

BACHELORARBEIT

GEWICHTETER KRITERIENKATALOG ZU BEWERTUNG AGILER TEAMS

in KMUs im Bereich Dienstleistungen in der automatischen Datenverarbeitung und
Informationstechnik

ausgeführt an der



FACHHOCHSCHULE DER WIRTSCHAFT

am Studiengang
Wirtschaftsinformatik

Von: Harald Beier
Matrikelnummer: 00640165

Graz, am 04.10.2021

.....
Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.....

Unterschrift

KURZFASSUNG

Kleine und mittlere Unternehmen sind das Rückgrat der österreichischen Wirtschaft. Viele Technologie- und Softwareunternehmen sind in Graz angesiedelt. In dieser Arbeit wird ein gewichteter Kriterienkatalog erstellt, der auf den bisherigen Erkenntnissen des Autors aufbaut. Diese Kriterien sollen KMUs helfen bei der Einführung und Etablierung effizienter Kriterien zur Messung der Produktivität in agilen Softwareentwicklungsteams. Es wird eine qualitative Befragung von Softwaremanagern und Lektoren für Softwareentwicklung an Grazer Hochschulen durchgeführt. Die Ergebnisse deuten darauf hin, dass ein Fokus auf Produktqualität den wirtschaftlichen Nutzen für KMUs maximiert. KMUs sollten sich auf einige wenige Schlüsselkriterien konzentrieren, um die Produktivität ihrer agilen Softwareentwicklungsteams zu messen, und diese zuerst vollständig implementieren, bevor zusätzliche Kriterien eingeführt werden. Die vorgeschlagenen Schlüsselkriterien sind die Einführung von einem umfassenden Softwaretestplan, Messung von Bugs und dazugehöriger Lösungszeiten, korrektes und präzise Anforderungserhebung sowie Kundenzufriedenheit.

ABSTRACT

Small and medium-sized enterprises (SMEs) are the backbone of the Austrian economy. Many tech and software companies are situated in Graz. This thesis establishes a weighted catalogue of criteria building on the author's previous findings. These criteria help SMEs introduce and establish efficient criteria to measure productivity in agile software development teams. A qualitative survey of software managers and professors for software development at universities in Graz is conducted. The results suggest that a focus on product quality maximizes economic utility for SMEs. SMEs should focus on a few key criteria to measure the productivity of their agile software development teams and fully implement them first when additional ones are introduced. The suggested key criteria are test plans and test protocols, measuring of bugs, correct requirements engineering and customer satisfaction.

INHALTSVERZEICHNIS

1	EINLEITUNG	1
1.1	Problemstellung	1
1.2	Zielsetzung	1
1.3	Forschungsfrage	1
1.4	Vorgehen und Methode	2
2	GRUNDLAGE PRODUKTIVITÄT UND AGILE SOFTWARE ENTWICKLUNG	3
2.1	Generelle Definition Produktivität	3
2.2	Produktivität in der Softwareentwicklung	4
2.3	Definition Agilität und agile Softwareentwicklung	5
2.4	Performance Indikatoren und Metriken	8
2.5	Entwicklung des priorisierten Kriterienkataloges	10
2.5.1	Kriterien Bildung	10
2.5.2	Gewichtung	11
3	METHODIK	14
3.1	Experteninterview	14
3.1.1	Auswahl der Expertinnen und Experten	14
3.1.2	Durchführung des Interviews	15
3.1.3	Aufbau des Interviews	15
3.2	Qualitative Inhaltsanalyse nach Mayring	17
3.2.1	Deduktive Kategorienbildung	17
3.2.2	Erstellung des Datenmaterials	18
3.2.3	Kodierung	19
4	AUSWERTUNG DER EXPERTENINNENINTERVIEWS	20
4.1	Agile Softwareentwicklung	20
4.1.1	Begriffsverständnis agile Softwareentwicklung und agile Teams	21
4.1.2	Metriken innerhalb der agilen Softwareentwicklung	21
4.1.3	Priorität bzw. Präferenz bei Kennzahlen	22

4.2	Produktqualität.....	22
4.2.1	Begriffsverständnis von Produktqualität in der agilen Softwareentwicklung	23
4.2.2	Bekannte Metriken und Eignungsgrad für die Produktqualität	23
4.2.3	Einsatz von Metriken zur Messung der Produktqualität innerhalb der Firma.....	24
4.3	Teamproduktivität	24
4.3.1	Begriffsverständnis von Teamproduktivität in der agilen Softwareentwicklung	26
4.3.2	Bekannte Metriken und Eignungsgrad für die Teamproduktivität	26
4.3.3	Einsatz von Metriken zur Messung der Teamproduktivität innerhalb der Firma	27
4.4	Vorhersagbarkeit	27
4.4.1	Begriffsverständnis von Vorhersagbarkeit in der agilen Softwareentwicklung.....	29
4.4.2	Bekannte Metriken und Eignungsgrad für die Vorhersagbarkeit.....	29
4.4.3	Einsatz von Metriken zur Messung der Vorhersagbarkeit innerhalb der Firma	30
5	DISKUSSION	31
6	FAZIT 32	
6.1	Forschungsfrage – Erkenntnisse	32
6.2	Ergänzung des Kriterienkataloges.....	32
6.3	Limitationen	35
6.4	Ausblick	35
	ANHANG A - INTERVIEWLEITFADEN	36
	ANHANG B - KODIERLEITFADEN	37
	ANHANG C - QUALITATIVE INHALTSANALYSE	41
	ABKÜRZUNGSVERZEICHNIS.....	54
	ABBILDUNGSVERZEICHNIS	55
	TABELLENVERZEICHNIS	56
	LITERATURVERZEICHNIS.....	57

1 EINLEITUNG

Agile Softwareentwicklung hat seit dem Jahr 2001 in die meisten Softwareunternehmen Einzug gehalten, zumindest in bestimmten Bereichen. Die Firmen legen verstärkt die Priorität auf kleine Iterationszyklen, selbständige Teams und regelmäßige inkrementelle Lieferungen an den Kunden. Es wird verstärkt versucht, frühzeitig das Kundenfeedback einzuholen und auch einzuarbeiten. Durch das Wegfallen traditioneller Methoden der Anforderungserhebung und Vertragsgestaltung wie Lasten- und Pflichtenheften ist eine der wichtigen Fragen, wie Produktivität, Leistungsumfang, Effizienz und Effektivität von agilen Teams und der agilen Softwareentwicklungsprozesse valide gemessen werden kann.

1.1 Problemstellung

Firmen benötigen die Möglichkeit für die valide Bewertung der Leistung Ihrer Teams, daher ist es für KMUs mit beschränkten personellen und finanziellen Ressourcen notwendig jene Bewertungskriterien als erster einzuführen die die höchste Aussagekraft haben und größten Mehrwert liefern bei der Bewertung der Leistung von agilen Teams, ihrer internen agilen Softwareprozesse sowie um eine korrekte Balance zwischen Kosten und Umsatz zu finden. Die Einführung von Metriken soll dabei auch den Grundstein für die Integration von Continuous Integration und Deployment Praktiken bieten.

1.2 Zielsetzung

Aufbauend auf der in Bachelorarbeit 1 stattgefundenen Literaturrecherche und dem extrahierten Wissen, soll ein gewichteter Kriterienkatalog erstellt werden. Es sollen die verschiedenen Metriken und deren Nennungshäufigkeit in der Literatur eruiert werden, im Anschluss dazu werden ExpertInnen aus der Praxis der Softwareentwicklung sowie mit Lektoren Tätigkeit im Hochschulwesen herangezogen. Der so zweistufig validierte gewichtete Kriterienkatalog wird dann auf die primären Metriken reduziert und zuvor entwickelter Dimension zugeordnet.

1.3 Forschungsfrage

Für die Erarbeitung des Kriterienkataloges sowie die Durchführung der ExpertInneninterviews, wurde folgende Forschungsfrage entwickelt:
Welche Kriterien sind für KMUs im Bereich Dienstleistungen in der automatischen Datenverarbeitung und Informationstechnik, für die Bewertung der Leistung agiler Teams entscheidend und sollten als erste eingeführt werden?

1.4 Vorgehen und Methode

Für die Grundlagen, Begriffsdefinition sowie die Erstellung der ersten Version des gewichteten Kriterienkataloges, wird eine Literaturrecherche durchgeführt. Die primäre Forschungsmethode dieser Bachelorarbeit, ist das ExpertenInneninterview, dieses zählt zu den qualitativen Forschungsmethoden. Anhand eines zuvor entwickelten Leitfadens werden Expertinnen und Experten in der Softwareentwicklung zu Metriken in der agilen Softwareentwicklung befragt. Genauere Informationen zur verwendeten Methodik sind in Kapitel 3 zu finden.

2 GRUNDLAGE PRODUKTIVITÄT UND AGILE SOFTWARE ENTWICKLUNG

Ausgehend von der Bachelorarbeit 1, „**SOFTWARE METRIKEN UND MECHANISMEN**

zu Bewertung agiler Teams“ sollen nur die für diesen Arbeit essenziellen Teile kurz pointiert dargestellt werden. (Harald Beier, 2021)

2.1 Generelle Definition Produktivität

Produktivität wird in unterschiedlichen Fachgebieten und Wissenschaftszweigen untersucht und definiert. Es gibt verschiedene Abstraktionsebenen und für die Softwareentwicklung fehlt eine einheitliche Definition. (Harald Beier, 2021; Sadowski & Zimmermann, 2019)

Ausgehend von der bereits in Bachelorarbeit 1 (Harald Beier, 2021) stattgefundenen genauen Darstellung und Analyse der einzelnen Parameter innerhalb der wissenschaftlichen Literatur, erfolgt nur eine kurze Zusammenfassung in Tabelle 1 der wichtigsten Begriff und in welcher Literatur die Definition zu finden ist und welche Formeln bzw. Definition für diese Arbeit gültig ist.

Faktor	Zusammenfassung/Formel/Definition	Literatur
Produktivität	$Produktivität = \frac{Output}{Input}$	(Chew, 1988; Sadowski & Zimmermann, 2019)
Rentabilität	$Rentabilität = \frac{Umsatz}{Kosten}$	(Sadowski & Zimmermann, 2019)
Effizienz und Effektivität	<div>$Effizienz = \frac{ideale\ systemabhängige\ Zeit}{Gesamtzeit}$$Effektivität = \frac{Wertschöpfungszeit}{ideale\ systemabhängige\ Zeit}$</div> <div><div><div>Effizienz = $\frac{Erwarteter\ Ressourcenverbrauch}{Rechtzeitig\ verbrauchte\ Ressourcen}$</div><div>Effektivität = $\frac{Tatsächlicher\ Output}{Erwarteter\ Output}$</div></div><div><div>Vorgelagertes System</div><div>Input</div><div>Transformations-prozess</div><div>Ausgang</div><div>Nachgelagertes System</div></div></div>	(Jackson, 2000; Sink & Tuttle, 1995; Tangen, 2005)

Abbildung 1: Prozessmodell mit Effizienz- und Effektivität-Definition und Einordnung basierend auf (Sink & Tuttle, 1995) (Harald Beier, 2021)

Tabelle 1: Zusammenfassung der einzelnen Parameter die unter Produktivität in der Literatur vorhanden sind, basieren auf (Harald Beier, 2021) ©Harald Beier

2.2 Produktivität in der Softwareentwicklung

Die Definition von Leistung in der Softwareentwicklung hat keine klare Definition. Ein Beispielsmodell wird von Wagner und Deissenboeck entwickelt, das bereits in Bachelorarbeit 1 präsentiert wurde, für das bessere Verständnis dient die Abbildung 2. (Sadowski & Zimmermann, 2019). Für nähere Informationen wird auf die Bachelorarbeit 1 verwiesen (Harald Beier, 2021)

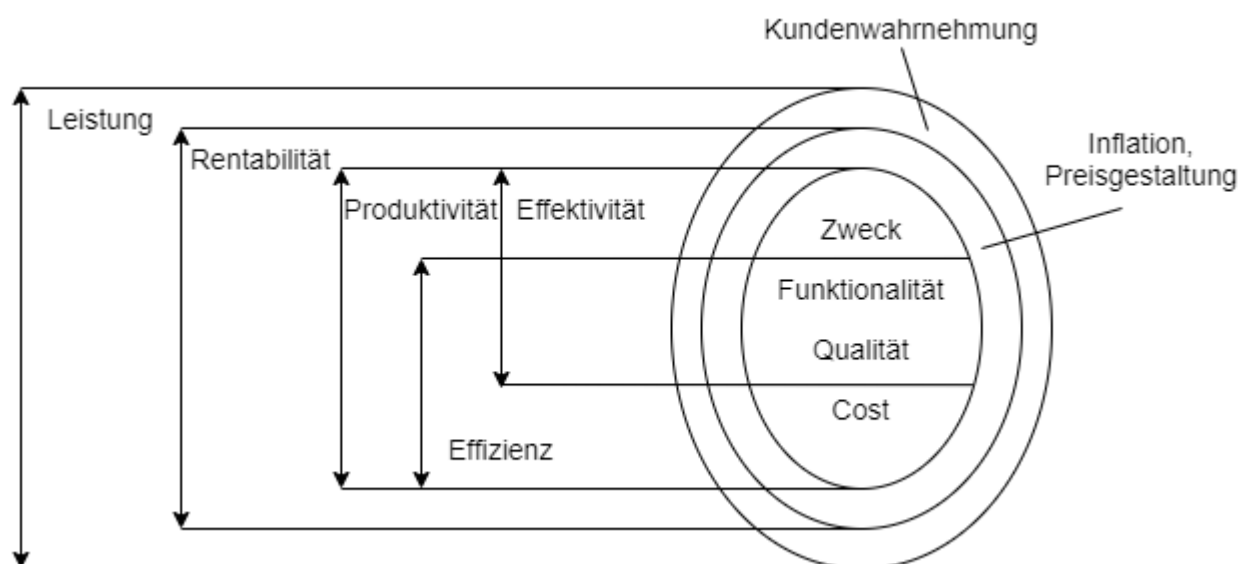


Abbildung 2: Leistungsmessungs-Modell basierend auf (Sadowski & Zimmermann, 2019) ©Harald Beier (Harald Beier, 2021)

Da die Forschungsfrage und deren Beantwortung einen Mehrwert für österreichische KMUs liefern soll, ist es notwendig eine Verkürzung vorzunehmen und die beschriebenen Dimensionen zu reduzieren. Es wird daher wie bereits in Bachelorarbeit 1 auf die vorgeschlagenen Leistungsdimensionen Produktqualität, Teamproduktivität und Vorhersagbarkeit des Artikel **Use of Software Metric in Agile Software Development Process** zurückgegriffen (Harald Beier, 2021; Padmini et al., 2015)

Für die Gültigkeit dieser Verkürzung auf der Dimensionsebenen werden zwei zusätzliche wissenschaftliche Arbeiten und deren Dimensionen sowie etwaige Indikatoren Metriken in Relation der vorab definierten drei Parameter gesetzt.

Dimensionen nach (Padmini et al., 2015)	Dimensionen und Metriken die zugeordnete werden können
Produktqualität	<ul style="list-style-type: none"> Produktion – Vermeidung von Prozessabfällen(Ertaban et al., 2018) Qualität – Anzahl von Mängeln Fehlern (Ertaban et al., 2018) Qualität – Fehlerdichte (Ertaban et al., 2018)

	<ul style="list-style-type: none"> ▪ Kundenzufriedenheit – Anzahl zufriedener Kunden im Verhältnis zu Gesamtanzahl an Kunden (Ertaban et al., 2018) ▪ Effizienz (Shah et al., 2015) ▪ Qualität (Shah et al., 2015) ▪ Effektivität (Shah et al., 2015)
Teamproduktivität	<ul style="list-style-type: none"> ▪ Produktion – Velocity (Ertaban et al., 2018) ▪ Produktion – Vermeidung von Prozessabfällen(Ertaban et al., 2018) ▪ Umfang (Shah et al., 2015) ▪ Autonomie (Shah et al., 2015) ▪ Effizienz (Shah et al., 2015) ▪ Projekterfolg (Shah et al., 2015) ▪ Kundenzufriedenheit (Shah et al., 2015)
Vorhersagbarkeit	<ul style="list-style-type: none"> ▪ Produktion – Durchlaufzeit und Zykluszeit ▪ Kosten (Shah et al., 2015) ▪ Zeitdisziplin (Shah et al., 2015) ▪ Projekterfolg (Shah et al., 2015)

Tabelle 2: Verkürzung der Leistungsdimensionen auf drei Parameter nach (Padmini et al., 2015) und Zuordnung anderer Dimensionsebenen ©Harald Beier

2.3 Definition Agilität und agile Softwareentwicklung

Agilität und agile Softwareentwicklung basieren auf dem Manifest für Agile Softwareentwicklung das im Februar 2001 von 17 Softwareentwicklern in einer Skihütte in Utah entwickelt wurde. Es war ein Gegenentwurf zu den traditionellen Methoden der Softwareentwicklung wie der Wasserfallmethode oder anderer Ideen.



Abbildung 3: Agile Leitsätze mit Icons von [Freepik](#) und [Eucalyp](#) von [www.flaticon.com](#), basierend auf (Kent et al., 2001) ©Harald Beier

Hauptgrundlage waren die in Abbildung 3 vorliegenden Agilen Leitsätze und die in Abbildung 4 aufgezeigten Prinzipien.



Abbildung 4: Agile Prinzipien mit Icons von [Freepik](#) von [www.flaticon.com](#), basierend auf (Kent et al., 2001) ©Harald Beier

Diese bilden die Grundlage für eine Vielzahl an unterschiedlichen Frameworks und Praktiken, die sich auf Basis des Agilen Manifests entwickelt haben. Auf umfangreichere Informationen muss in dieser Arbeit verzichtet werden und es wird auf die Kapiteln 2.1 bis 2.3 in der Bachelorarbeit 1 **Software Metriken zu Bewertung agiler Teams** verwiesen (Harald Beier, 2021).

Seit dem Jahr 2001 in dem das agile Manifest entstanden ist haben zusätzliche Praktiken die Softwareentwicklung beeinflusst. Aktuell sind die kontinuierlichen Praktiken wie kontinuierliche Integration, Lieferung und Bereitstellung, die Praktiken der Softwareentwicklungsbranche, die es Unternehmen ermöglichen, neue Funktionen und Produkte häufig und zuverlässig zu veröffentlichen. (Gallaba, 2019 - 2019; Shahin et al., 2017)

Helena Holmström Olsson, Hiva Alahyari und Jan Bosch entwickelten 2012 eine Stufenmodell für Firmen um die „Stairway to Heaven“ zu erklimmen.

