 2013. The Impact of Scrum on Customer Satisfaction: An Empirical Study. In: *2013 27th Brazilian Symposium on Software Engineering*, 129–136. Brasilia, Brazil: IEEE, Oktober. doi:10.1109/SBES.2013.10, <http://ieeexplore.ieee.org/document/6800189/> (zugegriffen: 27. April 2022).
- Cockburn, Alistair. 2002. *Agile software development*. Agile software development series. Boston: Addison-Wesley.
- Cohen, David, Mikael Lindvall und Patricia Costa. 2003. DACS State-of-the-Art / Practice Report Agile Software Development. In: .
- Döring, Nicola und Jürgen Bortz. 2016. *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften*. 5. vollständig überarbeitete, aktualisierte und erweiterte Auflage. Springer-Lehrbuch. Berlin Heidelberg: Springer. doi:10.1007/978-3-642-41089-5, .
- Duvall, Paul M., Steve Matyas und Andrew Glover. 2007. *Continuous integration: improving software quality and reducing risk*. Addison-Wesley signature series. Upper Saddle River, NJ: Addison-Wesley.
- Dybå, Tore und Torgeir Dingsøy. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50, Nr. 9–10 (August): 833–859. doi:10.1016/j.infsof.2008.01.006, .
- Engstler, Martin, Masud Fazal-Baqaie, Eckhart Hanser, Martin Mikusz, Alexander Volland, Gesellschaft für Informatik, Gesellschaft für Informatik und Gesellschaft für Informatik, Hrsg. 2015. *Projektmanagement und Vorgehensmodelle 2015: hybride Projektstrukturen erfolgreich umsetzen; gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V. ; 22. und 23. Oktober 2015 in Elmshorn*. Lecture notes in informatics (LNI). Proceedings volume P-250. Bonn: Gesellschaft für Informatik e.V. (GI).
- Fairley, R. E. 2009. *Managing and leading software projects*. Los Alamitos, CA : Hoboken, N.J: IEEE Computer Society ; Wiley.
- Faullant, Rita. 2007. *Psychologische Determinanten der Kundenzufriedenheit: der Einfluss von Emotionen und Persönlichkeit*. 1. Aufl. Gabler Edition Wissenschaft. Wiesbaden: Dt. Univ.-Verl.
- Festge, Fabian. 2007. *Kundenzufriedenheit und Kundenbindung im Investitionsgüterbereich*. Wiesbaden: Springer Fachmedien. <http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=748489> (zugegriffen: 4. Dezember 2021).
- Flanagan, David. 2011. *JavaScript: the definitive guide*. 6th ed. Beijing ; Sebastopol, CA: O'Reilly.

- Foerster, Bastian. 2015. Kundenzufriedenheit messen. 15. Oktober. <https://www.marketinginstitut.biz/blog/kundenzufriedenheit-messen/> (zugegriffen: 7. Januar 2015).
- Fowler, Martin. 2006. Continuous Integration. *Continuous Integration*. 1. Mai. <https://www.martinfowler.com/articles/continuousIntegration.html> (zugegriffen: 4. Dezember 2021).
- Gnatz, Michael Andreas Josef. 2005. *Vom Vorgehensmodell zum Projektplan*. München.
- Gnatz, Michael Andreas Josef. 2005. Vom Vorgehensmodell zum Projektplan. München: Institut für Informatik der Technischen Universität München, 28. Oktober. (zugegriffen: 16. Oktober 2021).
- Grady, Robert B. und Deborah L. Caswell. 1987. *Software metrics: establishing a company-wide program*. Englewood Cliffs, N.J: Prentice-Hall.
- Huber, Eberhard. 2011. Wann ist ein Projekt erfolgreich? *Wann ist ein Projekt erfolgreich?* 9. März. <https://www.pentaeder.de/projekte/2010/03/09/wann-ist-ein-projekt-erfolgreich/> (zugegriffen: 4. Dezember 2021).
- Iyawa, Gloria E., Marlien E Herselman und Alfred Coleman. 2016. Customer Interaction in Software Development: A Comparison of Software Methodologies Deployed in Namibian Software Firms. *The Electronic Journal of Information Systems in Developing Countries* 77, Nr. 1 (November): 1–13. doi:10.1002/j.1681-4835.2016.tb00560.x, .
- Jardina, Jo Rain, S. Camille Peres, Vickie Nguyen, Ashitra Megasari, Katherine R. Griggs, Rosalinda Pinales und April N. Amos. 2009. Keyboard Shortcut Users: They Are Faster at More than Just Typing. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 53, Nr. 15 (Oktober): 975–979. doi:10.1177/154193120905301508, .
- John Reh, F. 2020. The Basics of Key Performance Indicators. 19. März. <https://www.thebalancecareers.com/key-performance-indicators-2275156> (zugegriffen: 11. Dezember 2021).
- Kirchhoff, Sabine, Sonja Kuhnt, Peter Lipp und Siegfried Schlawin, Hrsg. 2010. *Der Fragebogen: Datenbasis, Konstruktion und Auswertung*. 5. Aufl. Lehrbuch. Wiesbaden: VS Verlag für Sozialwissenschaften.
- Kneuper, Ralf. 2018. *Software Processes and Life Cycle Models: An Introduction to Modelling, Using and Managing Agile, Plan-Driven and Hybrid Processes*. 1st ed. 2018. Cham: Springer International Publishing: Imprint: Springer. doi:10.1007/978-3-319-98845-0, .
- Krumbholz, Henrik. 2020. Anforderungen an gute Anforderungen. 12. Juni. <https://www.softwareone.com/de-at/blog/artikel/2020/03/11/requirements-engineering-dokumentation> (zugegriffen: 11. Dezember 2021).
- Kuß, Alfred. 2007. *Marktforschung: Grundlagen der Datenerhebung und Datenanalyse*. 2., überarb. und erw. Aufl. Lehrbuch. Wiesbaden: Gabler.
- LeMay, Matt und Frank Langenau. 2019. *Agil im ganzen Unternehmen Wie Sie eine dynamische, flexible und kundenorientierte Organisation gestalten*. <https://nbn->

- resolving.org/urn:nbn:de:101:1-2019091204013098132438 (zugegriffen: 27. November 2021).
- Liggemeyer, Peter. 2009. *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. 2. Aufl. Heidelberg: Spektrum Akad. Verl.
- Loeliger, Jon und Matthew McCullough. 2012. *Version control with Git*. Second edition. Beijing: O'Reilly.
- Malhotra, Naresh K. 2010. *Marketing research: an applied orientation*. 6th ed. Upper Saddle River, NJ: Pearson.
- Mikusz, Martin, Alexander Volland, Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Oliver Linssen und Gesellschaft für Informatik, Hrsg. 2018. *Projektmanagement und Vorgehensmodelle 2018, PVM 2018: der Einfluss der Digitalisierung auf Projektmanagementmethoden und Entwicklungsprozesse: gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V. in Kooperation mit der Fachgruppe IT-Projektmanagement der GPM e.V.: 15. und 16. Oktober 2018 in Düsseldorf*. GI-Edition - Lecture Notes in Informatics (LNI) Proceedings\$lvolume P-286. Bonn: Gesellschaft für Informatik e.V. (GI).
- Moksony, Reka. 2014. When, Why, and How to use the Agile-Waterfall Hybrid Model. 14. Mai. <https://content.intland.com/blog/agile/when-why-how-to-use-the-hybrid-model>.
- Natrapei, Iulia. 2019. Was ist der Customer Satisfaction Score (CSAT)? 16. Dezember. <https://blog.hubspot.de/service/customer-satisfaction-score> (zugegriffen: 11. Dezember 2021).
- Nielsen, Jakob. 2020. 10 Usability Heuristics for User Interface Design. 15. November. <https://www.nngroup.com/articles/ten-usability-heuristics/>.
- O A. 2020. Key Performance Indicator. *sixclicks*. <https://www.sixclicks.de/wissen/glossar/key-performance-indicator> (zugegriffen: 11. Dezember 2021).
- O A. IEEE Guide for Software Requirements Specifications. IEEE. doi:10.1109/IEEESTD.1984.119205, <http://ieeexplore.ieee.org/document/278253/> (zugegriffen: 11. Dezember 2021).
- Oliver, CK. 2016. Microservices at Netflix: Stop system problems before they start. 25. April. <https://articles.microservices.com/microservices-at-netflix-stop-system-problems-before-they-start-e07c51a1d52d> (zugegriffen: 4. Januar 2022).
- Otte, Stefan. 2009. Version control systems. *Computer Systems and Telematics*: 11–13.
- Pecoraro, Caroline. 2021. CSAT: Zufriedene Kunden mit dem Customer Satisfaction Score ermitteln. 30. August. <https://www.netigate.net/de/articles/kundenzufriedenheit/customer-satisfaction-score-csat/> (zugegriffen: 11. Dezember 2021).
- Royce, Winston W. 1970. Managing the Development of Large Software Systems (1970). In: *Ideas That Created the Future: Classic Papers of Computer Science*, hg. von Harry R. Lewis, 0. The MIT Press. doi:10.7551/mitpress/12274.003.0035,