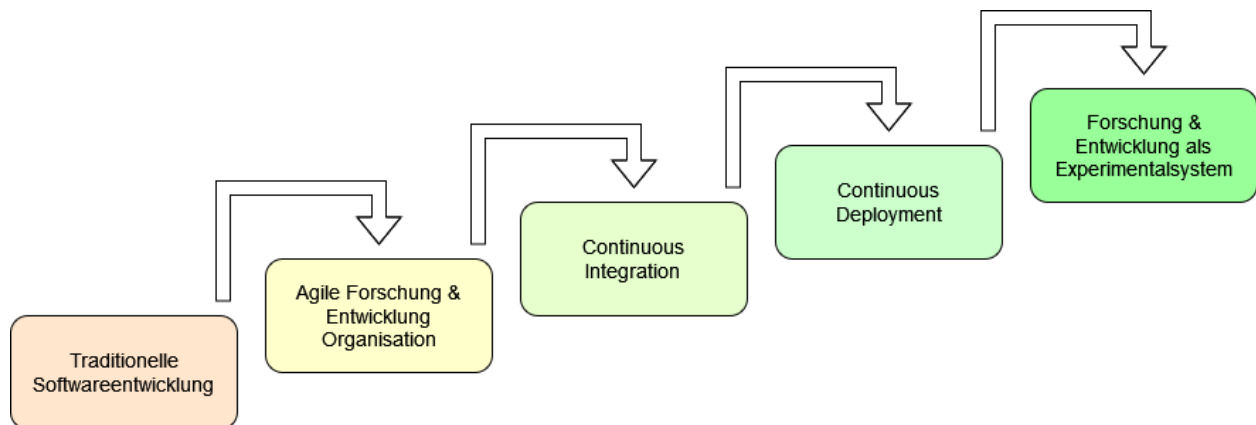


Abbildung 5: „The stairway to heaven“ basierend auf (Olsson & Bosch, 2014) ©Harald Beier

Die einzelnen Stufen werden durch Reifegrade innerhalb von vier Dimensionen charakterisiert: Produktmanagement(PM), Forschung & Entwicklung(F&E), Verifikation und Validierung(V&V) das die entwickelte Software die spezifizierten Anforderungen erfüllt sowie die Integration von Kundenfeedback(KF). Sie definieren dafür die Stufen:

- T für traditionelle Softwareentwicklungsmethodik
- A für agile Softwareentwicklung
- SZ für kurze Entwicklungs- bzw. Feedbackzyklen

Stufe	PM	F&E	V&V	KF
A -Traditionelle Softwareentwicklung	T	T	T	T
B – Agile Forschung & Entwicklungsorganisation	T	A	T	T
C – Continuous Integration	T	SC	SC	T
D – Continuous Deployment	A	SC	SC	A
E – Forschung & Entwicklung als Experimentalsystem	SC	SC	SC	SC

Tabelle 3: Zusammenfassung der Stufen der „Stairway to Heaven“ (Olsson & Bosch, 2014, S. 16) ins Deutsch übersetzt

Die kontinuierlichen Praktiken die notwendig sind für die Erklommung der höheren Stufen, sind bereits in der wissenschaftlichen Literatur ein stark diskutiertes Thema. (Fitzgerald & Stol, 2017) Es gibt bereits wissenschaftliche Papers, die sich mit verschiedenen Parametern und Umsetzungspraktiken auseinandersetzen. In der Arbeit **An empirical characterization of bad practices in continuous integration** definieren die Autoren verschiedene Praktiken für Softwarefirmen die Continuous Integration umsetzen, um erfolgreicher zu sein. Diese sind das Etablieren von auf die Firma angepasste Quality Gates im Entwicklungsprozess und Nutzung von Werkzeugen um diese spezifischen Quality Gates zu überprüfen sowie die regelmäßige Kontrolle der Build Logs und warum spezifische Builds bei automatisierten Test Warnungen verursachen oder es generell zu Failed Builds kommt. (Zampetti et al., 2020)

In der wissenschaftlichen Literatur wird bereits intensiv die Bedeutung der automatisierten Tests, die während des Continuous Integration genutzt werden, diskutiert. Eine der Herausforderung, die in der Arbeit ***An extensive study of class-level and method-level test case selection for continuous integration*** genauer untersucht wird, ist wie geeignete Tests auszuwählen sind, um optimale Ressourcennutzung ohne Beeinträchtigung der Qualität zu gewährleisten. (Yingling Li et al., 2020). Dieses Feld der Testpriorisierung erfährt aktuell ein steigendes Interesse in der Wissenschaft, und es gibt unterschiedliche Ansätze für die Priorisierung von Testfällen in Softwareentwicklungs-Umgebungen mit Continuous Integration. (Jackson A. Prado Lima & Silvia R. Vergilio, 2020)

Die Forschungsfrage dieser Arbeit konzentriert sich aber auf die zweite Stufe der „*Stairway to Heaven*“ und soll durch den entwickelten gewichteten Kriterienkatalog KMUs ein solides Fundament für das Voranschreiten in die nächsten Stufen liefern und somit nachhaltiges Wachstum und Entwicklung ermöglichen.

2.4 Performance Indikatoren und Metriken

Für die korrekte Einordnung bzw. die Entwicklung des gewichteten Kriterienkataloges im nächsten Kapitel 2.5 ist es notwendig vorab die bereits in der Literatur gelisteten und bearbeiteten Metriken zu überprüfen. Die Autoren Eetu Kupiainen, Mika V.Mäntylä und Juha Itkonen haben 2015 in Ihrem Paper ***Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies*** eine umfassende Literaturanalyse von über 774 wissenschaftlichen Arbeiten durchgeführt, die auch als Basis für diese Arbeit dient. (Kupiainen et al., 2015)

Nachfolgend wird das Ergebnis mit der Kategorisierung durch Fenton und Pfleger (Fenton & Bieman, 2015) tabellarisch dargestellt. Es wird dabei zwischen Entitäten und Attribute unterschieden. Entitäten sind laut Fenton und Pfleger „*eine Sammlung aller Software spezifischen Artefakte*“, Produkte die als Endergebnis eines Prozesses entstehen, wie zum Beispiel Dokumentation und Ressourcen, die für die Prozesse benötigt werden. Es erfolgt eine weitere Differenzierung in externe und interne Attribute abhängig davon, ob diese intern gemessen werden können oder nur mit Unterstützung der Außenwelt. Die Anzahl der wissenschaftlichen Arbeiten, in denen diese Metrik genannt wird, ist in Klammer dargestellt, damit deren Gewichtung für das anschließende Kapitel 2.5 nachvollzogen werden kann.

Entitäten	Attribute	
Produkte	Interne	Externe
Produkte	Getestete Features anbieten(1), Build Status(2)	Kundenzufriedenheit(6), Fortschritt durch gelieferten funktionierend Code(1)
Test Pläne	Nummer der Testfälle(1)	

Code	Technische Altlasten/Schulden in Kategorien (1), Technische Altlasten/Schulden im Aufwand (1) Verstöße gegen die statische Codeanalyse (1)	
Features	das voraussichtliche Enddatum der Arbeitspakete(1), erledigte Arbeitspakete(1), Aufwandsschätzung(8), Prozent der abgeschlossenen User Stories(1)	Gelieferter Business Value(1)
Requirements/Anforderungen	Kostenarten der Anforderung(1), Prozentsatz der für den Sprint vorbereiteten Stories(1)	
Defekte		Defekt-Trend-Indikator(1), voraussichtliche Anzahl der Defekte(1)
Prozesse	Interne	Externe
Testen	Fehlerzahl (8), Testerfolgsrate (1), Testfehlerrate(3), aufgeschobene Behebung von Fehler(1), Testabdeckung(1), Zuwachsrate der Test(1)	Anzahl der Regressionen eine Codeversion (1), Anzahl wie oft ein Fehler gemeldet wird (1)
Implementierung	Velocity (12), Anzahl der Unit Test (5), Fertiggestellte Webpages(1), Kosten-Leistungsindex(1), Zeitplan-Leistungs-Index(1), geplante Velocity(1), durchschnittliche Geschwindigkeit(1) Check-Ins/Commits pro Tag(3), Dauer bis ein fehlgeschlagener Build gefixt ist(1)	Flowcharts für die Umsetzung von Stories (1)
Requirements engineering / Anforderungserhebung	Geschwindigkeit der Ausarbeitung von Anforderungen und Features(1)	
Gesamter Software Entwicklungszklus	Zykluszeit (2), Durchlaufzeit(4), Bearbeitungszeit(1),	

	Wartezeit(1), Wartungsaufwand(1), Arbeitspakete im Backlog/Bearbeitung(7), Varianz bei Übergabe(1), Durchsatz(1), Größe der Queue der Arbeitspakete(1), umgesetzte vs. Verschwendete Anforderungen(1)	
Ressourcen	Interne	Externe
Team		Teameffizien(1)
Kunde	Umsatz pro Kunde(1)	

Tabelle 4: Metriken die in der Softwareentwicklung genutzt werden und der Nennungshäufigkeit in der wissenschaftlichen Literatur, basieren auf (Kupiainen et al., 2015)

2.5 Entwicklung des priorisierten Kriterienkataloges

Anhand der zuvor präsentierten Literatur und für die besser Lesbarkeit und Übersichtlichkeit werden in diesem Unterkapitel, die einzelnen Kriterien und Möglichkeiten für die Einführung von Metriken und Kennzahlen bei der Bewertung agiler Teams in einer Tabelle zusammengeführt. Es handelt sich dabei um einen Kriterienkatalog, der als Basis für den empirischen Teil dienen soll. Die geführten Experteninterviews werden mit den Ergebnissen der Literatur abgeglichen und der Katalog, um fehlende Metriken zu ergänzen.

2.5.1 Kriterien Bildung

Anhand der in Kapitel 2.2 getroffenen Verkürzung auf die Dimensionen Produktqualität, Teamproduktivität und Vorhersagbarkeit, werden die in Kapitel 2.4 angeführten Metriken, diesen Dimensionen zugeordnet. Die Zuordnung nach Entitäten und Attributen laut Fenton und Pfleger wird dafür aufgelöst, um die notwendige Verkürzung der Metriken in einem KMU Umfeld zu ermöglichen.

Dimension	Metrik Häufigkeit der Nennung
Produktqualität	Getestete Features anbieten (1), Buildstatus(2), Kundenzufriedenheit (6), Nummer der Testfälle (1), Technische Altlasten/Schulden in Kategorien (1), Technische Altlasten/Schulden im Aufwand (1), Verstöße gegen die statische Codeanalyse (1), Aufwandsschätzung (8), Defekt-Trend-Indikator (1), voraussichtliche Anzahl der Defekte (1), Fehlerzahl (8), Testerfolgsrate (1), Testfehlerrate (3), aufgeschobene Behebung von Fehler

	(1), Testabdeckung (1), Zuwachsrates der Tests (1), Anzahl der Regressionen in einer Codeversion (1), Anzahl wie oft ein Fehler gemeldet wird (1), Anzahl der Unit Tests (5), Geschwindigkeit der Ausarbeitung von Anforderungen und Features (1), Varianz bei Übergabe (1), Wartungsaufwand (1), Kostenarten der Anforderung (1)
Teamproduktivität	das voraussichtliche Enddatum der Arbeitspakete (1), erledigte Arbeitspakete (1), Prozentsatz der abgeschlossenen User Stories (1), Prozentsatz der für den Sprint vorbereiteten Stories (1), Velocity (12), Check-Ins/Commits pro Tag (3), Dauer, bis ein fehlgeschlagener Build gefixt ist (1), Teameffizienz (1), Bearbeitungszeit (1), Zykluszeit (2), Durchlaufzeit (4)
Vorhersagbarkeit	das voraussichtliche Enddatum der Arbeitspakete (1), Kosten-Leistungsindex (1), Zeitplan-Leistungs-Index (1), geplante Velocity (1), Zykluszeit (2), Durchlaufzeit (4), durchschnittliche Geschwindigkeit (1), Arbeitspakete im Backlog/ Bearbeitung(7), Durchsatz (1), Größe der Queue der Arbeitspakete(1)

Tabelle 5: Kriterienbildung für gewichteten Katalog basierend auf (Kupiainen et al., 2015) ©Harald Beier

Für den nächsten Schritt der Gewichtung, werden die Dimensionen anhand der zuordenbaren Metriken sowie der Gesamtanzahl der Arbeiten innerhalb dieser Dimension gewichtet. Die einzelnen Metriken werden anhand der Nennung innerhalb von Artikeln gewichtet. (Kupiainen et al., 2015)

2.5.2 Gewichtung

In der Sozialwissenschaft gibt es die Methode der Gewichtung, um Priorität und Relevanz der einzelnen Kriterien besser festzulegen. Es wird eine sehr einfache Priorisierungsskala angewandt, die sich an einem 1-2-3 Schema orientiert. Für die wichtigsten bzw. wirkungsvollsten Kennzahlen bzw. Metriken wird eine „1“, für wichtige Kennzahlen, die nicht die allerhöchste Notwendigkeit aufweisen eine „2“ und kleinere, weniger wichtige Anforderungen eine „3“ vergeben. (Gabler & Ganninger, 2010)

Damit diese nachvollziehbar ist, werden die Anzahl der Nennungen den Kategorien zugeordnet. 1-3 Nennungen sind eine Gewichtung von 3, 4-6 eine Gewichtung von 2 und 7- ∞ eine Gewichtung von 1. Ziel ist es außerdem, die Gewichtungen anhand der gewonnenen Ergebnisse der geführten ExpertInneninterviews anzupassen und zu ergänzen.

Dimension	Metrik	Nennung	Gewichtung
Produktqualität (23 Metriken und 49 Nennungen)	Aufwandsschätzung (8)	8	1
	Fehlerzahl (8)	8	1
	Kundenzufriedenheit (6)	6	2
	Anzahl der Unit Test (5)	5	2
	Testfehlerrate (3)	3	3
	Buildstatus(2)	2	3
	Getestete Features anbieten (1)	1	3
	Nummer der Testfälle (1)	1	3
	Technische Altlasten/Schulden in Kategorien (1)	1	3
	Technische Altlasten/Schulden im Aufwand (1)	1	3
	Verstöße gegen die statische Codeanalyse (1)	1	3
	Defekt-Trend-Indikator (1)	1	3
	voraussichtliche Anzahl der Defekte (1)	1	3
	Testerfolgsrate (1)	1	3
	aufgeschobene Behebung von Fehler (1)	1	3
	Testabdeckung (1)	1	3
	Zuwachsrate der Test (1)	1	3
	Anzahl der Regressionen in einer Codeversion (1)	1	3
	Anzahl wie oft ein Fehler gemeldet wird (1)	1	3
	Geschwindigkeit der Ausarbeitung von Anforderungen und Features (1)	1	3
	Varianz bei Übergabe (1)	1	3
	Wartungsaufwand (1)	1	3
	Kostenarten der Anforderung (1)	1	3
Teamproduktivität (11 Metriken, 28 Nennungen)	Velocity (12)	12	1
	Durchlaufzeit (4)	4	2
	Check-Ins/Commits pro Tag (3)	3	3
	Zykluszeit (2)	2	3

	das voraussichtliche Enddatum der Arbeitspakete (1)	1	31
	erledigte Arbeitspakete (1)	1	3
	Prozent der abgeschlossenen User Stories (1)	1	3
	Prozentsatz der für den Sprint vorbereiteten Stories (1)	1	3
	Dauer, bis ein fehlgeschlagener Build gefixt ist (1)	1	3
	Teamffizienz (1)	1	3
	Bearbeitungszeit (1),	1	3
Vorhersagbarkeit (10 Metriken, 20 Nennungen)	Arbeitspakete im Backlog/ Bearbeitung (7)	7	1
	Durchlaufzeit (4)	4	2
	Zykluszeit (2)	2	3
	das voraussichtliche Enddatum bei Lieferung der Arbeitspakete (1)	1	3
	Kosten-Leistungsindex (1)	1	3
	Zeitplan-Leistungs-Index (1)	1	3
	geplante Velocity (1)	1	3
	durchschnittliche Geschwindigkeit (1)	1	3
	Durchsatz (1)	1	3
	Größe der Queue der Arbeitspakete(1)	1	3

Tabelle 6: Gewichteter Kriterienkatalog mit zugeordneten Metriken basierend auf (Kupiainen et al., 2015) ©Harald Beier

Anhand der Nennungen in wissenschaftliches Paper sowie der Anzahl der zugeordneten Metriken, ist zu erkennen, dass es in der fachwissenschaftlichen Literatur eine Rangordnung anhand des Interesses gibt:

- Produktqualität mit 23 zugeordneten Metriken und 49 Nennungen in wissenschaftlichen Arbeiten
- Teamproduktivität mit 11 zugeordneten Metriken und 28 Nennungen in wissenschaftlichen Arbeiten
- Vorhersagbarkeit mit 10 zugeordneten Metriken und 20 Nennungen in wissenschaftlichen Arbeiten

Es ist bereits ein Trend betreffend des Priorisierungsschritte bei der Einführung von Metriken zu erkennen, dieser muss mit den ExpertInneninterviews und deren Auswertung in Kapitel 4 abgeglichen werden.

3 METHODIK

Ziel des Kapitels ist es, die notwendige methodische Grundlage für den darauffolgenden empirischen Teil der Arbeit zu entwickeln und zu beschreiben. Dies ist notwendig um die Forschungsfrage theoriegestützt anhand der wissenschaftlichen Literatur und Methodik zu beantworten. Damit die in Kapitel 1.3 gestellte Forschungsfrage beantwortet werden kann, werden Experteninterviews in Kombination mit der qualitativen Inhaltsanalyse nach Mayring genutzt. Dieses Kapitel liefert sowohl die notwendigen Informationen zum Aufbau, Durchführung der Interviews und die Begründung für die gewählte Methodik.

3.1 Experteninterview

Das Thema „Software Metriken“ und die Beantwortung der Forschungsfrage erfordert Branchenkenntnisse in der Softwareentwicklung sowie Wissen in der Mitarbeiterführung und über Agilität im Bereich der Softwareentwicklung. Nur ein begrenzter Personenkreis kann daher für Interviews herangezogen werden. Durch diese Einschränkung, wird als Forschungsmethode das Experteninterview für diese Bachelorarbeit angewandt. Das Experteninterview ist eine qualitative Forschungsmethode und wird mithilfe eines zuvor erstellten Leitfadens durchgeführt. (Bogner et al., 2014). Durch Experteninterviews lässt sich fokussiert praxisrelevante Information gewinnen.

3.1.1 Auswahl der Expertinnen und Experten

Anhand der Forschungsfrage werden für die Interviews Personen ausgewählt, die innerhalb von Unternehmen arbeiten im Bereich Dienstleistungen in der automatischen Datenverarbeitung und Informationstechnik sowie direkt involviert sind in die Führung von agilen Softwareentwicklungsteams. Es wird versucht unterschiedliche Firmengrößen abzudecken und im besten Fall bereits über das bestehende professionelle Netzwerk, Kontakt zu den ausgewählten Interviewpartnern aufzubauen. Die Expertinnen und Experten haben durch ihre unterschiedlichen Rollen in den Unternehmen jeweils verschiedene Sichtweisen auf das Thema und können somit Informationen vom technischen Aspekt, aus Management Sicht sowie aus Kundensicht bzw. Consulting Sicht wiedergeben. Es konnten insgesamt vier Expertinnen und Experten aus vier verschiedenen Unternehmen gefunden werden, die sich für ein Interview bereit erklärt haben. Angefragt wurden 15 Personen in 15 unterschiedlichen Firmen, wobei leider bei KMUs oft die Personen aufgrund von Zeitmangel die Interviews absagen mussten. Aus Gründen des Datenschutzes wurden die Personen sowie die Unternehmen, für die Sie arbeiten pseudonymisiert. Eine Übersicht sowie eine Kurzbeschreibung über die interviewten Personen, erfolgt in der nachfolgenden Tabelle. Im Ergebniskapitel wird auf die Kodierung dieser Tabelle zurückgegriffen, wenn die Aussagen einer speziellen Person zugeordnet werden.

Personen	Position/Sicht	Beschreibung	Unternehmen
----------	----------------	--------------	-------------

P1	CEO/CTO innerhalb einer SaaS Firma	Bewertung der Metriken der agilen Teams als CTO. Einbettung dieser Daten in die Firmenstrategie als CEO	50 – 100 Mitarbeiter,
P2	CEO und Consulting für Firmen, Lektor Hochschule	Consulting bei anderen Firmen, Lektor an Hochschule, Implementierung innerhalb der eigenen Firma	10 – 50 Mitarbeiter,
P3	Lead Softwaredesigner von 2 Teams, Lektor Hochschule	Zuständig für 2 Entwicklungsteams, ist involviert in der Kooperation zwischen Management und Softwareentwicklung. Lektor für Software-Entwicklung.	50-100 Mitarbeiter,
P4	Teamleiter Implementierung und Adaptierung der Standardsoftware	Lange Implementierungsphasen, große Projekte mit hohen Umsätzen, Im Changeprozess von traditionellen Projektmanagement und Waterfall Delivery System zu Agilität	100 -249 Mitarbeiter,

Tabelle 7: Aufschlüsselung der Interviewpartner ©Harald Beier

3.1.2 Durchführung des Interviews

Angesetzt waren für die Interviews als durchschnittlicher Zeitraum 30 Minuten. Abhängig von den jeweiligen Interviewpartner wurden die Dauer situativ adaptiert. Es gibt daher auch deutlich längere Interview entsprechend der Antworten des Gegenübers. Die Interviews wurden persönlich sowie über die Plattform Microsoft Teams durchgeführt und mit der Software Audacity aufgezeichnet. Anschließend erfolgte das manuelle Transkribieren der Aufzeichnungen mit der Software Maxdaq. Der Interviewleitfaden, welcher im nachfolgenden Kapitel beschrieben wird, wurde vor dem Interview an die Expertinnen und Experten verschickt, damit die Interviewpartner die Möglichkeit hatten bekannt zu geben, ob diese Informationen vorhanden sind und auch beantwortet werden können. Damit die Expertinnen und Experten sich nicht vorab speziell für das Interview Wissen erarbeiten und vorgefertigte Antworten liefern, werden die Fragen entsprechend des Gesprächsverlaufes adaptiert um spontane Aussagen zu ermöglichen. (Bogner et al., 2014; Wassermann, 2014)

3.1.3 Aufbau des Interviews

Damit die durchgeführten Interviews Theorie gestützt durchgeführt werden können musste ein Leitfaden entwickelt werden. Um sicherzustellen, dass sowohl die forschungsrelevanten Themen besprochen werden und es einen roten Faden sowie grobe Struktur gibt, wurde entsprechend

der Literaturempfehlungen dieser Leitfaden als Hilfswerkzeug für den Interviewer vorab erstellt. (Bogner et al., 2014; Helfferich, 2014; Niederberger & Wassermann, 2014)

Der Leitfaden wurde entsprechend der Antworten und der jeweiligen Expertise und Erfahrungen der Interviewpartner situativ adaptiert. Die Reihenfolge der thematischen Bereiche wurde versucht so gut wie möglich einzuhalten, um eine größere Vergleichbarkeit der Antworten sicherzustellen. Essenz des Leitfadens und auch des Interviews selbst ist es aber einen hohen Grad an destillierten Informationen für die Forschungsfrage zu gewinnen. Der ausgearbeitete Leitfaden, ist in Anhang A einsehbar. Entsprechend der in der Literatur üblichen Aufteilung der Experten Interviews in strukturierte, semistrukturierte und unstrukturierte Interviews, handelt es sich bei der für diese Arbeit angewandten Interviewform um ein qualitatives semistrukturiertes Interview. (Bogner et al., 2014; Helfferich, 2014; Niederberger & Wassermann, 2014). Die verschiedenen Inhaltlichen Fragebereiche des Leitfadens werden nun näher beschrieben.

- **Interview Einleitung:** Der Aufbau sowie die Strukturierung wurden basierend auf der Arbeit von an Bogner et al. (2014) erstellt. Als Startpunkt des Interviews dient die Einleitung, hier kommt es zu Vorstellung der eigenen Person als Interviewer. Es folgt nun die Beschreibung des Ablaufes für die interviewte Person sowie die nochmalige Information wie mit den Daten im Zuge der Bachelorarbeit verfahren wird. Zudem haben alle Interviewpartner eine Einverständniserklärung entsprechend der Vorlage der Fachhochschule Campus02 erhalten und bereits vorab unterzeichnet. Vor Beginn der Audio-Aufzeichnung wird nochmal um ein Einverständnis für die Aufzeichnung und der Datenverarbeitung gebeten und das bei Videokonferenz per MS-Teams nur die Tonspur aufgezeichnet wird.
- **Thematische Einleitung:** In diesem Abschnitt geht es darum dem Interviewpartner einen Einblick in die Thematik der Bachelorarbeit zu geben. Es werden sowohl die Forschungsfrage sowie das Ziel der Bachelorarbeit vorgestellt. Zusätzlich werden die Bereiche präsentiert die genauer untersucht werden und welche Ergebnisse durch die Vorarbeit erwartet werden.
- **Fragen zur Person / Berufliche Tätigkeit:** In diesem Abschnitt erfolgt die Einstiegsfrage in das Interview. Die Einstiegsfrage dient in erster Linie dazu, dass sich die befragte Person in die Interviewsituation versetzen kann (Bogner et al., 2014)
- **Agile Softwareentwicklung:** In diesem Themenbereich wird die generelle Einstellung bzw. Vorwissen des Interviewpartner zu agilen Softwareentwicklung und agile Teams abgefragt. Danach wird die Vorerfahrung betreffend Metriken zu Bewertung von agilen Softwareentwicklungsteam abgefragt. Am Schluss werden die Kennzahlen die bereits vorhanden und gemessen sind in der Firma erfragt.
- **Produktqualität:** In diesem Abschnitt des Interviews wird entsprechend der Definition und basierend auf (Padmini et al., 2015) der Augenmerk auf die Erfahrungen, Meinungen und gelebte Praxis betreffend Produktqualität sowie der Möglichkeit diese Produktqualität mit Metriken zu messen gelegt.

- **Teamproduktivität:** Auf die Gliederung von (Padmini et al., 2015) basierend sowie den Befragungsergebnissen (Meyer et al., 2014) wird in diesem Bereich versucht die Standpunkte der Interviewpartner zur Thematik Teamproduktivität und dazu passenden Kennzahlen abzufragen.
- **Vorhersagbarkeit:** Im Rahmen dies Abschnittes geht es darum die Voreinstellungen und Erfahrungen der Interviewpartner zu Metriken betreffend der Zeitdisziplin bei der Lieferung von Feature bzw. Softwareartefakten abzufragen.
- **Abschluss:** Nachdem alle Fragen des Leitfadens behandelt und das Interview abgeschlossen wurde, wird die Aufnahme gestoppt. Der Expertin oder dem Experten wird für die Mithilfe gedankt und ein Feedback über die Durchführung des Interviews wird eingeholt.

3.2 Qualitative Inhaltsanalyse nach Mayring

Für die Auswertung der Interviewgespräche wird die qualitative Inhaltsanalyse gewählt und verwendet. Die Methodik wird entsprechend der Arbeiten von Mayring und Kuckartz umgesetzt. (Kuckartz, 2018; Mayring, 2015) Innerhalb dieser Methodik findet eine systematische Analyse der verschiedenen Interviews statt um Überschneidungen in den Inhalten zu identifizieren und damit eine Vergleichbarkeit zu schaffen. Mayring definiert 3 Grundformen der qualitativen Inhaltsanalyse(Kuckartz, 2018; Mayring, 2015):

- **Zusammenfassende Inhaltsanalyse:** Ziel ist es einen Kurztext zu produzieren, der das untersuchte Material auf die Kernaussagen reduziert. Es ist die inhaltliche Ebene des Materials im Fokus.
- **Explizierende Inhaltsanalyse:** Im Gegensatz zur Zusammenfassenden Inhaltanalyse, werden unklare Textstellen, mit zusätzlichem Material bzw. Literatur ergänzt.
- **Strukturierende Inhaltanalyse:** Vorrangige Zielsetzung dieser Form, der Inhaltsanalyse, ist es einen Kodierleitfaden zu erstellen, um das Material anhand von vorher festgelegten Kategorien analysieren zu können.

Für die Analyse der Transkripte wird die strukturierende Inhaltsanalyse angewandt. Die strukturierende qualitative Inhaltsanalyse nach Mayring wird in mehreren Schritten durchgeführt. Die Durchführung aller Schritte wird im folgenden Kapitel beschrieben. (Kuckartz, 2018; Mayring, 2015)

3.2.1 Deduktive Kategorienbildung

Mayring definiert als Arbeitsschritt für die Strukturierende Inhaltanalyse, dass das Analysematerial in Kategorien zusammengefasst werden muss. Durch Deduktion im Sinne der Schlussfolgerung anhand von Prämissen, werden die Kategorien ermittelt. Durch die zuvor entwickelten Parameter in den Theoriekapiteln 2.4 und 2.5 und aufbauend auf der Bachelorarbeit 1 (Harald Beier, 2021) wurden die Kategorien erstellt und festgelegt bevor die Untersuchung des

zu analysierenden Materialen erfolgte. Zwischen Kategoriensystem und Interviewleitfaden gibt es eine große Schnittmenge. Es wurden mehr Subkategorien gebildet, um ein Clustering entsprechend der Literaturanalyse durchführen zu können:

- **K1:** Agile Softwareentwicklung
 - **K1-1:** Begriffsverständnis Agile Softwareentwicklung und agile Teams
 - **K1-2:** Metriken innerhalb der agilen Softwareentwicklung
 - **K1-3:** Priorität bzw. Präferenz bei Kennzahlen
- **K2:** Produktqualität
 - **K2-1:** Begriffsverständnis von Produktqualität in der agilen Softwareentwicklung
 - **K2-2:** Bekannte Metriken und Eignungsgrad
 - **K2-3:** Einsatz von Metriken innerhalb der Firma
- **K3:** Teamproduktivität
 - **K3-1:** Begriffsverständnis von Teamproduktivität in der agilen Softwareentwicklung
 - **K3-2:** Bekannte Metriken und Eignungsgrad
 - **K3-3:** Einsatz von Metriken innerhalb der Firma
- **K4:** Vorhersagbarkeit
 - **K4-1:** Begriffsverständnis von Vorhersagbarkeit der Liefermenge und Lieferdisziplin in der agilen Softwareentwicklung
 - **K4-2:** Bekannte Metriken und Eignungsgrad
 - **K4-3:** Einsatz von Metriken innerhalb der Firma

K1 dient der Feststellung des Vorwissens und der Kontextualisierung der Thematik, damit eine Vergleichbarkeit mit anderen Forschungsarbeiten zu ähnlichen Forschungsfragen gegeben ist. K2 bis K4 sind für die Gewichtung und Bildung des Kriterienkataloges entscheidend und um damit schlussendlich die Forschungsfrage beantworten zu können.

3.2.2 Erstellung des Datenmaterials

Das vorliegende Datenmaterial, dessen Entstehungshistorie sowie die damit verbundenen formalen Anforderungen und Charakteristika werden im nächsten Schritt näher erläutert. Die Interviews wurden sowohl persönlich als auch über die Software Microsoft Teams durchgeführt. Da die Lizenz des Campus02 keine Aufzeichnungen der Sitzungen innerhalb von Microsoft Teams erlaubt wurde auf die Open Source Software Audacity und Rückkoppelung über das Headset zurückgegriffen, um die Interviews online aufzuzeichnen. Bei persönlichen Interviews wurde die Aufzeichnung mit dem integrierten Rekorderapp durchgeführt. Aufgrund des vorliegenden Datenmaterials in MP3 bzw. als Audiodateien war es notwendig für die danach

folgende Kodierung und Paraphrasierung ein schriftliches Transkript zu erstellen. Es wurde eine sogenannte einfache Transkription durchgeführt, daher es wurden keine grammatikalischen Fehler korrigiert und es erfolgt nur bei starkem Dialekt bzw. Umgangssprache eine leichte Glättung des gesagten in die Standarddeutsche Varietät. Wortabbrüche, Stotterer, Füll- und Zögerungsäußerungen wurden nicht ins Transkript übernommen.

3.2.3 Kodierung

Grundlage für die notwendigen Schritte der Kodierung, der vorhandenen Transkripte bildet ein Kodierleitfaden der vor der Durchführung der Interviews erstellt wurde, damit eine größtmögliche Objektivität und es zu keiner präjudiziellen Vorwegnahmen bzw. Interpretation des gesagten kommen kann. Für den korrekten Einsatz als Analysewerkzeug enthält der Kodierleitfaden die Definition der einzelnen Kategorien, die durch Ankerbeispiele und Kodierregeln zusätzlich ausdifferenziert und ergänzt werden. Erst im Zuge des Transkribierens sowie der Kodierung der entstandenen Transkriptartefakte war es möglich die Ankerbeispiele zu ergänzen. Im Rahmen des Kodierprozesses, wurden die Kategorien mit den Beispielen sowie den Regeln genutzt, um die getroffenen Aussagen der Interviewpartner einzuordnen. Durch die erstellten Kategorien können etwaige anschließende Forschungsarbeiten leichter mit dem vorhandenen Datenmaterial arbeiten und die innerhalb getroffener Bewertungen und Einordnungen kritische untersuchen und überprüfen. (Kuckartz, 2018; Mayring, 2015)

Bei der Kodierung sowie der Kategorieabbildung innerhalb der einzelnen Transkripte wurde auf die Software MAXDAQ und die Arbeit von Rädiker und Kuckartz zurückgegriffen. (Rädiker & Kuckartz, 2019)

4 AUSWERTUNG DER EXPERTENINNENINTERVIEWS

In den nachfolgenden Unterkapiteln werden die Ergebnisse der Experteninterview, der daraus entstandenen Transkripte und der darauf aufbauenden qualitativen Inhaltsanalyse präsentiert. Die Kategorie K1- Agile Softwareentwicklung, die in Kapitel 4.1 untersucht und bearbeitet wird, war notwendig, um sicherzustellen dass es ein gleiches gemeinsames Verständnis von Agilität gibt und damit alle Interviewpartner die gleiche oder eine hohe Schnittmenge in der Interpretation des Begriffes aufweisen. Die Kapiteln 4.2 Produktqualität, 4.3 Teamproduktivität und 4.4 Vorhersagbarkeit sind essenziell für die Beantwortung der Forschungsfrage. Anhand der Paraphrasen und Generalisierungen wird der bereits in Kapitel 2.5 entwickelte gewichtete Kriterienkatalog um zusätzliche Informationen erweitert, ergänzt bzw. die Gewichtung neu interpretiert.

4.1 Agile Softwareentwicklung

Nachfolgend wird in Tabelle 3, die Generalisierungen der bereits paraphrasierten Aussagen der Interviewpartner, die den Kategorien K1-1 bis K1-3 zugeordnet werden konnten dargestellt. Dies dient der leichteren Übersichtlichkeit und fungiert zudem als Einleitung für die anschließende Auswertung der Aussagen. Für die detaillierten Paraphrasen sei in diesem Fall auf Anhang C – Qualitative Inhaltsanalyse verwiesen, dort sind sowohl die Personenkürzel als auch die gesamten paraphrasierten Textstellen zu finden.

Kategorie	Generalisierungen
K1-1 Begriffsverständnis Agile SWE und agile Teams	<p>Agilität ist selbständige Teams, kurze Arbeitszyklen und zeitnahes Feedback</p> <p>Für die agile Softwareentwicklung gibt es wenige relevante Metriken und diese sind immer relativ niemals absolut zu sehen.</p> <p>Agile Softwareentwicklung orientiert sich an Prinzipien und Werten im Rahmen der Selbstorganisation, Selbststeuerung und des Selbstmanagement der agilen Teams.</p> <p>Im Vordergrund der agilen SWE steht Eigenverantwortung der Teams, iteratives Vorgehen und Kundenfeedback.</p> <p>Geringer Bezug zu Agilität, stark Wasserfall orientiert. Continuous Integration und Continuous Deployment als QA Methoden</p> <p>Agile SWE = selbständige Organisationseinheiten setzen in iterativen Prozessen mit Kundenorientierung in Eigenverantwortung um</p>
K1-2 Metriken innerhalb der agilen SWE	<p>Kennzahlen in der Softwareentwicklung schwieriger als in anderen Teams zu erfassen.</p> <p>Produktqualitätsmetrik sind Bugs pro Release (Quantität und Schweregrad), Entwicklungsgeschwindigkeit beachten.</p> <p>KPIs müssen situativ angepasst werden und dürfen nicht Alleinzweck werden.</p> <p>Wichtigsten KPI sind die Velocity und die Story Points, sowie das Verhältnis zugesagten und gelieferten Story Points.</p>

	<p>Im Scrum Framework geht es um Story Point und wieviel Issues im Backlog sind.</p> <p>Kein durchgängiges agiles Framework. Starker interner Fokus auf Pflichtenheft</p> <p>Agilität im Projektbacklog und Projektmanagement. Burndowncharts zu Visualisierung</p>
<p>K1-3</p> <p>Priorität/Präferenz bei Kennzahlen</p>	<p>Wichtigste KPI intern Velocity und Story Points. Gefahr der wohlwollenden Schätzung</p> <p>KPIs dienen der Transparenz des Outputs nicht der Überwachung. Einordnung und Unterstützung der Unternehmensziele durch KPIs</p> <p>Intern wird nach Kanban gearbeitet, hier ist die Durchlaufzeit am wichtigsten.</p> <p>Intern werden die Kanban-Spalten dem Branching und Release Modells angepasst</p> <p>Die Durchlaufzeit beginnt dann zu laufen, wenn ein Issue Ready ist und endet, wenn durch den Continuous Integration Prozess das Issue beim Kunden ist.</p> <p>Kennzahlen dürfen nicht nur die Softwarequalität sein, sondern auch Time-to-Market</p> <p>Testcoverage ist abhängig von der Kritikalität des Bereiches. Viel Eigenverantwortung beim Team</p> <p>Durchsatz als Kerngröße = Arbeitspakete pro Sprint. Erfassungen Störungen je Modul bei Auslieferung</p>

Tabelle 8: Generalisierung der Paraphrasen der Kategorie K1 – Agile Softwareentwicklung ©Harald Beier

4.1.1 Begriffsverständnis agile Softwareentwicklung und agile Teams

Als Grundlage für die darauffolgenden Kapitel ist es notwendig, festzustellen, ob alle Interviewpartner ein einigermaßen deckungsgleiches Verständnis von agiler Softwareentwicklung und agilen Teams haben, damit die darauffolgenden Antworten in Relation zum abgefragten Begriffsverständnis gesetzt werden können. Es gab in diesem Bereich eine sehr starke Schnittmenge bei den Interviewpartnern. P1 definiert Agilität das selbständige Teams in kleinen Iterationen oder Zyklen, regelmäßig Produkt liefern und vom Kunden Feedback einholen. Diese Selbständigkeit und Selbststeuerung, sowie das iterative Vorgehen und die Zusammenarbeit mit dem Kunden wird auch von allen anderen Interviewpartnern genannt. Die beiden Interviewpartner P2 und P3 nennen nochmal explizit das Agile Manifest und die darin enthaltenen Werte und Prinzipien wie in Kapitel 2.3 dargestellt. Es ist daher auszugehen, dass alle Interviewpartner zumindest in großen Teilen ein analoges Verständnis von Agilität haben.

4.1.2 Metriken innerhalb der agilen Softwareentwicklung

Bei den Metriken innerhalb der agilen Softwareentwicklung gibt es weniger Deckungsmenge, was aber von P1, P2 und P3 genannt werden als wichtigste Kennzahlen sind die Velocity und die Story Points bzw. generell die Entwicklungsgeschwindigkeit. P4 arbeitet firmenintern in der Softwareentwicklung noch eher nach einem traditionellen Wasserfallmodell, beschreibt aber für das Projektmanagement bereits Backlog und Burndowncharts als wichtige Metriken bzw. Analysetools.

4.1.3 Priorität bzw. Präferenz bei Kennzahlen

Bei der Präferenz herrscht zwischen den Interviewpartnern eine hohe Kohärenz mit unterschiedlichen Benamungen, abhängig vom angewandten Ansatz. P4 spricht vom Durchsatz also Arbeitspaketen pro Sprint, P3 nennt die wichtigste Kennzahl die Time-to-Market, P2 die Durchlaufzeit innerhalb des Kanbanboards und P1 die bereits erwähnte Velocity und Story Points im Scrumframework. Daher ist für alle Interviewpartner die wichtigste Kennzahl die benötigte Entwicklungszeit für Arbeitspaket. P2 und P3 betonen aber bereits die Wichtigkeit von Softwarequalität als zusätzlich Kennzahl

4.2 Produktqualität

Tabelle 4 zeigt die Generalisierungen der Kategorien K2-1 bis K2-3. In Anhang C – Qualitative Inhaltsanalyse können zusätzliche Informationen wie Kürzel des Interviewpartners und Paraphrase überprüft werden.