- <https://doi.org/10.7551/mitpress/12274.003.0035> (zugegriffen: 27. November 2021).
- Shiklo, Boris. 2019. 8 Vorgehensmodelle der Softwareentwicklung: mit Grafiken erklärt. 6. Oktober. <https://www.scnsoft.de/blog/vorgehensmodelle-der-softwareentwicklung> (zugegriffen: 1. März 2022).
- Singer, Otto. 2010. Aktueller Begriff: Cloud Computing. Fachbereich WD 10, Kultur, Medien und Sport, 12. März. https://www.bundestag.de/resource/blob/191178/22a7553089d81c2e06866e15fc354a0e/cloud_computing-data.pdf (zugegriffen: 30. November 2021).
- Statistik Austria. 2021. Internetnutzerinnen und Internetnutzer 2005 bis 2021. 3. November. http://www.statistik.at/wcm/idc/idcplg?IdcService=GET_PDF_FILE&RevisionSelectionMethod=LatestReleased&dDocName=053946 (zugegriffen: 10. Januar 2021).
- The Standish Group International. 2015. Chaos Report 2015. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf.
- Van Schaik, Tommy. 2020. Waterfall vs. agile. 31. August. <https://www.pluralsight.com/blog/software-development/waterfall-vs-agile> (zugegriffen: 7. Januar 2022).
- . 2021. Move away from wagile: How to break anti-patterns in agile transitions. <https://www.pluralsight.com/blog/it-ops/move-away-from-wagile>.
- Verband Österreichischer Software Industrie. 2021. IT-Branche hat stark an Bedeutung gewonnen. 17. Juni. https://www.voesi.or.at/wp-content/uploads/2021/06/V%C3%96SI-PI-SOFTWARE-Studie-2021_06_17-1.pdf.
- Vivenzio, Alberto und Domenico Vivenzio. 2013. Vorgehensmodelle in der Softwareentwicklung. In: *Testmanagement bei SAP-Projekten: Erfolgreich Planen • Steuern • Reporten bei der Einführung von SAP-Banking*, hg. von Alberto Vivenzio und Domenico Vivenzio, 5–10. Wiesbaden: Springer Fachmedien Wiesbaden. doi:10.1007/978-3-8348-2142-3_2, https://doi.org/10.1007/978-3-8348-2142-3_2.
- WKO. 2022. Statistische Daten für Fachverbände - Aktuelle Branchenprofile und Kennzahlen. Wirtschaftskammer Österreich. 12. April. <https://www.wko.at/service/zahlen-daten-fakten/branchendaten-fachverbaende.html>.
- Wolff, Eberhard. 2018. *Das Microservices-Praxisbuch: Grundlagen, Konzepte und Rezepte*. 1. Auflage. Heidelberg: dpunkt.verlag.

10. Abbildungsverzeichnis

Abbildung 1 -Wasserfallmodell	5
Abbildung 2 - Gute und Schlechte Anforderungen analysieren.....	7
Abbildung 3 - Iterationen	10
Abbildung 4 - Bausteine agiler Softwareentwicklung	11
Abbildung 5 - Ablauf CI	15
Abbildung 6 - Kundenzufriedenheit.....	18
Abbildung 7 - Erfolgreiche IT Projekte	19
Abbildung 8 - Erfolgreiche IT Projekte nach Größe.....	20
Abbildung 9 - Vorgehensmodelle - Agil, Klassisch (Linear), Hybrid	23
Abbildung 10 - Key Performance Indicator	25
Abbildung 11 - Branchen.....	35
Abbildung 12 - Softwarenutzung beruflich	36
Abbildung 13 - Individualsoftwarenutzung im Unternehmen	36
Abbildung 14 - Softwarenutzung privat.....	37
Abbildung 15 - Kenntnis über grundlegende Vorgehensmodelle	38
Abbildung 16 - Vorgehensmodelle.....	39
Abbildung 17 - Vorgehensmodell für Software bekannt	40
Abbildung 18 - Einsatz von Vorgehensmodellen	41
Abbildung 19 - Übersicht Fixpreisprojekt	41
Abbildung 20 - Kundenzufriedenheit.....	42
Abbildung 21 - Vorgehensmodell bekannt nach Branche	44
Abbildung 22 - Vorgehensmodelle bei Festpreisprojekten.....	45
Abbildung 23 - Kundenzufriedenheit bei Agilen Projekten	46
Abbildung 24 - Kundenzufriedenheit bei Linearen Vorgehensmodellen.....	46
Abbildung 25 - Kundenzufriedenheit bei Hybriden Vorgehensmodellen.....	47
Abbildung 26 - Kundenzufriedenheit nach Projektart und Vorgehensmodell.....	49
Abbildung 27 - Kundenzufriedenheit nach Branche.....	50
Abbildung 28 - Vorgehensmodelle unbekannt nach Branchen	50

11. Appendix

Online Fragebogen



Sehr geehrte Damen und Herren,

vielen Dank, dass Sie sich die Zeit nehmen, um mich bei meiner Arbeit zu unterstützen.