Kategorie	Generalisierung
K2-1 Begriffsverständnis von Produktqualität in agiler SWE	<p>Interne Evaluationsprozess im Laufen. Produktqualität ist durch Kundenzufriedenheit und Bugreports stark beeinflusst.</p> <p>Kundenzufriedenheit soll durch Feedbackmöglichkeiten und Usability Test/Scores aktiv abgefragt werden.</p> <p>Produktqualität ist das positive absolvieren mehrere Quality Gates mit mehreren Testzyklen (Unit, Integrations, User Acceptance)</p> <p>Erfolgreiche Tests entscheiden, ob ein Feature im Kanban Prozess weiterwandern darf.</p> <p>Produktqualität in Außensicht ist Bedienbarkeit, Mehrwert für den Kunden, fehlerfrei, Responsiveness.</p> <p>Produktqualität in der Innensicht ist Performance und noch wichtiger Wartbarkeit da diese sich direkt auf die Velocity auswirkt.</p> <p>Produktqualität = Kundenzufriedenheit, interne Wartungskosten und Wartbarkeit, Usability, Systemverfügbarkeit und Stabilität</p>
K2-2 Bekannte Metriken und Eignungsgrad - Produktqualität	<p>Metriken müssen auf interne Eignung geprüft werden. Wichtigste für die Produktqualität wahrscheinlich die Anzahl der Bugs.</p> <p>Bei neuen Features sollte der Mehrwert für den Kunden überprüft um die Produktqualität sicherzustellen.</p> <p>Die verwendete Metrik dafür ist die Test Success Rate sowie durchgelaufene Test. Zudem erfolgt Linting, Code Coverage und statische Codeanalyse.</p> <p>Linting wird aggressiv für Einhaltung Coding Style/Guidelines genutzt. SonarQube für Analyse zum Refactoring des Backends</p> <p>Wartbarkeit selbst nicht messbar. Code Qualität und Coding Style durch Tools stellvertretend.</p> <p>Metriken Codequalität: Funktionen (Codelänge, Verschachtelungen), Anzahl von Funktionen pro Klasse, Indirektionen</p>

	Bugreports pro Modul und durchschnittliche Lösungszeit und Schweregrad als Metrik. angefallene Wartungskosten als monetärer Parameter Geschwindigkeit der Entwicklungszeit pro Modul, Verlangsamung abnehmende Produktqualität. Kundenzufriedenheit
K2-3 Einsatz von Metriken innerhalb der Firma - Produktqualität	Intern werden aktuell nur die Bugs pro Release erhoben Backend hat intern die Stabilität der API Kommunikation als Priorität. Frontend die erfolgreichen User Interface Tests. Tools sekundär, Prozess wichtiger(Pair Programming, Reviews) Bei Tools wichtig: statische Typisierung, Codeformatter Einsatz Linting wohldosiert, SonarQube für Trends. Merge-,Pullreques und Reviewanalyse für kurzfristige Codequalitätüberprüfung Reviews schwer messbar, Gibt viele dutzend andere Metriken Intern Bugreports pro Modul, im Reportzeitraum gelöst, Länge der ungelösten Zeit, Unit- und Integrationstestabdeckung

Tabelle 9: Generalisierung der Paraphrasen der Kategorie K2 – Produktqualität ©Harald Beier

4.2.1 Begriffsverständnis von Produktqualität in der agilen Softwareentwicklung

P3 teilt die Produktqualität in eine Innen- und Außensicht was für die weitere Analyse als Begrifflichkeit bzw. auch übernommen wird, da diese Kategorisierung eine leichtere Differenzierung der verschiedenen Metriken ermöglicht. Bei der Außensicht der Produktqualität gibt es einen Konsens der Interviewpartner P1, P3 und P4 das Kundenzufriedenheit die essenzielle Kenngröße ist. Es sollte mit Usability Tests/Scores und Feedbackmöglichkeiten gearbeitet werden, um sicherzustellen das den Kunden ein Mehrwert geliefert wird und nicht an den Bedürfnissen und Anforderungen der externen bzw. internen Stakeholder vorbei entwickelt wird.

Bei der Innensicht stechen das Schlagwort Wartbarkeit, Wartungskosten sowie Stabilität und Performance ins Auge. P3 betont das Wartbarkeit sich auch auf die in Kapitel 5.1 von allen Interviewpartnern als wichtigste Metrik die Entwicklungsgeschwindigkeit auswirkt. P2 stellt eine proaktive Vorgehensweise durch Testzyklen (Unit Test, Integrationstest und User Acceptance Test) und das Absolvieren von Quality Gates bereits als integralen Bestandteil für das Fortschreiten eines Issues oder Feature im Kanbanprozess in den Vordergrund trotz der geringsten MitarbeiterInnenanzahl aller Interviewpartner. P1 setzt aktuell eher auf einen reaktiven Ansatz um, in dem die Bugreports als Kennzahl für die Innensicht herangezogen wird.

Es wird in diesem Fall die Gewichtung für die späteren Überarbeitung des Kriterienkataloges für die Innensicht der Produktqualität auf die von P2,3 und P4 betonte Wartbarkeit gelegt und damit dem proaktiven Ansatz von P2 der Vorzug gegeben.

4.2.2 Bekannte Metriken und Eignungsgrad für die Produktqualität

P2 stellt fest das Wartbarkeit als Metrik für die Innensicht selbst nicht messbar ist und dafür andere Parameter herangezogen werden müssen. P2 und P3 legen einen Schwerpunkt auf

Codequalität und Umsetzung bzw. die Adhärenz eines definierten Codestyle. P2 betont sowohl Code Coverage, statische Codeanalyse sowie die Test-Success Rate als wichtige Kennzahlen für die Innensicht. P2 betont auch die Bedeutung von aggressiven Linting insbesondere bei interpretierten Programmiersprachen. (vgl. Tomasdottir et al., 2020) für die Innensicht der Produktqualität. P3 beschreibt insbesondere bei Codequalität unterschiedliche Parameter die als typische Metriken bei Reviews beachtet werden sollten wie Verschachtelungen, Indirektionen bei Klassenorientierten Programmiersprachen, Codelänge von Funktionen sowie Laufzeiten. P4 und P1 nennen Bugreports pro Feature bzw. Modul und P4 definiert zusätzlich die Parameter durchschnittliche Lösungszeit, Schweregrad des Fehlers und dadurch angefallene Wartungskosten. P4 nennt wiederum die Entwicklungsgeschwindigkeit und das eine Abnahme als Indiz für eine sinkende Produktqualität bzw. Wartbarkeit herangezogen werden kann. Für die Kundenzufriedenheit selbst wird keine spezifische Metrik von den Interviewpartnern genannt.

Aus den Aussagen der Interviewpartner ist ein starker Fokus auf Codequalität, Adhärenz zu Codestyle und Best Practices sowie verschiedene Testmetriken abzulesen. Diese wären als proaktive Metriken für die Gewährleistung einer guten Produktqualität zu bewerten, als reaktive werden wiederum Bugs, Schweregrad und monetäre Kosten betont. Im Bereich der Kundenzufriedenheit die zwar als wichtigste Kennzahl zuvor genannt wurde, gibt es aber keine spezifische Kennzahl, die genannt wird, außer den in Kapitel 5.2.1 genannten Usability Score. Hier wird ein Abgleich mit der wissenschaftlichen Literatur notwendig sein.

4.2.3 Einsatz von Metriken zur Messung der Produktqualität innerhalb der Firma

Bei den bereits erhobenen Metriken für die Produktqualität, gibt es zwischen den Interviewpartnern eine gewisse Diskrepanz. P2 und P3 setzen sehr stark auf Tests und zum Teil Tools und Methoden wie statische Typisierung, Codeformater und Linting. Beide betonen aber das Tools sekundär sind, das aber die Prozesse und Quality Gates wichtiger sind für die Produktqualität z.B. Pair-Programming, Codereviews sowie Kontrollen der Merge- und Pullrequests. Auch bei P4 werden Unit- und Integrationstestabdeckung als Kennzahl für die Produktqualität firmenintern herangezogen.

Beide Interviewpartner setzen zu Analyse SonarQube (SonarSource S.A, 2021) ein um Trends, Komplexität und Qualität im Code überprüfen zu können. P1 verwenden aktuell nur Bugreports pro Release quantitativ erhoben aber nicht in einen qualitativen Kontext bzw. in Relation zu anderen Kennzahlen gesetzt. In der Firma von P4 werden zudem als zusätzliche Parameter die Menge an gelösten Bugs innerhalb eines Reportzeitraum sowie die Dauer die ein Bug ungelöst bleibt in Relation zu quantitativen Menge gesetzt.

4.3 Teamproduktivität

In diesem Kapitel, werden wie in den Abschnitten zuvor die Generalisierung betreffend die Kategorie „Teamproduktivität“ ausgewertet. Zu leichten Übersichtlichkeit und für etwaige auf

diese Bachelorarbeit aufbauenden Forschungsarbeiten, werden die Aussagen zu Beginn des Kapitels wieder nachfolgend tabellarisch dargestellt.

Kategorie	Generalisierung
K3-1 Begriffsverständnis Teamproduktivität in SWE	<p>Teamproduktivität wird durch die Velocity ausgedrückt</p> <p>Mehrwert wird über Story Points definiert. Bugs und Regressionen erhalten keine Story Points</p> <p>Teamproduktivität sind die Arbeitspaket die ein Team fertigstellt. Diese können mit Business Value versehen werden(Gefahr der Demotivation)</p> <p>Wichtigste Kennzahl ist die Geschwindigkeit bis ein Kundenanliegen gelöst wird.(Cylce und Leadtimes)</p> <p>Teamproduktivität ist mit Einschränkungen, Story Points pro Sprint</p> <p>Einschätzung durch Team selbst</p> <p>Zeit bis zu Umsetzung gefordertes Feature. Story Points und Line of Codes möglicherweise falsche Anreize und verzerrend</p> <p>Intern nur Verhältnis Arbeitszeit von Kunden bezahlten Code und Code für Bugfixes, Projektbudget</p>
K3-2 Bekannte Metriken und Eignungsgrad - Teamproduktivität	<p>Da Story Points nur für Issues mit Value vergeben werden, kann Produktivität auf Story Points heruntergebrochen werden.</p> <p>Story Points können nur sehr schwer mit einem monetären Wert gekoppelt werden.</p> <p>In der Agilität ist Teamproduktivität immer selbst gesteuert und organisiert und damit schwer messbar.</p> <p>Die einzige relevante Metrik bleibt was ist ein Team in der Lage zu leisten. Teamzusammenarbeit, Teamklima und Teamzufriedenheit tragen dazu bei.</p> <p>Wichtigste Kennzahl intern Story Points und gelieferte Menge. Über 85% grün, 85 -60 grau, unter 60% rot</p> <p>Velocity nur als Vergleichswert, ob Wartbarkeit oder Qualitätrequirements abnimmt</p> <p>Verhältnis produktiven zu nichtproduktiven Aufwänden als Kennzahl</p> <p>Theoretische Kennzahlen oft falsche Anreize. Interes Vorgehen produktiv zu nichtproduktiv funktioniert gut.</p>
K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.	<p>Interne Hauptkennzahlen sind Velocity und Story Points und die Berechnung in Durchschnittszeiträumen.</p> <p>Aufgrund der Komplexität werden keine zusätzlichen Kennzahlen wie LoC, Laufzeiten,etc. gemessen.</p> <p>Als Zukunftsprojekt sollen auch Dauer von Request und Laufzeiten unter gleichbleibenden Bedingungen als Metrik eingeführt werden.</p> <p>Code Review sind ein essentieller Bestandteil des Produktqualitätsprozess. Es gibt hier aber keine gemessenen Kennzahlen dazu.</p> <p>Performanceanalyse auf Personenebene widersprechen den agilen Manifest und Werten.</p> <p>Beim intern Kanbanprozess geht es um Dimensionierung und Bearbeitungsgeschwindigkeit der Issues.</p>

	<p>Kanban erlaubt eine hohe Schlagkraft und kurze Umsetzung. Story Points und Release Planung können in kleinen Teams zu viel Overhead sein.</p> <p>Für externe/interne Stakeholder außerhalb SWE nur Time-To-Market Zeit von Features wichtig</p> <p>Teamtrends werden anhand Smileys und Retrospektive am Ende des Sprints</p> <p>Feedbackgespräche innerhalb von 14 Tagen</p> <p>Intern alle zwei Wochen Verhältnis produktiv/unproduktiv, monatlich Projektbudget</p>
--	---

Tabelle 10: Generalisierung der Paraphrasen der Kategorie K3 – Teamproduktivität ©Harald Beier

4.3.1 Begriffsverständnis von Teamproduktivität in der agilen Softwareentwicklung

Die Interviewpartner haben von Teamproduktivität ein beinahe uniformes Verständnis, dass es sich hierbei um die Menge der gelieferten Features bzw. Softwareprodukte handelt. Am pointiertesten bringt dies Interviewpartner P2 zum Ausdruck das es sich primär um die Geschwindigkeit handelt, bis Kundenanliegen gelöst werden.

Es handelt sich als generell um Story Points bzw. Arbeitspakete, die innerhalb einer gewissen Zeiteinheit geliefert werden. P4 merkt an, dass der monetäre Faktor auch berücksichtigt werden muss, daher welche Menge der Arbeitspakete bzw. Teile des Codes den Kunden direkt verrechnet werden können und wie diese im Verhältnis zu Arbeitspaketen, die nicht verrechnet werden können, steht. Es muss besonders bei Individualentwicklungen aber auch bei Software as a Service auf das geplante Budget Rücksicht genommen werden.

4.3.2 Bekannte Metriken und Eignungsgrad für die Teamproduktivität

P1 und P3 benennen Story Points als die wichtigste Kennzahl und die damit zusammenhängende gelieferte Menge als wichtigste Metrik und Kennzahl. P3 definiert eine Bandbreite inwieweit die angekündigten bzw. prognostizierte Story Points durch ein Team nach einem Sprint auch wirklich geliefert werden. P4 nennt das Verhältnis der produktiven zu nichtproduktiven Aufwände als Kennzahl.

P2 verengt dies und definiert als einzige relevante Metrik die Menge, die ein Team in der Lage ist zu leisten. Dies wird durch Teamzusammenarbeit, Teamzufriedenheit und Teamklima aber stark beeinflusst. Runtergebrochen ist die wichtigste Kennzahl aber für alle Interviewpartner die Liefermenge. P2 betont das in der Agilität die Selbständigkeit essenziell ist, P3 betont zusätzlich, dass Velocity eigentlich nur als Indikator fungieren kann betreffend der Wartbarkeit bzw. um Trends frühzeitig zu finden. Es gibt zwischen den Interviewpartner einen gewissen Konsens das viele Metriken betreffend der Teamproduktivität mit einem gewissen Skeptizismus betrachtet werden müssen.

4.3.3 Einsatz von Metriken zur Messung der Teamproduktivität innerhalb der Firma

Bei den intern verwendeten Metriken zu Teamproduktivität, gibt es teilweise stark abweichende Antworten, abhängig welches agile Framework verwendet wird. P2 legt aber direkt fest, dass Performanceanalyse auf Personenebene abzulehnen ist, da dies dem agilen Manifest und dessen Werten widersprechen. P3 beschreibt das für externe Stakeholder generell auch intern nur die Time-to-Market als Kennzahl herangezogen wird. Das für die die Teamproduktivität intern mit Feedbackzyklen und 1:1 alle 14 Tagen gearbeitet wird, um Trends ableiten zu können.

P1 definiert als interne Hauptkennzahlen Velocity und Story Points und hält fest das Lines of Code bzw. andere Parameter aufgrund der Komplexität nicht gemessen werden, bzw. möglicherweise zu falschen Schlüssen führt. P4 nennt das alle zwei Wochen Verhältnis zwischen produktiven Stunden und unproduktiven Stunden berechnet wird und monatlich durch die Projektleiter die Einhaltung des Projektbudget überprüft wird.

Bis auf P4 werden durch die Interviewpartner keine monetären Parameter bzw. Faktoren genannt für die Messung der Teamproduktivität. Es wird von allen intern ein großer Wert auf die Bearbeitungsgeschwindigkeit und die Liefermenge von funktionalen Arbeitspaketen gelegt und es wird zudem die Bedeutung zumindestens durch P2 und P3 klar auch die Selbständigkeit und Selbstverantwortung der agilen Teams gelegt.

4.4 Vorhersagbarkeit

Das letzte Unterkapitel der Auswertung der Interviews widmet sich der Aufarbeitung der Thematik Vorhersagbarkeit. In der nachfolgenden Tabelle 6 werden die Generalisierungen der Paraphrasen für die Kategorien K4-1 bis K4-3 zur besseren Übersichtlichkeit zusammengefasst.

Kategorie	Generalisierung
K4-1 Begriffsverständnis Vorhersagbarkeit in SWE	<p>Roadmaps und Vorhersagbarkeit leben von Priorisierung und dem Austarieren zwischen Zeit/Qualität/Kosten.</p> <p>Durch das festsetzen von zwei Parameter wird einer variable, oft ist dies die Zeit.</p> <p>Vorhersagbarkeit über Teams ist schwierig, es darf keinen absoluten Story Point Wert geben, dies erfolgt wenn eine Ebene darüber.</p> <p>Velocity ist für die Vergleichbarkeit die beste Metrik. Dies muss immer nur Team zu Team verglichen werden. Abnahme oder Zunahme kann zwischen Teams verglichen werden</p> <p>Bei mehreren Teams sollten Vorhersagen auf Storyebene durch Teams erfolgen und darüber in den Epics durch den Produktmanager.</p> <p>Vorhersagbarkeit erfolgt anhand Backlog und der Menge an Story Point die pro Sprint geschafft werden</p> <p>Vorhersagbarkeit von Größe Features und Arbeitspaketen abhängig. Mehr Größe führt zu mehr Unsicherheit</p> <p>Vorhersagbarkeit bei Performanceoptimierung sehr schwierig</p>

	Vorhersagbarkeit = Zuverlässigkeit Liefertermin, Umfang, Qualität, Kosten vorab zu beurteilen
K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit	<p>Vorhersagbarkeit wird durch die Treffgenauigkeit des Story Point Commitment der Teams möglich</p> <p>Vorhersagbarkeit für neue Features ist leichter als bei Wartung/Austausch von Legacy Code</p> <p>Für den Einsatz von Story Points wird eine Referenzstory benötigt. Diese hat den Wert drei. Alle anderen Schätzungen erfolgen in Relation dazu in Fibonacci Folge</p> <p>Die Schätzung der Referenzstory ist wichtig damit bei 2 Wochen Sprint und pro 5-7 Personen Team circa 20-25 Story Points geschafft werden.</p> <p>Für die Vorhersagbarkeit und Planung durch den Product Owner wird die Durchschnittsvelocity über 4-6 Sprint erfasst.</p> <p>Die wichtigsten Metriken sind die Velocity und ein geschätzter Backlog mit entsprechend definierten Stories</p> <p>Grundlage für korrekte Schätzungen bildet ein gleichbleibendes Team.</p> <p>Neue Produkte erfordern Backlog Aufbau und 3 Sprints zum Erfahrungen machen, selbiges gilt für neue Teams.</p> <p>Für Framework Änderungen sind eine Iteration Zero und Spikes/Forschungsaufträge mit Zeitlimit anzuwenden.</p> <p>Für Spikes müssen in Sprint Story Point vorgesehen werden</p> <p>User Stories werden vom Team geschätzt, Spikes vom Product Owner</p> <p>Qualität Anforderungsaufnahme sehr wichtig(innerhalb Entwicklungsabteilung weniger)</p> <p>Definition of Ready und Refinement sehr wichtige Parameter</p> <p>Vergleichbarkeit gelieferten Arbeitspakete mit geplanten im Beobachtungszeitraum. Resultat = Liefertermin & Kostenabweichung</p>
K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit	<p>Intern werden für die Vorhersagbarkeit das Commitment herangezogen und für die Erfüllung der Erwartungen die Definition of Ready und die Definition of Done.</p> <p>Erfüllungsgrad der Wünsche von Stakeholder ist an Domänen und Produktwissen der Entwickler gekoppelt.</p> <p>Das Commitment und die das Erreichen dieses Commitment im Zeitverlauf wird auch gemessen</p> <p>Intern werden Qualitätskennzahlen und Quality Gates mit Cycle Time für die Vorhersagbarkeit genutzt.</p> <p>Vorhersagbarkeit sind selbständige Teams mit Fullstackentwicklern wichtig.</p> <p>Intern Requirements Abklärung Product Owner und Lead Engineer. Teilweise sogar 2 Refinements</p> <p>Definition of Done enthält auch Wünsche der Stakeholder. 3/4 sind aber vom Team selbst.</p> <p>Für gute Vorhersagbarkeit ist Stabilität der Requirements wichtig. Roadmap für Veränderungen und Frühwarnsystem.</p> <p>Roadmap als Ankerpunkt. Tools für Vorhersagbarkeit sind Roadmap, Feedbackzyklen sowie Priorisierung.</p>

	Kosten intern schwierig da Fixpreis. Durchsatz und offene Issues Backlog für Vorhersagbarkeit des Liefertermins
	Abweichung geplante/tatsächliche Arbeitspakete, Arbeitspakete im Backlog

Tabelle 11: Generalisierung der Paraphrasen der Kategorie K4 – Vorhersagbarkeit ©Harald Beier

4.4.1 Begriffsverständnis von Vorhersagbarkeit in der agilen Softwareentwicklung

Als zusammenfassende Aussage bietet sich die Generalisierung des Interviewpartner P4 an. Dieser definiert Vorhersagbarkeit als die Zuverlässigkeit mit der vorab Liefertermin, Umfang, Qualität sowie Kosten beurteilen zu können. P1 stellt fest, dass Roadmaps sowie Vorhersagbarkeit immer in Relation zu Priorisierung der Arbeitspakete und dem Austarieren zwischen den Faktoren Zeit, Qualität und Kosten gesetzt werden muss. Es ist in diesem Fall eine klare Schnittmenge zwischen den beiden Interviewpartnern zu erkennen.

P2 und P3 gehen dazu mehr ins Detail und beziehen sich direkt auf Kennzahlen bzw. Metriken in der Agilität. P2 nennt die Velocity als wichtigste Metrik und stellt aber fest, dass dies immer nur für Vergleichbarkeit zwischen Sprint des gleichen Teams sorgt. P3 definiert ebenso die Größe des Backlog sowie die Menge an Story Points, die pro Sprint bearbeitet werden können als wichtigste Metriken der Vorhersagbarkeit.

Alle Interviewpartner stellen fest, dass Vergleichbarkeit bzw. Vorhersagbarkeit über die Teams hinweg schwierig bzw. unmöglich ist und dies immer nur auf ein Team bezogen, werden sollte. P3 betont zusätzlich das die Größe der Features bzw. der Arbeitspaketen auch die Vorhersagbarkeit stark beeinflusst. Je mehr Umfang bzw. Größe ein Arbeitspaket hat, desto größer wird die Unsicherheit, diese Einschätzung wird auch durch die Arbeit von Rempel und Mäder geteilt. (Rempel & Mäder, 2015) Wenn die spezifischen Begrifflichkeiten, Terminus Technicus für die Auswertung nicht berücksichtigt werden, gibt es auch hier eine große Schnittmenge beim Begriffsverständnis der Interviewpartner. Der Nukleus der Vorhersagbarkeit ist die Zuverlässigkeit, in der es möglich ist Liefertermin, Umfang, Qualität sowie Kosten vorab zu beurteilen zu können heruntergebrochen werden.

4.4.2 Bekannte Metriken und Eignungsgrad für die Vorhersagbarkeit

Als Metriken wird von allen Interviewpartner die Velocity und das Verhältnis der zugesagten oder geplanten Arbeitspaketen bzw. Story Points und der tatsächlichen gelieferten Arbeitspaketen und Story Points. P2 geht ins Detail und beschreibt, dass bestimmte Parameter erfüllt sein müssen, damit Story Points als Metrik genutzt werden kann. Er legt klar fest, dass es eine gute Referenzstory geben muss, die mit 3 Story Points gewichtet wurde. Zu dieser Referenz kann dann jede andere User Story in Relation unter zu Hilfenahme der Fibonaccifolge geschätzt werden. Dies ist die Grundlage damit eine Velocity von 20-25 Story Points eines 5-7 Personen Teams innerhalb von 2 Wochen Sprints genutzt werden kann. Dies führt dazu das ein korrekt geschätzter Backlog mit entsprechend sauber definierten Stories und die Velocity, die wichtigste

Metrik für die Vorhersagbarkeit wird. P2 stellt zudem fest, dass ein gleichbleibendes Team die Grundlage für eine korrekte Schätzung ist.

P3 definiert das die Qualität der Anforderungsaufnahme und das Refinement von Arbeitspaketen besonders wichtig ist. Sowohl eine saubere und klar definierte Definition of Ready als auch die Definition of Done als Richtschnur für das Entwicklungsteam sind laut P3 wichtige Faktoren für die korrekte Vorhersagbarkeit.

P1 beschreibt, dass die Treffgenauigkeit der Verpflichtung eines Entwicklungsteam zu einer gewissen Menge an Story Points und umzusetzenden Stories die wichtigste Metrik für Vorhersagbarkeit ist. Als Addendum zu den vorhergegangenen Aussagen nennt P4 noch das durch diese Vergleichbarkeit es möglich ist als Resultat Liefertermin und Kostenabweichung besser zu definieren.

P2 beschreibt im Interview einen Prozess, wie mit neuen Entwicklungsteams, neuen Produkten, Änderungen des Frameworks bzw. der Programmiersprache und neuen Technologien umgegangen werden kann, um die zuvor vorhandene Vorhersagbarkeit wieder neu zu etablieren und zu entwickeln. Ausgehend von einer Iteration Zero die als Fundament der Forschungsarbeit dient, sollten mindestens 3 Sprints genutzt werden, um Erfahrungen zu machen und eine neue Velocity zu etablieren. Sollten normale User Stories und Umstellungsaufgaben gleichzeitig erfolgen, werden die Forschungs- bzw. Umstellungsaufgaben als Spikes mit einem Zeitlimit in den Sprint integriert. Bei Spikes sollte eine Umkehr der normalen Schätzung erfolgen. Spikes werden durch den Product Owner gewichtet und damit mit einem Zeitlimit versehen.

Zusammenfassend ist bei allen Interviewpartner ein Konsens zu erkennen, das als Metrik für die Vorhersagbarkeit korrekte Schätzungen der gelieferten Arbeitspakete und die Adhärenz der selbstständigen Teams zu diese angekündigten Liefermenge am besten funktioniert. Diese Verpflichtung bzw. Commitment und die Treffgenauigkeit sind die Grundlage der Vorhersagbarkeit für alle Interviewpartner. Es wird aber klar festgestellt das Parameter wie die Qualität der Anforderungserhebung, korrekte Referenzstories sowie Stabilität des Teams für die Vorhersagbarkeit wichtig sind.

4.4.3 Einsatz von Metriken zur Messung der Vorhersagbarkeit innerhalb der Firma

P1, P3 nutzen intern als interne Metriken für die Vorhersagbarkeit „Definition of Ready“ und „Definition of Done“ und deren Erfüllungsgrad sowie das Verhältnis von zugesagten Arbeitspaketen und tatsächlich gelieferten Arbeitspaketen. Letzteres wird auch innerhalb der Firma von P4 genutzt.

P2 betont das die primäre Kennzahl für die Vorhersagbarkeit, Qualitätskennzahlen sind und die Erfüllung der Quality Gates ist. Durch diese Qualitätskennzahlen kann dann in der Kombination mit der Cycle Time als das Durchlaufen des Kanbanprozesses eine sehr gute Vorhersagbarkeit erreicht werden.

5 DISKUSSION

In der erfolgten Auswertung ist in den vorangegangenen Kapiteln erkennbar, dass bei den Interviews der FachexpertInnen ein ähnliches Verständnis betreffend Agilität und der damit verknüpften Werte und Prinzipien gibt. Es ist daher nicht notwendig, die Aussagen unterschiedlich der Aussagekraft der einzelnen ExpertInnen zu gewichten. Alle 4 Firmen bilden unterschiedliche Parameter ab und befinden sich auf unterschiedlichen Stufen der agilen „Stairway to Heaven“. Die Einschätzung des Autors ist das die Firma von P4 sich auf Stufe 1 befindet, die Firma von P1 auf Stufe 2-3, P3 auf Stufe 3-4 und P2 ist am weitersten fortgeschritten und hat bereits Stufe 4-5 erreicht. Generell gibt es eine hohe Überschneidung bei den Themenpunkte und insbesondere bei Teamproduktivität und auch Vorhersagbarkeit einen hohen Konsens über die Definition. Es wird stark auf Story Points oder Dimensionierung sowie Velocity gesetzt, bzw. Cycle oder Leadtimes. Es wird von allen InterviewpartnerInnen die Bedeutung der Selbständigkeit, Selbstverantwortung und Selbstorganisation der agilen Teams betont. Ein direktes Herunterbrechen auf personelle Ebene bei diesen beiden Metriken, wird von keinem der ExpertInnen befürwortet, sogar direkt abgelehnt.

Betreffend Produktqualität gibt es ein klares Bekenntnis zu hohen Standards, da diese die Wartbarkeit erhöhen und überhaupt erst die Grundlage für das Voranschreiten entlang des Kanbanprozess liefern. Insbesondere der Aspekt der Wartbarkeit, der sich direkt auf die Faktoren Teamproduktivität und Vorhersagbarkeit auswirkt, wird betont. P3 stellt sehr pointiert fest, dass wenn es im Code bzw. in der Software ganze Bereiche gibt, die sich niemand angreifen traut, die gesamte Time-to-Market verlangsamt wird und auch die Produktqualität darunter leidet. Dass die Umsetzung einer rigiden Produktqualität orientierten Umsetzungsstrategie auch in sehr kleinen Entwicklungsteams möglich ist, wird durch den Interviewpartner P2 unterstrichen. Inwieweit fehlende Wartbarkeit sich auch monetär drastisch auswirken kann und damit den eigenen Handlungsspielraum, Schritt für Schritt einengt betont P4. Diese Ansicht entspricht auch dem bereits durch die Literaturrecherche gezeichneten Bild in Kapitel 2.5.2, es ist daher die Ansicht des Autors, dass Produktqualität die fundamentale Dimension ist. Generell gibt es eine starke Interdependenz zwischen den Dimensionen. Gleichbleibende und nachvollziehbare Teamproduktivität, die auf hohe Produktqualitätsstandard basiert führt für das Produktmanagement zu einer gute Vorhersagbarkeit der Lieferung von Arbeitspaketen.

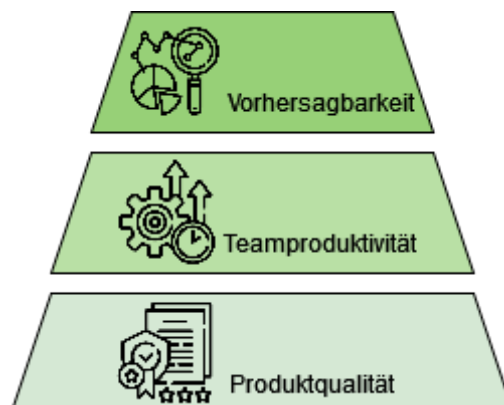


Abbildung 6: Darstellung Dimensionen mit Icons von [Freepik](#) und [Eucalyp](#) von [www.flaticon.com](#) ©Harald Beier

6 FAZIT

Innerhalb dieser Arbeit war es das Ziel die Forschungsfrage *Welche Kriterien sind für KMUs im Bereich Dienstleistungen in der automatischen Datenverarbeitung und Informationstechnik, für die Bewertung der Leistung agiler Teams entscheidend und sollten als erster eingeführt werden?* zu beantworten. Im Zuge dessen wurden sowohl die in der Literatur vorhandenen Dimensionen summiert und Metriken zugeordnet. Als weiterer Schritt wurden danach ExpertInnen interviewt, um den notwendigen Input von PraktikerInnen und LektorInnen zusätzlich einarbeiten zu können. Dafür wurde auf eine qualitative Inhaltsanalyse zurückgegriffen und in Kapitel 5 ausgewertet.

6.1 Forschungsfrage – Erkenntnisse

Die generalisierten Aussagen der Expertinneninterviews wurden 4 Kategorien zugeordnet, wobei für die Beantwortung der Forschungsfrage Kategorie 1 nicht essenziell ist. Nachfolgend werden die unterschiedlichen Kriterien und Metriken summiert pro Kategorie in Tabelle 11 dargestellt.

Produktqualität
Kundenzufriedenheit (P1, P2, P3, P4), Quality Gates(P2, P3) vorhandene Testzyklen – Unittest, Integrationstest, User Acceptance Tests(P2, P3, P4), Testerfolgs/fehlerrate(P2, P3), Adhärenz zu Coding Style/Guidelines (P2,P3,P4), Bugs pro Sprint/Modul (P1, P4)
Teamproduktivität
Story Points (P1, P2, P3, P4), Velocity (P1), Verhältnis produktiv /unproduktiv (P4), Cycle und Leadtimes (P2, P3)
Vorhersagbarkeit
Velocity (P1, P2, P3, P4), Story Points als Grundlage (P2), Größe der Features (P3), Qualität Anforderungsaufnahme (P3), Stabilität der Anforderungen (P3), gleichbleibendes motiviertes Team (P2, P3), korrekt geschätzter Backlog (P2, P3, P4), Verhältnis zugesagte/gelieferte Arbeitspakete (P1, P2, P4)

Tabelle 12: Erhobene Metriken durch die Interviews nach den drei Dimension und Nennung gegliedert ©Harald Beier

Diese Metriken werden nun den verwandten Metriken die in der Literaturrecherche gefunden wurden zugeordnet, wenn keine passende Metrik gefunden werden kann, werden die Aussagen hinzugefügt.

6.2 Ergänzung des Kriterienkataloges

Im nachfolgenden Schritt wird, der durch die Literaturrecherche bereits gewichtete Kriterien- und Metriken Katalog mit den Antworten, die im Zuge der qualitativen Inhaltsanalyse identifiziert wurden, erweitert. Die geänderten bzw. addierten Spalten werden fett markiert.

Dimension	Metrik	Nennung	Gewichtung
Produktqualität (23 Metriken und 49 Nennungen)	Aufwandsschätzung (8)	10	1
	Fehlerzahl (8)	10	1
	Kundenzufriedenheit (6)	10	1
	Anzahl der Unit Test (5)	8	1
	Testfehlerrate (3)	5	2
	Buildstatus (2)	2	3
	Getestete Features anbieten (1)	3	3
	Nummer der Testfälle (1)	1	3
	Technische Altlasten/Schulden in Kategorien (1)	1	3
	Technische Altlasten/Schulden im Aufwand (1)	1	3
	Verstöße gegen die statische Codeanalyse (1)	3	3
	Defekt-Trend-Indikator (1)	1	3
	voraussichtliche Anzahl der Defekte (1)	1	3
	Testerfolgsrate (1)	3	3
	aufgeschobene Behebung von Fehler (1)	1	3
	Testabdeckung (1)	4	2
	Zuwachsrate der Test (1)	1	3
	Anzahl der Regressionen in einer Codeversion (1)	1	3
	Anzahl wie oft ein Fehler gemeldet wird (1)	1	3
	Geschwindigkeit der Ausarbeitung von Anforderungen und Features (1)	1	3
	Varianz bei Übergabe (1)	1	3
	Wartungsaufwand (1)	1	3
	Kostenarten der Anforderung (1)	1	3
Teamproduktivität (11 Metriken, 28 Nennungen)	Velocity (12)	13	1
	Durchlaufzeit (4)	5	2
	Check-Ins/Commits pro Tag (3)	3	3
	Zykluszeit (2)	3	3

	das voraussichtliche Enddatum der Arbeitspakete (1)	1	31
	erledigte Arbeitspakete (1)	5	2
	Prozent der abgeschlossenen User Stories (1)	5	2
	Prozentsatz der für den Sprint vorbereiteten Stories (1)	1	3
	Dauer, bis ein fehlgeschlagener Build gefixt ist (1)	1	3
	Teamffizienz (1)	1	3
	Bearbeitungszeit (1),	1	3
Vorhersagbarkeit (10 Metriken, 20 Nennungen)	Arbeitspakete im Backlog/ Bearbeitung (7)	10	1
	Durchlaufzeit (4)	8	1
	Zykluszeit (2)	2	3
	das voraussichtliche Enddatum bei Lieferung der Arbeitspakete (1)	1	3
	Kosten-Leistungsindex (1)	1	3
	Zeitplan-Leistungs-Index (1)	2	3
	geplante Velocity (1)	1	3
	durchschnittliche Geschwindigkeit (1)	1	3
	Durchsatz (1)	1	3
	Größe der Queue der Arbeitspakete (1)	1	3
	Größe der Features (1)	1	3
	Stabilität und Qualität der Anforderungen (1)	1	3
	Gleichbleibendes motiviertes Team	1	3

Tabelle 13: Um Ergebnisse der Auswertung erweiterter Gewichteter Kriterienkatalog ©Harald Beier

Einige der in der obigen Tabelle genannten Metriken haben eine starke Überschneidung und werden daher zusammengefasst. Anhand der Erweiterung des vorhandenen Kriterienkataloges lassen sich für jede Dimension drei Metriken feststellen die für KMUs sinnvoll sind priorisiert einzuführen.

Produktqualität:

1. Integration von verschiedenen Softwaretest entsprechend der Testpyramide (Anzahl der Unit Test 8 + Testfehlerrate 5 + Getestete Features anbieten 3+ Testerfolgsrate 3 + Testabdeckung 3 = 21)

2. Messen der Fehleranzahl innerhalb einer Iteration und wie lange es dauert diese zu beheben (Fehlerzahl 10 + Anzahl wie oft ein Fehler gemeldet wird 1 + voraussichtliche Anzahl der Defekte 1 + aufgeschobene Behebung von Fehler 1 = 13)
3. In diesem Fall haben zwei Kennzahlen, die gleiche Menge an Nennungen:
 - a. Qualität der Anforderungserhebung 10
 - b. Kundenzufriedenheit 10

Teamproduktivität

1. Velocity 13
2. Abgeschlossene Arbeitspakete (erledigte Arbeitspakete 5 + Prozent der abgeschlossenen User Stories 5 = 10)
3. Durchlaufzeit/Zykluszeit (Durchlaufzeit 5 + Zykluszeit 3 = 8)

Vorhersagbarkeit:

1. Velocity 16
2. Arbeitspakete im Backlog/ Bearbeitung 10
3. Durchlaufzeit/Zykluszeit 10

Ausgehend von der in Kapitel 5 entwickelten Reihung der Dimensionen wird es daher empfohlen, das Hauptaugenmerk der Implementierung auf die Metriken der Produktqualität zu legen, damit der notwendige Grundstein gelegt wird für ein schnelles Voranschreiten entlang der agilen „Stairway to Heaven“

6.3 Limitationen

Im Rahmen dieser Arbeit wurden nur vier Expertinnen und Experten aus vier unterschiedlichen Unternehmen befragt. Aufgrund hoher Absagerate konnte nur eine kleine Gruppe befragt werden und die Ergebnisse, können damit nicht repräsentativ für die gesamte Branche gewertet werden. Die ExpertenInneninterviews stellen zudem eine subjektive Einschätzung dar, die in Relation zu der erfolgten Literaturrecherche gesetzt wurden. Die Arbeit bearbeitet die Thematik zudem auch nicht Branchen- oder Firmengrößen unabhängig, sondern fokussiert auf KMUs in der Softwareentwicklung die dem IT-KV angehören.

6.4 Ausblick

Zu Validierung des entwickelten gewichteten Kriterienkataloges, bietet es sich an die verschiedenen Metriken weitergehend innerhalb einer Nutzwertanalyse zu untersuchen und auch die entstehenden Kosten als Faktor zu berücksichtigen. Als nächster Schritt bietet sich eine begleitete Umsetzungsstudie, die den Kriterienkatalog auf die Auswirkungen in realen Firmenbedingungen testet.

ANHANG A - Interviewleitfaden

Interviewleitfaden:

Forschungsfrage:

Welche Kriterien sind für KMUs im Bereich Dienstleistungen in der automatischen Datenverarbeitung und Informationstechnik, für die Bewertung der Leistung agiler Teams entscheidend und sollten als erster eingeführt werden?

Interview Einstieg

- Begrüßung
- Hinweis, dass die Besprechung aufgezeichnet wird. Kontrolle der Einverständniserklärung
- Hinweis auf pseudonymisierung der Personendaten
- Aufnahme starten
- Ablauf des Interviews wiedergeben
- Thema und Ziel der Bachelorarbeit
- Welche Forschungsfrage wird behandelt

Fragen zur Person

1. Welche Position haben Sie im Unternehmen/Lehreinrichtung
2. Welchen Bezug haben Sie zu agilen Softwareentwicklung und Metriken für die Bewertung der Leistung agiler Teams

Agile Softwareentwicklung

3. Wie würden Sie agile Softwareentwicklung und agile Teams definieren?
4. In welchem Kontext setzen Sie bereits Metriken zu Bewertung agiler Softwareentwicklungsteams ein?
5. Welche Kennzahlen sind für Sie in der agilen Softwareentwicklung wichtig?