Im Rahmen meiner Bachelorarbeit des Studiengangs Wirtschaftsinformatik auf der Ferdinand Porsche FernFH beschäftige ich mich mit dem Thema Kundenzufriedenheit bei Softwarelösungen und eingesetztem Vorgehensmodell bei der Entwicklung des Produktes.

christoph.kerschenbauer@mail.fernfh.ac.at

Umfrage starten

press Enter ↵

“ Die Umfrage gliedert sich in 3 Sektionen

Sektion 1 Persönliche Angaben

Sektion 2 Vorgehensmodelle

Sektion 3 Softwarelösungen

Continue press Enter ↵

1 → Section 1 | Persönliche Angaben

Continue press Enter ↵

a. In welcher Branche sind Sie beruflich tätig? *

A Bank und Versicherung

B Gewerbe und Handwerk

C Handel

D Industrie

E IT und Consulting

F Tourismus und Freizeitwirtschaft

G Transport und Verkehr

H Sonstiges

I ich bin derzeit nicht beruflich tätig

- b. Arbeiten Sie beruflich mit Softwareprogrammen bzw. Softwarelösungen? Dazu zählen auch webbasierte Softwaresysteme die im Browser ausgeführt werden *

z.B Tools die in Ihrem Unternehmen eingesetzt werden wie spezielle Software für Ihren Bereich, Zeitbuchungssysteme, Zeichenprogramme oder irgendwelche anderen Programme.

A Ja

B Nein

C Ich weiß es nicht

- c. Arbeiten Sie in einem Unternehmen das Software bzw. Programme einsetzt, die speziell für Ihr Unternehmen entwickelt worden sind? (Individualsoftware) *

z.B. Zeitbuchungen, Interne Softwarelösungen, Kundenverwaltungen, Bestellprogramme etc.

A Ja

B Nein

C ich weiß es nicht



d. Arbeiten Sie privat mit Softwareprogrammen bzw. Softwarelösungen? *

z.B. Tools für SmartHome, Streamingdienste, Zeichenprogramme oder andere Programme.

A Ja

B Nein

C ich weiß es nicht



Powered by **Typeform**

2 → Section 2 | Vorgehensmodelle

Continue press Enter ↵

a. Welche grundlegenden Vorgehensmodelle zur Softwareentwicklung sind Ihnen bekannt? *

Choose as many as you like

A Lineare Vorgehensmodelle

B Agile Vorgehensmodelle

C Hybride Vorgehensmodelle

D Keines

OK ✓



Powered by **Typeform**

b. Wie schätzen Sie Ihren persönlichen Wissensstand zu den folgenden Vorgehensmodellen ein? *

Welche der folgenden Vorgehensmodelle sind Ihnen bekannt?

	ausgeprägtes Wissen/Erfahrung	Basiswissen	Unbekannt
Scrum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kanban	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Extreme Programming	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Wasserfallmodell	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
V-Modell	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Wagile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Spiralmodell	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rational Unified Process (RUP)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
noch weitere/noch andere	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

OK ✓



Powered by Typeform

3 → Section 3 | Softwarelösungen

Continue press Enter ↵

- a. Benutzen Sie eine Software bei der Ihnen das Vorgehensmodell, mit dem die Software entwickelt wurde, bekannt ist? *

A Ja

B Nein

b. Mit welchem Vorgehensmodell wurde die Software entwickelt? *

A Agiles Vorgehensmodell (z.B. Scrum, XP etc.)

B Lineares Vorgehensmodell (z.B. Wasserfallmodell etc.)

C Hybrides Vorgehensmodell (Mischung aus beiden - z.B. Pflichtenheft + agile Entwicklung)

c. Waren beim Softwareprojekt zum Entwicklungsstart bereits Umfang und Preis bekannt? (Fixpreisprojekt) *

A Ja

B Nein

C Ich weiß es nicht



d. Wie zufrieden sind Sie mit dem gewählten Softwareprodukt? *

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Gar nicht zufrieden

Extrem zufrieden

Submit



Powered by **Typeform**

d. Wie zufrieden sind Sie mit dem gewählten Softwareprodukt? *

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Gar nicht zufrieden

Extrem zufrieden

Submit

FERDINAND PORSCHE



Vielen Dank für Ihre Teilnahme an der
Umfrage.

Christoph Kerschenbauer

How you ask is everything [Create a typeform](#)