Produktqualität

6. Wie würden Sie Produktqualität von Lieferobjekten/Artefakten in der agilen Softwareentwicklung definieren?
7. Welche Metriken bzw. Kennzahlen eignen sich Ihrer Meinung nach, um die gelieferte Produktqualität von einem agilen Team zu bewerten?
8. Welche Kennzahlen davon messen Sie bereits in Ihrer Firma und welche halten Sie für die wichtigsten

Teamproduktivität

9. Wie würden Sie Teamproduktivität innerhalb einer agilen Iteration definieren?
10. Welche Metriken bzw. Kennzahlen eignen sich Ihrer Meinung nach, Teamproduktivität innerhalb einer agilen Iteration oder einem Jahr von einem agilen Team zu bewerten?
11. Welche Kennzahlen davon messen Sie bereits in Ihrer Firma und welche halten Sie für die wichtigsten

Vorhersagbarkeit

12. Wie würden Sie Vorhersagbarkeit der gelieferten Softwareartefakte innerhalb einer agilen Iteration definieren?
13. Welche Metriken bzw. Kennzahlen eignen sich Ihrer Meinung nach, Vorhersagbarkeit der gelieferten Artefakte innerhalb einer agilen Iteration oder einem Jahr von einem agilen Team zu bewerten?
14. Welche Kennzahlen davon messen Sie bereits in Ihrer Firma und welche halten Sie für die wichtigsten

ANHANG B - Kodierleitfaden

Kategorie	Definition	Ankerbeispiel	Kodierregeln
K1-1 Begriffsverständnis Agile Softwareentwicklung und agile Teams	Dieser Code wird verwendet bei Textstellen, welche das Definitionen von Agiler Softwareentwicklung und agilen Teams behandeln.	P1:“ Für mich persönlich heißt Agilität oder agile Softwareentwicklung, dass man selbstständige Teams hat, die in sich funktionieren und die in sehr kleinen Zyklen arbeiten“ P	Ausschließlich Aussagen die das Begriffsverständnis generell betreffen, keine Metriken oder firmenspezifischen Aussagen
K1-2 Metriken innerhalb der agilen Softwareentwicklung	Dieser Code wird verwendet bei Textstellen, welche das Metriken innerhalb Agiler Softwareentwicklung behandeln.	P1:“ Da muss man halt da die entsprechende Kurve kriegen zu sagen, na ja, wir wollen natürlich ein Produkt haben, das möglichst wenige Bugs hat, sodass der Kunde nicht angefressen ist bzw. der Kunde soll zufrieden sein. Aber man muss immer schauen, was bewirkt der/ dieses KPI/ was hat das für einen Einfluss vielleicht, wenn Mitarbeiter nur auf dieses eine Key PI hinarbeiten“	Ausschließlich Aussagen die generell Metriken behandeln, kein Begriffsverständnis
K1-3 Priorität bzw. Präferenz bei Kennzahlen	Dieser Code wird verwendet bei Textstellen, die Kennzahlen und deren Prioritäten behandeln	P2: „Ja, ja und nein. Also wir arbeiten jetzt momentan eben/ aufgrund der kleineren Struktur arbeiten wir sogar sehr, sehr stark nach Kanban. (...) Wo die Velocity ja dann schon ein wenig wieder weniger relevant ist. Weil die/ nach Kanban geht's ja eben eher um die Frage was ist die	Ausschließlich Aussagen die zu Kennzahlen allgemein behandeln und deren Priorisierung

		Durchschnittliche Laufzeit eines Tickets von ganz links nach ganz rechts“	
K2 – 1 Begriffsverständnis von Produktqualität in der agilen Softwareentwicklung	Dieser Code wird verwendet bei Textstellen, welche das Begriffsverständnis von Produktqualität in der agilen Softwareentwicklung behandeln.	P4:“ Für mich setzt sich die Produktqualität aus verschiedenen Faktoren zusammen, nach denen man Software bewerten kann. Als passende Bewertungsfaktoren sehe ich dafür Kundenzufriedenheit, Usability, Wartbarkeit und laufende Wartungskosten, Systemverfügbarkeit und Stabilität.“	Ausschließlich Aussagen die das Begriffsverständnis zu Produktivität betreffen, keine Metriken oder firmenspezifischen Aussagen
K2 – 2 Bekannte Metriken und deren Eignungsgrad	Dieser Code wird verwendet bei Textstellen, welche bekannte Metriken für die Produktqualität und deren Eignungsgrad in der Agiler Softwareentwicklung behandeln.	P3:“ Das Problem ist halt, Wartbarkeit lässt sich nicht messen. Das heißt, man muss stellvertretend für die Wartbarkeit halt bestimmte Sachen finden. Und man kann jetzt einerseits zurückfallen auf eine Reihe von Tools und zweitens Prozesse optimieren. Also dass man zum Beispiel einen bestimmten Coding Style halt mithilfe von Tools einfach forciert““	Ausschließlich Aussagen die bekannte Metriken zu Produktqualität beschreiben bzw. genauer definieren. Keine firmeninternen Metriken oder Begriffsverständnis
K2 – 3 Einsatz von Metriken innerhalb der eigenen Firma	Dieser Code wird verwendet bei Textstellen, welche den internen Einsatz von Metriken zu Erhebung der Produktqualität in der Agiler Softwareentwicklung behandeln.	P3: „Und das ist/ wenn ein Team eine bestimmte Größe erreicht, entstehen so viele Fehler, die sich durch statische Typisierung abfangen ließen, dass es/ also dass das eine der ersten Empfehlungen ist, die ich	Ausschließlich Aussagen zu firmenintern verwendeten Metriken zur Messung von Produktqualität

		an Firmen hätte, (...) die wachsen“	
K3 – 1: Begriffsverständnis von Teamproduktivität in der agilen Softwareentwicklung	Dieser Code wird verwendet bei Textstellen, welche das Begriffsverständnis von Teamproduktivität in der agilen Softwareentwicklung behandeln.	P3:“ Mit Einschränkung wie viele Issues einmal beim Sprint erledigt sind. Beziehungsweise, nachdem Issues ja unterschiedlich groß sind/ also wie viele Story Points erledigt sind. Allerdings eben mit Einschränkung. Weil Programmieren/ oder die Aufgabe vom Team ist ja nicht nur vor der IDE zu sitzen und irgendwelche Zeilen reinzuklopfen“	Ausschließlich Aussagen die das Begriffsverständnis zu Teamproduktivität betreffen, keine Metriken oder firmenspezifischen Aussagen
K3 – 2 Bekannte Metriken und deren Eignungsgrad	Dieser Code wird verwendet bei Textstellen, welche bekannte Metriken für die Teamproduktivität und deren Eignungsgrad in der Agiler Softwareentwicklung behandeln.	P2:“Das heißt, wenn mans jetzt so betrachtet, dann müsste man sagen, okay, die einzig relevante Metrik ist das, dass man sagt, okay, was ist das Team in der Lage zu leisten und wie lange ist das Team schon zusammen in einer bestimmten Konstellation?“	Ausschließlich Aussagen die bekannte Metriken zu Teamproduktivität beschreiben bzw. genauer definieren. Keine firmeninternen Metriken oder Begriffsverständnis
K3 – 3 Einsatz von Metriken innerhalb der Firma	Dieser Code wird verwendet bei Textstellen, welche den internen Einsatz von Metriken zu Erhebung der Teamproduktivität in der Agiler Softwareentwicklung behandeln.	P4:“ Wir werten das Verhältnis der produktiven und nicht produktiven Stunden aller Teams alle zwei Wochen aus. Die Projektbudgets werden zumindest monatlich von den Projektleitern geprüft“	Ausschließlich Aussagen zu firmenintern verwendeten Metriken zur Messung von Teamproduktivität
K4 – 1: Begriffsverständnis von Vorhersagbarkeit	Dieser Code wird verwendet bei Textstellen, welche das Begriffsverständnis von	P3: “Ich meine je einfacher das Issue ist, also je kleiner das Arbeitspaket ist, umso sicherer sind die	Ausschließlich Aussagen die das Begriffsverständnis zu

in der agilen Softwareentwicklung	Vorhersagbarkeit in der agilen Softwareentwicklung behandeln.	Vorhersagen. Je größer das Feature oder das Arbeitspaket ist, umso mehr Unsicherheit gibt es. Also das heißt, wenn man ein Feature hat, das in/ sagen wir mal jetzt in Summe 100 Story Points umfasst, dann ist das nicht weiter verwunderlich, wenn es auf einmal 150 braucht.“	Vorhersagbarkeit betreffen, keine Metriken oder firmenspezifischen Aussagen
K4-2: Bekannte Metriken und deren Eignungsgrad	Dieser Code wird verwendet bei Textstellen, welche bekannte Metriken für die Vorhersagbarkeit und deren Eignungsgrad in der Agiler Softwareentwicklung behandeln.	P2:“ Weil ich meine da sind wir wieder bei der Velocity. Wenn ich als Product Owner mein Team kenne dann kann ich damit alles planen. Wenn wir nach allen Regeln der Kunst vorgegangen sind. Und zwar weil sich in den vergangenen vier bis sechs Sprints, das ist so der Durchrechnungszeitraum für die Velocity, gezeigt hat, dass das Team in diesem Setup in der Lage ist, dieses Volumen an Aufgaben fertig zu machen“	Ausschließlich Aussagen die bekannte Metriken zu Vorhersagbarkeit beschreiben bzw. genauer definieren. Keine firmeninternen Metriken oder Begriffsverständnis
K4-3: Einsatz von Metriken innerhalb der Firma	Dieser Code wird verwendet bei Textstellen, welche den internen Einsatz von Metriken zu Erhebung der Vorhersagbarkeit in der Agiler Softwareentwicklung behandeln.	P4:“ Wir messen die Abweichung zwischen geplanten Aufwänden und tatsächlichen Aufwänden in Prozent. Genauso die Aufwände, welche als offene oder geplante Arbeitspakete im Backlog liegen.“	Ausschließlich Aussagen zu firmenintern verwendeten Metriken zur Messung von Vorhersagbarkeit

ANHANG C - Qualitative Inhaltsanalyse

Nr	Person	Generalisierung	Paraphrasen	Codes
1	P1	Agilität ist selbständige Teams, kurze Arbeitszyklen und zeitnahes Feedback	Agilität bedeutet selbstständige Teams, die in sich funktionieren und in sehr kleinen Zyklen arbeiten. Also wirklich MVP liefern und so kleine Pakete wie möglich zu releasen mit zeinahn Feedback durch die Stakeholder.	K1-1 Begriffsverständnis Agile SWE und agile Teams
2	P1	Kennzahlen in der Softwareentwicklung schwieriger als in anderen Teams zu erfassen.	Wir sind aktuell im Prozess KPIs einzuführen, um einen Fortschritt transparent zu machen und Planbarkeit zu gewährleisten. Diese sind in der SWE nicht so einfach zu gestalten und in anderen Teams einfacher.	K1-2 Metriken innerhalb der agilen SWE
3	P1	Produktqualitätsmetrik sind Bugs pro Release (Quantität und Schweregrad), Entwicklungsgeschwindigkeit beachten.	Zum Beispiel Bugs pro Release wäre eine Qualitäts-Kennzahl. Eine Null Bug Policy pro Release kann aber die Entwicklung extrem verlangsamen. Dies gilt es abzuwägen. Ziel ist es dem Kunden ein Produkt zu liefern mit dem er zufrieden ist.	K1-2 Metriken innerhalb der agilen SWE
4	P1	KPIs müssen situativ angepasst werden und dürfen nicht Alleinzweck werden.	KPIs müssen immer angepasst sein und der Einfluss auf die Arbeitsweise muss beachtet werden. Es kann vorkommen dann nur mehr auf die Erfüllung dieser KPIs dominiert und eigentlich nichts mit Mehrwert produziert wird.	K1-2 Metriken innerhalb der agilen SWE
5	P1	Wichtigste KPI intern Velocity und Story Points. Gefahr der wohlwollenden Schätzung	Die wichtigste interne KPI ist auf jedenfall intern die Velocity, da diese aber durch die Story Points definiert werden, kann es vorkommen das hier versucht wird diese KPI durch eine wohlwollende Schätzung zu meinen Gunsten als Team zu beeinflussen.	K1-3 Priorität/Präferenz bei Kennzahlen
6	P1	KPIs dienen der Transparenz des Outputs nicht der Überwachung. Einordnung und Unterstützung der Unternehmensziele durch KPIs	Ziel ist es intern den Output transparenter zu machen und nicht die Mitarbeiter zu überwachen. Die SWE Entwicklung soll in eine Richtung getrieben werden die sich den Unternehmenszielen unterordnet, es sollen alle an einen Strang ziehen und nicht ausbremsen	K1-3 Priorität/Präferenz bei Kennzahlen
7	P1	Interne Evaluationsprozess im Laufen. Produktqualität ist durch Kundenzufriedenheit und Bugreports stark beeinflusst.	Hier ist aktuell in der Firma ein gewisser Evaluationsprozess. Anbieten für die Produktqualität ist Anzahl der Bug Report/Quantifizierung und Schweregrad und zudem natürlich das Kundenfeedback aufs Produkt bezogen.	K2-1 Begriffsverständnis von Produktqualität in agiler SWE
8	P1	Kundenzufriedenheit soll durch Feedbackmöglichkeiten und Usability Test/Scores aktiv abgefragt werden.	Für die Kundenzufriedenheit bieten sich Sachen an wie Feedbackbuttons an. Außerdem Parameter wie Usability Scores der einen Soll/Ist Stand zum Beispiel im Handling der App aufzeigt.	K2-1 Begriffsverständnis von Produktqualität in agiler SWE

9	P1	Metriken müssen auf interne Eignung geprüft werden. Wichtigste für die Produktqualität wahrscheinlich die Anzahl der Bugs.	Hier muss betreffend der Eignung intern, vorhandenen Best Practices noch Recherche betrieben werden. Ein wichtiger Punkt ist auf jedenfall die Anzahl der gemeldeten Bugs mit einfließen zu lassen.	K2-2 Bekannte Metriken und Eignungsgrad - Produktqualität
10	P1	Bei neuen Features sollte der Mehrwert für den Kunden überprüft um hier die Produktqualität sicherzustellen.	Was auch sinnvoll ist die Überprüfung wieviele Interaktionen gibt es bei bestimmten neuen Features, um festzustellen wurde etwas mit Mehrwert entwickelt, genauso wie direktes Kundenfeedback. Der Detailgrad ist hier noch nicht definiert.	K2-2 Bekannte Metriken und Eignungsgrad - Produktqualität
11	P1	Intern werden aktuell nur die Bugs pro Release erhoben	Intern werden aktuell nur die Bugs erhoben, die innerhalb eines Sprint bzw. Releases entstehen.	K2-3 Einsatz von Metriken innerhalb der Firma - Produktqualität
12	P1	Teamproduktivität wird durch die Velocity ausgedrückt	Als Teamproduktivität wird vorallem die Velocity genutzt und ist bekannt. Die Vergleichbarkeit ist hier natürlich schwierig.	K3-1 Begriffsverständnis Teamproduktivität in SWE
13	P1	Mehrwert wird über Story Points definiert. Bugs und Regressionen erhalten keine Story Points	Mehrwert von Storys für die Kunden und die Firma wird primär nur über die Story Points ausgedrückt. Bugs oder Regressionen die keinen Value erzeugen liefern keine Story Points.	K3-1 Begriffsverständnis Teamproduktivität in SWE
14	P1	Da Story Points nur für Issues mit Value vergeben werden, kann Produktivität auf Story Points heruntergebrochen werden.	Dieser Value oder Mehrwert ist wirklich nur Entwicklung, die zum Kunden oder in der Infrastruktur für den Kunden geschieht. Daher kann Produktivität in diesem Fall auf Story Points heruntergebrochen werden.	K3-2 Bekannte Metriken und Eignungsgrad - Teamproduktivität
15	P1	Story Points können nur sehr schwer mit einem monetären Wert gekoppelt werden.	Eine monetäre Ziffer also welchen Umsatz liefert ein Story Point, der kann hier nicht getroffen werden. Dies ist möglicherweise eine Entwicklung für die Zukunft.	K3-2 Bekannte Metriken und Eignungsgrad - Teamproduktivität
16	P1	Interne Hauptkennzahlen sind Velocity und Story Points und die Berechnung in Durchschnittszeiträumen.	Hauptkennzahlen sind aktuell intern die Velocity und die Story Points. Diese werden auch über einen Durchschnittszeitraum mitberechnet, um hier Entwicklungen festzustellen. Zudem erkennt der Scrummaster die Leistungsbereitschaft der Teams	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.
17	P1	Aufgrund der Komplexität werden keine zusätzlichen Kennzahlen wie LoC, Laufzeiten, etc. gemessen.	Es gibt aufgrund der Komplexität keine Kennzahlen wie im Sales oder Customer Care betreffend Lines of Code, Laufzeitoptimierung, Codequality oder Code Guideline Adherence.	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.
18	P1	Als Zukunftsprojekt sollen auch Dauer von Request und Laufzeiten unter gleichbleibenden Bedingungen als Metrik eingeführt werden.	Es gibt intern Bestrebungen die Dauer von API Request zu überprüfen und in unterschiedlichen Szenarien bei gleichbleibenden Bedingungen zu analysieren als Teil der Produktivitätsmetrik eines Commits. Dies ist aber sehr komplex	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.

19	P1	Code Review sind ein essenzieller Bestandteil des Produktqualitätsprozess. Es gibt hier aber keine gemessenen Kennzahlen dazu.	Code Review ist ein essentieller Bestandteil des Qualitätsprozesses. Es sollte keine Zeile Code ohne Review deployed werden. Dies ist aber aktuell keine Kennzahl die aktiv genauso wie Qualität und Menge von Dokumentation die gemessen wird.	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.
20	P1	Roadmaps und Vorhersagbarkeit leben von Priorisierung und dem Austarieren zwischen Zeit/Qualität/Kosten.	Vorhersagbarkeit und die Roadmap lebt von der Priorisierung der großen Themen. Was dann zum Kunden darf muss hier gefiltert werden. Bei Features ist dies leichter vorhersagbar. Deadlines sind aber gefährlich da die Element Zeit/Qualität/Kosten mitspielen.	K4-1 Begriffsverständniss Vorhersagbarkeit in SWE
21	P1	Durch das Festsetzen von zwei Parameter wird einer variable, oft ist dies die Zeit.	Es ist wichtig zum Verstehen, dass wenn ich zwei dieser Elemente festsetze und unvorhergesehenes geschieht, dann muss ein anderer Parameter angepasst werden. Ich kann nun bei einer Deadline zum Beispiel dann Qualität oder Kosten opfern.	K4-1 Begriffsverständniss Vorhersagbarkeit in SWE
22	P1	Vorhersagbarkeit wird durch die Treffgenauigkeit des Story Point Commitment der Teams möglich	Für diese Vorhersagbarkeit ist die Treffgenauigkeit des Commitment zu den Story Points und umzusetzenden Stories am wichtigsten. Dies birgt aber die Gefahr in sich, dass Teams für Treffgenauigkeit Ihre Velocity opfern und der Gesamtoutput niedrig ist.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
23	P1	Vorhersagbarkeit für neue Features ist leichter als bei Wartung/Austausch von Legacy Code	Das Commitment sowie die Vorhersagbarkeit für neue Features ist deutlich einfacher als arbeiten innerhalb eines Legacy Code Bereiches.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
24	P1	Intern werden für die Vorhersagbarkeit das Commitment herangezogen und für die Erfüllung der Erwartungen die Definition of Ready und die Definition of Done.	Für die Vorhersagbarkeit der Arbeitsmenge wird das Commitment herangezogen. Für die Qualität und die Passgenauigkeit des Lieferobjektes die Definition of Ready/Refinement sowie die Definition of Done.	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit
25	P1	Erfüllungsgrad der Wünsche von Stakeholder ist an Domänen und Produktwissen der Entwickler gekoppelt.	Schwierig kann es sein die Parameter des Erfüllungsgrades der Wünsche der externen Stakeholder abzufangen. Dafür ist sowohl Fachdomänenwissen als auch spezielles Produktwissen bei allen Entwicklungsteams notwendig, damit die Featurefunktion klar ist.	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit
26	P1	Das Commitment und die das Erreichen dieses Commitment im Zeitverlauf wird auch gemessen	Es wird aktuell das Commitment und das Erreichen dieses Commitment im Zeitverlauf gemessen aber noch nicht als Kennzahl verwendet. Da hier auch das jeweilige Team internen sozial Mechanismen beachtet werden müssen.	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit

27	P2	Für die agile Softwareentwicklung gibt es wenige relevante Metriken und diese sind immer relativ niemals absolut zu sehen.	Betrachtung der Metriken aus Sicht des Consulter und CEO. Viele unterschiedliche Metriken die im agilen Coaching wichtig sind. In der Agilität eigentlich nur einige wenige die relevant sind und diese sind nicht absolut, sondern immer relativ.	K1-1 Begriffsverständnis Agile SWE und agile Teams
28	P2	Agile Softwareentwicklung orientiert sich an Prinzipien und Werten im Rahmen der Selbstorganisation, Selbststeuerung und des Selbstmanagement der agilen Teams.	Agile Softwareentwicklung orientiert sich immer an Werten und Prinzipien und welches Framework ist weniger entscheiden. Sondern dass diese Werte gelebt werden im Rahmen von Selbstorganisation, Selbststeuerung und Selbstmanagement der agilen Teams	K1-1 Begriffsverständnis Agile SWE und agile Teams
29	P2	Wichtigsten KPI sind die Velocity und die Story Points, sowie das Verhältnis zugesagten und gelieferten Story Points.	Zwei Metriken sind besonders wichtig zum einen die Velocity also in Story Points und die zweite Kennzahl, die kaum zu finden ist in der Literatur, das Verhältnis von zugesagten Story Points zu abgelieferten. Da dies die Schätzqualität des Teams abbildet.	K1-2 Metriken innerhalb der agilen SWE
30	P2	Intern wir nach Kanban gearbeitet, hier ist die Durchlaufzeit am wichtigsten.	Aufgrund der Firmengröße wird nach Kanban gearbeitet. Hier ist die Durchlaufzeit der Tickets von links nach rechts am wichtigsten. Also wenn ein Ticket aus dem Backlog in der Spalte Done ankommt.	K1-3 Priorität/Präferenz bei Kennzahlen
31	P2	Intern werden die Kanban-Spalten dem Branching und Release Modells angepasst	Die Spalten sind entsprechend des Prozesses modelliert und bilden das Branching- und Release-Modell ab. Hier wird nach Backlog, Ready, in Arbeit, Research, Develop und dann Master unterschieden entsprechend unseres internen Modelles.	K1-3 Priorität/Präferenz bei Kennzahlen
32	P2	Die Durchlaufzeit beginnt dann zu laufen, wenn ein Issue Ready ist und endet wenn durch den Continous Integration Prozess das Issue beim Kunden ist.	Die Durchlaufzeit beginnt ab wann ist es Ready. Es gibt dafür zwei Metriken die Cycle Time und die Lead Time. Sobald es im Master ist, ist es auch aufgrund des Continous Deployment Prozesses direkt am Kundensystem verfügbar.	K1-3 Priorität/Präferenz bei Kennzahlen
33	P2	Produktqualitäts ist das positive absolvieren mehrere Quality Gates mit mehreren Testzyklen (Unit,Integrations, User Acceptance)	Produktqualität bedeutet das ein Feature oder Arbeitspaket mehrere Quality Gates positiv absolviert hat entlang des Entwicklungsprozesses. Dafür gibt es mehrere Testzyklen wie Unit Tests, Integrationstest bis User Acceptance auf der Oberfläche.	K2-1 Begriffsverständnis von Produktqualität in agiler SWE
34	P2	Erfolgreiche Tests entscheiden, ob ein Feature im Kanbanprozess weiterwandern darf.	Diese erfolgreichen Tests sind das wichtigste Merkmal ob ein Feature von Research weiter in Develop darf oder in den Master. Erst wenn die Unittest und UI Test positiv absolviert wurden, darf es eine Spalte weiterwandern im Kanbanprozess.	K2-1 Begriffsverständnis von Produktqualität in agiler SWE

35	P2	Die verwendete Metrik dafür ist die Test Success Rate sowie durchgelaufene Test. Zudem erfolgt Linting, Code Coverage und statische Codeanalyse.	Als Metrik dient die Test Success Rate, durchgelaufene und erfolgreiche Test. Zudem wird auch die Code Coverage, statische Codeanalyse herangezogen. Dies sind aber nicht die Quality Gates. Linting wird zur Einhaltung von Coding Styles/ Guidelines genutzt.	K2-2 Bekannte Metriken und Eignungsgrad - Produktqualität
36	P2	Linting wird aggressiv für Einhaltung Coding Style/Guidelines genutzt. SonarQube für Analyse zum Refactoring des Backends	Linting wird sehr aggressiv eingesetzt, daher der Build bricht wirklich ab, wenn die Coding Guidelines und Coding Styles nicht eingehalten werden. Zudem wird für das Java Backend SonarQube genutzt, um Refactoring aufzuzeigen.	K2-2 Bekannte Metriken und Eignungsgrad - Produktqualität
37	P2	Backend hat intern die Stabilität der API-Kommunikation als Priorität. Frontend die erfolgreichen User Interface Tests.	Hier wird intern in Frontend und Backend unterschieden. Im Backend die Funktion der API-Kommunikation durch Unit Test mit dem Ziel der Stabilität. Im Frontend User Interface Tests und Client Tests die wichtigsten Use Cases abbilden, die ein User macht.	K2-3 Einsatz von Metriken innerhalb der Firma - Produktqualität
38	P2	Teamproduktivität sind die Arbeitspaket die ein Team fertigstellt. Diese können mit Business Value versehen werden (Gefahr der Demotivation)	Teamproduktivität ergibt sich aus den Arbeitspaketen, die ein Team in der Lage ist fertigzustellen. Dies könnte mit einem Business Value versehen werden und somit der Wert in gewissen Zeit errechnet. Hier besteht aber die Gefahr der Demotivation.	K3-1 Begriffsverständnis Teamproduktivität in SWE
39	P2	Wichtigste Kennzahl ist die Geschwindigkeit bis ein Kundenanliegen gelöst wird. (Cycle und Leadtimes)	Generell ist aber wahrscheinlich auch aufgrund unserer Größe die wichtigste Kennzahl die Geschwindigkeit bis eine Kunde von einem Anliegen/Problem eine Lösung bekommt. Also die bereits erwähnte Cycle Times und Lead Times.	K3-1 Begriffsverständnis Teamproduktivität in SWE
40	P2	In der Agilität ist Teamproduktivität immer selbst gesteuert und organisiert und damit schwer messbar.	Teamproduktivität messbar machen ist für einen Anhänger des agilen Manifests schwierig, da dies aus dem Team selbst heraus gesteuert und organisiert werden. Daher das Team hat die besten Leute und ist in der Lage eigenständig Probleme zu lösen.	K3-2 Bekannte Metriken und Eignungsgrad - Teamproduktivität
41	P2	Die einzige relevante Metrik bleibt was ist ein Team in der Lage zu leisten. Teamzusammenarbeit, Teamklima und Teamzufriedenheit tragen dazu bei.	Die einzige relevante Metrik bleibt damit was ist das Team in der Lage zu leisten und wie lang ist es in dieser Konstellation schonzusammen. Wichtig sind dann noch Teamklima, Teamzusammenarbeit und Teamzufriedenheit.	K3-2 Bekannte Metriken und Eignungsgrad - Teamproduktivität
42	P2	Performanceanalyse auf Personenebene widersprechen den agilen Manifest und Werten.	Generell ist es für mich nicht vereinbar mit dem agilen Manifest eine Performance Analyse auf Personenebene durchzuführen. Hier weigere ich mich persönlich das widerspricht all meinen agilen Werten.	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.

43	P2	Beim intern Kanbanprozess geht es um Dimensionierung und Bearbeitungsgeschwindigkeit der Issues.	Intern wird nach Kanban gearbeitet. Hier ist am wichtigsten die gleichgroße Dimensionierung der Tickets und die Geschwindigkeit.	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.
44	P2	Kanban erlaubt eine hohe Schlagkraft und kurze Umsetzung. Story Points und Release Planung können in kleinen Teams zu viel Overhead sein.	Dieser Einsatz von Kanban erlaubt bei kleinen Strukturen eine schnelle Schlagkraft und kurze Umsetzung. Durch Story Points und Release Planung entsteht immer Overhead.	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.
45	P2	Vorhersagbarkeit über Teams ist schwierig, es darf keinen absoluten Story Point Wert geben, dies erfolgt wenn eine Ebene darüber.	Vorhersagbarkeit ist eine Gretchenfrage, insbesondere bei Vergleichbarkeit über Teams hinweg. Es darf keinen absoluten Story Point Wert geben, falls muss dies in der Ebene darüber passieren.	K4-1 Begriffsverständnis Vorhersagbarkeit in SWE
46	P2	Velocity ist für die Vergleichbarkeit die beste Metrik. Dies muss immer nur Team zu Team verglichen werden. Abnahme oder Zunahme kann zwischen Teams verglichen werden	Für die Vergleichbarkeit ist immer die beste Metrik die Velocity. Damit können die Teamsprint im in Relation zu sich selbst gesetzt werden. Als Vergleich der Vorhersagbarkeit zwischen den Teams werden die Steigerung/Abnahmen der Storypoints herangezogen.	K4-1 Begriffsverständnis Vorhersagbarkeit in SWE
47	P2	Bei mehreren Teams sollten Vorhersagen auf Storyebene durch Teams erfolgen und darüber in den Epics durch den Produktmanager.	In einem Projekt als Consultant war es möglich diese Vergleichbarkeit über 10 Scrum Teams hinweg zu erreichen. In den Stories vom Team selbst und darüber in den Epics erfolgt die Schätzung durch Produktmanager. Diese ermöglicht Vergleichbarkeit.	K4-1 Begriffsverständnis Vorhersagbarkeit in SWE
48	P2	Für den Einsatz von Story Points wird eine Referenzstory benötigt. Diese hat den Wert drei. Alle anderen Schätzungen erfolgen in Relation dazu in Fibonacci Folge	Für den korrekten Einsatz und Definition von Story Points ist eine Referenzstory zu erstellen. Diese sollte eine gewisse Komplexität haben und 3 Story Points ausmachen. Dazu werden in klassischer Fibonacci Folge die anderen Stories in Relation gesetzt.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
49	P2	Die Schätzung der Referenzstory ist wichtig damit bei 2 Wochen Sprint und pro 5-7 Personen Team circa 20-25 Story Points geschafft werden.	Sollte die Referenzstory falschgeschätzt sein, werden die Story Points bei den Sprints zu hoch sein. Hier gilt als Richtwert 20-25 á zwei Wochen und 5-7 Leuten im Team. Höhere Story Points machen das schätzen in Fibonacci Folgen schwierig.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
50	P2	Für die Vorhersagbarkeit und Planung durch den Product Owner wird die Durchschnittsvelocity über 4-6 Sprint erfasst.	Für die Vorhersagbarkeit der Lieferungsmege gilt ein Durchrechnungszeitraum der Velocity von 4-6 Sprints. Damit kann ein Product Owner sehr gute Prognostizierbarkeit in die Zukunft extrapolieren und damit planen.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit

51	P2	Die wichtigsten Metriken sind die Velocity und ein geschätzter Backlog mit entsprechend definierten Stories	Schwankungen innerhalb der Velocity sind normal und deswegen gibt es einen Durchrechnungszeitraum von 4-6 Wochen. Daher die wichtigsten Metriken sind die Velocity und ein geschätzter Backlog mit entsprechend definierten Stories.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
52	P2	Grundlage für korrekte Schätzungen bildet ein gleichbleibendes Team.	Grundlage für korrekte Schätzungen ist ein gleichbleibendes Team und der Grad wie das Team aufeinander eingespielt ist. Sollten 90 Grad Drehungen erfolgen bzw. Kurven dann ist dies wie wenn von 0 gestartet wird.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
53	P2	Neue Produkte erfordern Backlog aufbau und 3 Sprints zum Erfahrungen machen, selbiges gilt für neue Teams.	Bei neuen Produkten oder sehr starken Veränderungen, ist es wichtig das Backlog erst aufzubauen. Eine Velocity anzunehmen und dann nach drei Sprints die Schätzungen hier an die Erfahrungen anzupassen. Dies kann auch mit neuen Teams passieren.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
54	P2	Für Frameworkänderungen sind eine Iteration Zero und Spikes/Forschungsaufträge mit Zeitlimit anzuwenden.	Bei einer Umstellung des Frameworks bzw. der verwendeten Programmiersprachen, bietet es sich an eine Iteration Zero zu machen. In dieser gibt es Spikes also Forschungsaufträge für das Team mit Zeitkontingent.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
55	P2	Für Spikes müssen in Sprint Story Point vorgesehen werden	Innerhalb eines Sprint wird dann eine gewisse Menge an Story Points für Spikes vorgesehen zum Beispiel 5 von 20. Diese Zeitmenge die verwendet werden darf gibt der Product Owner vor.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
56	P2	User Stories werden vom Team geschätzt, Spikes vom Product Owner	Daher es findet hier eine Umkehr statt. In normalen User Stories wird die Komplexität vom Team geschätzt, bei einem Spike wird dies durch den Product Owner übernommen.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
57	P2	Intern werden Qualitätskennzahlen und Quality Gates mit Cycle Time für die Vorhersagbarkeit genutzt.	Intern verwenden wir sehr stark Qualitätskennzahlen um die Quality Gates zu überwachen und steuern zu können. Diese Qualitätskennzahlen in Kombination mit der Cycle Time von ready bis Master liefert die Vorhersagbarkeit.	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit
58	P2	Vorhersagbarkeit sind selbständige Teams mit Fullstackentwicklern wichtig.	Für die Vorhersagbarkeit ist es wichtig, dass die Team selbständig sind und Fullstack und das gesamte Spektrum abdecken. Jeder hat zwar seinen Fokus aber muss eben alle Dinge können.	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit
59	P3	Scrum als Framework	Wir behaupten etwas mit agiler Softwareentwicklung zu machen zumindestens was Scrum-ähnliches	K1-1 Begriffsverständnis Agile SWE und agile Teams
60	P3	Ambivalenz zu Metriken	Ich stehe ambivalent zu Metriken und ihrer Sinnhaftigkeit	K1-1 Begriffsverständnis Agile SWE und agile Teams

61	P3	Im Vordergrund der agilen SW steht Eigenverantwortung der Teams, iteratives Vorgehen und Kundenfeedback.	Agile Software Entwicklung lehnt sich stark ans agile Manifest an, im Vordergrund steht die Eigenverantwortung der Teams. Was sie dann machen ist nicht essentiell, sondern iteratives Vorgehen und gemeinsame Entwicklung mit dem Kunden	K1-1 Begriffsverständnis Agile SWE und agile Teams
62	P3	Im Scrum Framework geht es um Story Point und wieviel Issues im Backlog sind.	Metriken sind abhängig was da nun genau darunter fällt. Wir arbeiten nach Scrum und hier werden die Issues mit Story Point bewertet und ein Sprint besteht aus einer gewissen Menge an Story Points. Wir achten aber auch wie viele Issues im Backlog sind.	K1-2 Metriken innerhalb der agilen SWE
63	P3	Hochschulen sind stärker auf fertige Produkte der StudentInnen fokussiert.	In Hochschulen geht es weniger um die Prozessbeurteilungen sondern mehr um fertiggestellte Produkte. Es ist hier auch eine höhere Eigenverantwortung beim Team als bei mir als Software Leiter.	K1-2 Metriken innerhalb der agilen SWE
64	P3	Kennzahlen dürfen nicht nur die Softwarequalität sein, sondern auch Time-to-Market	Als Kennzahlen nicht nur Qualität sondern auch der Prozess und hier die Basisgeschwindigkeit und die Time-to-Market. In der Softwarequalität das übliche: Lose Kopplung, sauber getrennte Schichten, wenig Spaghetticode, gewisse Menge an Tests.	K1-3 Priorität/Präferenz bei Kennzahlen
65	P3	Testcoverage ist abhängig von der Kritikalität des Bereiches. Viel Eigenverantwortung beim Team	Ich bin nicht der Verfechter von X Prozent Test Coverage sondern dies ist abhängig vom Bereich in der Software. Hier liegt viel Eigenverantwortung beim Team betreffend fehlerfrei zu sein, zu entscheiden ist der Bereich kritisch und nder Geschwindigkeit.	K1-3 Priorität/Präferenz bei Kennzahlen
66	P3	Produktqualität in Außensicht ist Bedienbarkeit, Mehrwert für den Kunden, fehlerfrei, Responsiveness.	Produktqualität ist getrennt in Außen- und Innensicht. Außensicht sind Bedienbarkeit, Auffindbarkeit der Funktionen, Mehrwert für Kunde, Firma, Responsiveness und einigermaßen fehlerfrei, Fehlermeldungen sollten handlungsorientiert sein	K2-1 Begriffsverständnis von Produktqualität in agiler SWE
67	P3	Produktqualität in der Innensicht ist Performance und noch wichtiger Wartbarkeit da diese sich direkt auf die Velocity auswirkt.	Die Innensicht der Produktqualität ist Wartbarkeit und Performance der Software. Essentiell ist aber die Wartbarkeit, weil diese direkt die Velocity, die Zeit für neue Feature und Möglichkeit sich auf neue Veränderungen bei Requirements auswirkt.	K2-1 Begriffsverständnis von Produktqualität in agiler SWE
68	P3	Wartbarkeit selbst nicht messbar. Code Qualität und Coding Style durch Tools stellvertretend.	Wartbarkeit per se lässt sich nicht messen. Daher muss stellvertretend etwas gefunden werden. Dafür gibt es eine Reihe von Tools und Möglichkeiten um Coding Style und Code Qualität zu forcieren.	K2-2 Bekannte Metriken und Eignungsgrad - Produktqualität
69	P3	Metriken Codequalität: Funktionen(Codelänge, Verschachtelungen), Anzahl	Typische Metriken für Codequalität wären: länge von Funktionen, zu viele Verschachtelungen, Funktionen in einer	K2-2 Bekannte Metriken und Eignungsgrad - Produktqualität

		von Funktionen pro Klasse, Indirektionen	Klasse, Codelänge von Funktionen und Indirektionen zum Beispiel in Java.	
70	P3	Tools sekundär, Prozess wichtiger(Pair Programming, Reviews)	Tools sind für die Kontrolle nur sekundär, hauptsächlich wird dies intern mit Prozessen, Reviews und Peer Programming abgefangen.	K2-3 Einsatz von Metriken innerhalb der Firma - Produktqualität
71	P3	Bei Tools wichtig: statische Typisierung, Codeformatter	Tools und Prozesse für Codequalität sind am wichtigsten: statische Typisierung besonders bei Sprachen ohne Typisierung. Dadurch werden viele Fehler vermieden und auch sanfter Druck zu Codingqualität erzeugt. Danach wohl dosiert Codeformatter	K2-3 Einsatz von Metriken innerhalb der Firma - Produktqualität
72	P3	Einsatz Linting wohl dosiert, SonarQube für Trends. Merge-, Pullrequests und Reviewanalyse für kurzfristige Codequalitätüberprüfung	Firmenintern werden ein sehr strenges Formatting Tool genutzt, Linting aber nur wohl dosiert aufgrund des möglichen Widerstandes und für langfristige Trends in der Komplexität SonarQube. Kurzfristig Merge-, Pull- Requests und Reviews.	K2-3 Einsatz von Metriken innerhalb der Firma - Produktqualität
73	P3	Reviews schwer messbar, Gibt viele dutzend andere Metriken	Möglichkeiten diese Prozesse wie Reviews zu messen sind schwierig, es gibt hier firmenintern bestimmte Prozesse diese führen aber zu keinen Managemententscheidungen und im Bezug auf Wartbarkeit gibt es neben den erwähnten tausende andere.	K2-3 Einsatz von Metriken innerhalb der Firma - Produktqualität
74	P3	Teamproduktivität ist mit Einschränkungen, Story Points pro Sprint	Teamproduktivität ist mit Einschränkungen wie viele Issues bzw. Story Points im Sprint erledigt wurden. Es gibt aber sehr viel mehr wie Architektur, Refinement, Requirements erfassen. Daher gibt es im Sprint keine harte Metrik. Sondern was geht pro Sprint.	K3-1 Begriffsverständnis Teamproduktivität in SWE
75	P3	Einschätzung durch Team selbst	Als Teamproduktivität geht es um die Einschätzung durch das Team selbst und der Velocity die ein Team regelmäßig erreicht	K3-1 Begriffsverständnis Teamproduktivität in SWE
76	P3	Wichtigste Kennzahl intern Story Points und glieferte Menge. Über 85% grün, 85 -60 grau, unter 60% rot	Wichtigste internen Kennzahlen für Teamproduktivität sind Story Points, hier ist es zudem wichtig wieviel der geschätzten Story Points sind am Ende vom Sprint fertig, über 85% sind ein guter Wert und grün unter 60% ist schlecht und im roten Bereich.	K3-2 Bekannte Metriken und Eignungsgrad - Teamproduktivität
77	P3	Velocity nur als Vergleichswert ob Wartbarkeit oder Qualität requirements abnimmt	Velocity über Story Points wird nur für den Vergleich herangezogen aber nicht um schneller zu werden sondern als Indikator ob Wartbarkeit abnimmt, Requirements schlechter aufgenommen werden, Störfaktoren von außen.	K3-2 Bekannte Metriken und Eignungsgrad - Teamproduktivität
78	P3	Für externe/interne Stakeholder außerhalb SWE nur Time-To-	Zu den internen und externen Stakeholdern zählen nur die Time-to-Market Zeit von	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.

		Market Zeit von Features wichtig	funktionsfähigen Features und wie oft diese fristgerecht oder eben nicht geliefert werden.	
79	P3	Teamrends werden anhand Smiley und Retrospektive am Ende des Sprints	Für die Erfassung des internen Trends im Team gibt es einen Stimmungsbarometer mit Smileys und die Retrospektiven am Ende des Sprints. Durch die interne Motivation der Teams ist hier sehr starke eigen Motivation vorhanden.	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.
80	P3	Feedbackgespräche innerhalb von 14 Tagen	Innerhalb von 14 Tagen erfolgen mit jedem Mitarbeiter persönliche Feedbackgespräche als Fixtermine um schnell auf Befindlichkeiten und Performance einzelner Teammitglieder reagieren kann.	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.
81	P3	Vorhersagbarkeit erfolgt anhand Backlog und der Menge an Story Point die pro Sprint geschafft werden	Als Vorhersagbarkeit in einer idealen Scrumwelt dient der Backlog und die Menge an Story Point pro Sprint die geschafft werden können. Daher ich kann immer sagen wie schnell ich den aktuellen Backlog eigentlich abarbeiten, dies ist aber nicht erreichbar.	K4-1 Begriffsverständniss Vorhersagbarkeit in SWE
82	P3	Vorhersagbarkeit von Größe Features und Arbeitspaketen abhängig. Mehr Größe führt zu mehr Unsicherheit	Vorhersagbarkeit ist von der größe der Features und der Arbeitspakete abhängig. Je größer diese werden desto höher ist die Unsicherheit in der Planung, da hier viele Parameter geschätzt werden müssen und die Komplexität steigt.	K4-1 Begriffsverständniss Vorhersagbarkeit in SWE
83	P3	Vorhersagbarkeit bei Performanceoptimierung sehr schwierig	Vorhersagbarkeit wird aber insbesondere bei Performanceoptimierung von Rechenzeit schwierig, hier ist oft zuerst die Frage ob dies theoretisch überhaupt möglich ist und praktisch bedeutet es wie viel nimmt man im kauf das Programme komplexer werden dafür	K4-1 Begriffsverständniss Vorhersagbarkeit in SWE
84	P3	Qualität Anforderungsaufnahme sehr wichtig(innerhalb Entwicklungsabteilung weniger)	Qualität der Anforderungsaufnahme sind generell wichtig außer es handelt sich intern um Issues die durch die Entwicklungsabteilung aufgenommen werden	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
85	P3	Definition of Ready und Refinement sehr wichtige Parameter	Für die korrekte Übernahme der Tickets ist sowohl das Refinement bei Tickets von externen Stakeholder als auch das Definition of Ready sehr wichtig. Die Definition of Done fungiert für die Entwickler beim Programmieren als Richtschnur.	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
86	P3	Intern Requirements Abklärung Product Owner und Lead Engineer. Teilweise sogar 2 Refinements	Requirements abklärungen müssen zuerst den Product Owner und dann bei größeren Features auch den Lead Engineer passieren. Es gibt ein manchmal zwei Refinements damit dies sauber im Definition of Ready umgesetzt wird.	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit

87	P3	Definition of Done enthält auch Wünsche der Stakeholder. 3/4 sind aber vom Team selbst.	Die Definition of Done sollte auf die Parameter und Wünsche der internen bzw. externen Stakeholder auch Rücksicht genommen werden. Bei internen vermischt dies leichter und kann in der Implementierung geklärt werden. 3/4 der DoD ist vom Team selbst.	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit
88	P3	Für gute Vorhersagbarkeit, ist Stabilität der Requirements wichtig. Roadmap für Veränderungen und Frühwarnsystem.	Für die Vorhersagbarkeit ist Stabilität der Requirements notwendig und nicht in unterschiedliche Richtungen gezerrt zu werden. Hier ist eine Roadmap sehr wichtig und hier aber auch Veränderungen festhalten und Abweichungen. Dies fungiert als Frühwarnung	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit
89	P3	Roadmap als Ankerpunkt. Tools für Vorhersagbarkeit sind Roadmap, Feedbackzyklen sowie Priorisierung.	Diese Roadmap ist der Ankerpunkt auf den sich alle einstellen können. Hier ist laufendes Feedback wichtig in kurzen Zyklen und korrekte Priorisierung damit die Vorhersagbarkeit weiterhin gegeben ist. Tools sind also Roadmap, Feedbackzyklen, Priorisierung	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit
90	P4	Geringer Bezug zu Agilität, stark Wasserfall orientiert. Continuous Integration und Continuous Deployment als QA Methoden	Aktueller Bezug ist noch gering und die internen Prozess Wasserfall geprägt aufgrund standalone on-premise Lösungen. Continuous Integration und Continuous Deployment dienen aber als QA Methoden. Änderungen erfolgen aber nur durch man. Releases	K1-1 Begriffsverständnis Agile SWE und agile Teams
91	P4	Agile SWE = selbständige Organisationseinheiten setzen in iterativen Prozessen mit Kundenorientierung in Eigenverantwortung um	Agile SWE ist ein leichtgewichtiger iterativer Prozess mit stärkerer kundenorientierung. Es gibt sehr viele verschiedene Konzepte. Agile Teams sind selbstständige Organisationseinheiten die Teile einer Software in Eigenverantwortung umsetzen können.	K1-1 Begriffsverständnis Agile SWE und agile Teams
92	P4	Kein durchgängiges agiles Framework. Starker interner Fokus auf Pflichtenheft	Da die interne Entwicklung eher der traditionellen Softwareentwicklung entspricht, gibt es hier keinen durchgängiges agiles Framework. Anforderungen werden mit sehr detailliert spezifizierten Pflichtenheften vom Kunden abgenommen vor Implementierung.	K1-2 Metriken innerhalb der agilen SWE
93	P4	Agilität im Projektbacklog und Projektmanagement. Burndowncharts zu Visualisierung	Agilität findet im Projektbacklog statt wo die Aktivitäten eines Projekt gesammelt werden. Agile Ansätze haben eher im Projektmanagement Verwendung. Es gibt für jedes Projekt Burndown Charts, Datenbasis sind dafür die Issues, es erfolgt Ist-Soll Abgleich	K1-2 Metriken innerhalb der agilen SWE

94	P4	Durchsatz als Kerngröße = Arbeitspakete pro Sprint. Erfassungen Störungen je Modul bei Auslieferung	Als wichtigste Kenngröße dient der Durchsatz, also wie viele Arbeitspakete pro Sprint abgearbeitet werden, hierbei unter der Berücksichtigung der Größe der Arbeitspakete. Es werden zudem die Störungen je Modul nach Auslieferung sowie Garantiezeit gewertet	K1-3 Priorität/Präferenz bei Kennzahlen
95	P4	Produktqualität = Kundenzufriedenheit, interne Wartungskosten und Wartbarkeit, Usability, Systemverfügbarkeit und Stabilität	Dies ist nicht nur auf agile SWE einzugrenzen. Produktqualität besteht aus verschiedenen Faktoren am wichtigsten sind Kundenzufriedenheit sowie laufende interne Wartungskosten außerdem Usability, Wartbarkeit, Systemverfügbarkeit und Stabilität.	K2-1 Begriffsverständnis von Produktqualität in agiler SWE
96	P4	Bugreports pro Modul und durchschnittliche Lösungszeit und Schweregrad als Metrik. angefallene Wartungskosten als monetärer Parameter	Spontan Anzahl Bugreports zu Modulen des Teams da diese am einfachsten greifbar und die durchschnittliche Lösungszeit sowie der Schweregrad des Fehlers. In dem Zusammenhang sind dann die angefallenen Wartungskosten wichtig.	K2-2 Bekannte Metriken und Eignungsgrad - Produktqualität
97	P4	Geschwindigkeit der Entwicklungszeit pro Modul, Verlangsamung abnehmende Produktqualität. Kundenzufriedenheit	Zusätzlich wäre die Geschwindigkeit von Entwicklungen im jeweiligen Softwaremodul wichtig, sollten kleine Adaptionen lange benötigen ist das ein Zeichen von abnehmender Produktqualität. Am wichtigsten auch hier die Kundenzufriedenheit.	K2-2 Bekannte Metriken und Eignungsgrad - Produktqualität
98	P4	Intern Bugreports pro Modul, im Reportzeitraum gelöst, Länge der ungelösten Zeit, Unit- und Integrationstestabdeckung	Innerhalb der Firma werden die Bugreports pro Modul, Bugs im Reportzeitraum die gelöst werden sowie wie lange Bugs ungelöst bleiben erfasst. Zusätzlich kommen Unit- und Integrationstest als Kennzahlen hinzu.	K2-3 Einsatz von Metriken innerhalb der Firma - Produktqualität
99	P4	Zeit bis zu Umsetzung gefordertes Feature. Story Points und Line of Codes möglicherweise falsche Anreize und verzerrend	Die Produktivitätsfrage ist schwierig zu beantworten. Generell sollte es die Zeit sein bis geforderte Features umgesetzt werden. Lines of Codes oder Story Points pro Iteration können aber falsche Anreize liefern und die Produktivitätsauswertung verzerren.	K3-1 Begriffsverständnis Teamproduktivität in SWE
100	P4	Intern nur Verhältnis Arbeitszeit von Kunden bezahlten Code und Code für Bugfixes, Projektbudget	Intern bewerten wir aktuell nur das Verhältnis an Arbeitszeit für neuen Code und der von einem Kunde bezahlt wird und Code für Bugfixes der durch uns selbst bezahlt wird. Innerhalb eines Projekt darf das geplante Budget nicht überschritten werden.	K3-1 Begriffsverständnis Teamproduktivität in SWE
101	P4	Verhältnis produktive zu nichtproduktiven Aufwänden als Kennzahl	Das Verhältnis zwischen produktiven und nichtproduktiven Aufwänden gibt uns Aussage ob der Entwicklungsprozess für dieses Modul in Ordnung ist. Wenn die Werte den geplanten Bereich verlassen müssen wir Ursachenforschung betreiben.	K3-2 Bekannte Metriken und Eignungsgrad - Teamproduktivität

102	P4	Theoretische Kennzahlen oft falsche Anreize. Interess Vorgehen produktiv zu nichtproduktiv funktioniert gut.	Generell eher der Meinung das viele Kennzahlen in der Theorie falsche Anreize liefern. Vorgehen bisher in der Firma funktioniert aktuell gut und führt zu akzeptablen Auswertungen.	K3-2 Bekannte Metriken und Eignungsgrad - Teamproduktivität
103	P4	Intern alle zwei Wochen Verhältnis produktiv/unproduktiv, monatlich Projektbudget	Es wird alle zwei Wochen das Verhältnis der produktiven und nicht produktiven Stunden aller Teams ausgewertet. Zudem überprüfen die Projektleiter monatlich die Projektbudgets.	K3-3 Einsatz von Metriken innerhalb der Firma - Teamprod.
104	P4	Vorhersagbarkeit = Zuverlässigkeit Liefertermin, Umfang, Qualität, Kosten vorab zu beurteilen	Vorhersagbarkeit besteht aus den Parametern Liefertermin, Umfang, Qualität und Kosten. Vorhersagbarkeit besteht daher aus der Zuverlässigkeit diese vier Parameter vorab zu beurteilen.	K4-1 Begriffsverständnis Vorhersagbarkeit in SWE
105	P4	Vergleichbarkeit gelieferten Arbeitspakete mit geplanten im Beobachtungszeitraum. Resultat = Liefertermin & Kostenabweichung	Es geht dabei um Vergleichbarkeit und gelieferten Arbeitspaketen im Beobachtungszeitraum und wie weit die geplanten Aufwände von tatsächlichen abweichen. Daraus resultieren Liefertermin und mögliche Kostenabweichungen	K4-2 Bekannte Metriken und Eignungsgrad - Vorhersagbarkeit
106	P4	Kosten intern schwierig da Fixpreis. Durchsatz und offene Issues Backlog für Vorhersagbarkeit des Liefertermins	Das Thema Kosten kann ein Problem darstellen, da wir mit einem Fixpreis auftreten. Wir zu langsam oder das falsche entwickelt, zahlt sich ein Projekt nicht aus. Anhand Durchsatz und offenen Punkte im Backlog können wir die Liefertermine gut vorhersagen.	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit
107	P4	Abweichung geplante/tatsächliche Arbeitspakete, Arbeitspakete im Backlog	Insbesondere die Abweichung zwischen geplanten und tatsächlichen Aufwänden sowie offene und geplante Arbeitspakete im Backlog werden gemessen.	K4-3 Einsatz von Metriken innerhalb der Firma - Vorhersagbarkeit

ABKÜRZUNGSVERZEICHNIS

Backlog	Ein Arbeitspaket innerhalb eines Ticketssystem 21
Liste mit priorisierten noch zu bearbeitenden Arbeitspaketen 21	KMUs
Branching	Kleine und Mittlere Unternehmen 1
Methode um mit einer gemeinsame zentrale Codebasis zusammenzuarbeiten 21	KPI
Bugs	Key Performance Indicators 20
Fehlerhafte Code bzw Programmteile 20	Linting
Builds	Als Linting wird das Analysieren von Code bezeichnet, um potenzielle Fehler und verdächtige Codeabschnitte zu kennzeichnen 22
Prozess in der Softwareentwicklung des Zusammenfügens von Programmteilen als auch die fertige Software-Version 7	QA
Burndowncharts	Qualitätassurance bzw. Qualitätssicherung 20
Darstellungsform bei der fertiggestellte Arbeitspakete schrittweise als absteigender Verlauf dargestellt werden 21	Ready
Continuous Integration	Bereit für die Entwicklung 21
Prozess des fortlaufenden Zusammenfügens von Komponenten zu einer Anwendung in der Softwareentwicklung 1	Release
Deployment	Veröffentlichung einer neuen oder überarbeiten Software 20
Softwareentwicklungsansatz bei dem Softwarefunktionen häufig durch automatisierte Bereitstellung dem Kunden zu Verfügung gestellt werden 1	Story Points
Issue	Masseinheit zur Aufwandsschätzung in der agilen Softwareentwicklung 21
	SWE
	Softwareentwicklung 20
	Velocity
	Maß zur Geschwindigkeitsmessung von Softwareentwicklungen 5

ABBILDUNGSVERZEICHNIS

Abbildung 1: Prozessmodell mit Effizienz- und Effektivität-Definition und Einordnung basierend auf (Sink & Tuttle, 1995) (Harald Beier, 2021)	3
Abbildung 2: Leistungsmessungs-Modell basierend auf (Sadowski & Zimmermann, 2019) ©Harald Beier (Harald Beier, 2021)	4
Abbildung 3: Agile Leitsätze mit Icons von Freepik und Eucalyp von www.flaticon.com , basierend auf (Kent et al., 2001) ©Harald Beier	6
Abbildung 4: Agile Prinzipien mit Icons von Freepik von www.flaticon.com , basierend auf (Kent et al., 2001) ©Harald Beier	6
Abbildung 5: „The stairway to heaven“ basierend auf (Olsson & Bosch, 2014) ©Harald Beier	7
Abbildung 6: Darstellung Dimensionen mit Icons von Freepik und Eucalyp von www.flaticon.com ©Harald Beier	31

TABELLENVERZEICHNIS

Tabelle 1: Zusammenfassung der einzelnen Parameter die unter Produktivität in der Literatur vorhanden sind, basieren auf (Harald Beier, 2021) ©Harald Beier	3
Tabelle 2: Verkürzung der Leistungsdimensionen auf drei Parameter nach (Padmini et al., 2015) und Zuordnung anderer Dimensionsebenen ©Harald Beier	5
Tabelle 3: Zusammenfassung der Stufen der „Stairway to Heaven“ (Olsson & Bosch, 2014, S. 16) ins Deutsch übersetzt.....	7
Tabelle 4: Metriken die in der Softwareentwicklung genutzt werden und der Nennungshäufigkeit in der wissenschaftlichen Literatur, basieren auf (Kupiainen et al., 2015).....	10
Tabelle 5: Kriterienbildung für gewichteten Katalog basierend auf (Kupiainen et al., 2015) ©Harald Beier	11
Tabelle 6: Gewichteter Kriterienkatalog mit zugeordneten Metriken basierend auf (Kupiainen et al., 2015) ©Harald Beier	13
Tabelle 7: Aufschlüsselung der Interviewpartner ©Harald Beier	15
Tabelle 8: Generalisierung der Paraphrasen der Kategorie K1 – Agile Softwareentwicklung ©Harald Beier	21
Tabelle 9: Generalisierung der Paraphrasen der Kategorie K2 – Produktqualität ©Harald Beier	23
Tabelle 10: Generalisierung der Paraphrasen der Kategorie K3 – Teamproduktivität ©Harald Beier.....	26
Tabelle 11: Generalisierung der Paraphrasen der Kategorie K4 – Vorhersagbarkeit ©Harald Beier.....	29
Tabelle 12: Erhobene Metriken durch die Interviews nach den drei Dimension und Nennung gegliedert ©Harald Beier	32
Tabelle 13: Um Ergebnisse der Auswertung erweiterter Gewichteter Kriterienkatalog ©Harald Beier	34

LITERATURVERZEICHNIS

- Bogner, A., Littig, B. & Menz, W. (2014). *Interviews mit Experten: Eine praxisorientierte Einführung. Lehrbuch*. Springer VS.
- Chew, B. W. (1988). No-Nonsense Guide to Measuring Productivity. *Harvard Business Review*(66), 110–115. <https://hbr.org/1988/01/no-nonsense-guide-to-measuring-productivity>
- Ertaban, C., Sarikaya, E. & Bagriyanik, S. (2018). Agile Performance Indicators for Team Performance Evaluation in a Corporate Environment. In *XP '18, Proceedings of the 19th International Conference on Agile Software Development: Companion*. Association for Computing Machinery. <https://doi.org/10.1145/3234152.3234156>
- Fenton, N. E. & Bieman, J. (2015). *Software metrics: A rigorous and practical approach. Chapman & Hall/CRC innovations in software engineering and software development*. CRC Press. <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=11002789>
- Fitzgerald, B. & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 123, 176–189. <https://doi.org/10.1016/j.jss.2015.06.063>
- Gabler, S. & Ganninger, M. (2010). Gewichtung. In C. Wolf & H. Best (Hrsg.), *Handbuch der sozialwissenschaftlichen Datenanalyse* (S. 143–164). VS Verlag für Sozialwissenschaften. https://doi.org/10.1007/978-3-531-92038-2_7
- Gallaba, K. (2019, 29. September - 2019, 4. Oktober). Improving the Robustness and Efficiency of Continuous Integration and Deployment. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (S. 619–623). IEEE. <https://doi.org/10.1109/ICSME.2019.00099>
- Harald Beier. (2021). *Software Metriken und Mechanismen: zu Bewertung agiler Teams* [Bachelorarbeit]. CAMPUS 02 Fachhochschule der Wirtschaft, Graz.
- Helfferrich, C. (2014). Leitfaden- und Experteninterviews. In N. Baur & J. Blasius (Hrsg.), *Handbuch. Handbuch Methoden der empirischen Sozialforschung* (S. 559–574). Springer VS. https://doi.org/10.1007/978-3-531-18939-0_39
- Jackson, M. (2000). *An analysis of flexible and reconfigurable production systems: An approach to a holistic method for the development of flexibility and reconfigurability*. Zugl.: Linköping, Univ., Diss., 2000. *Linköping studies in science and technology Dissertations: Bd. 640*. Univ.
- Jackson A. Prado Lima & Silvia R. Vergilio (2020). Test Case Prioritization in Continuous Integration environments: A systematic mapping study. *Information and software technology*, 121, 106268. <https://doi.org/10.1016/j.infsof.2020.106268>

- Kent, B., Mike, B., Arie van Bennekum & Alistair, Cockburn, Et al. (2001). *Manifest für Agile Softwareentwicklung*. <https://agilemanifesto.org/iso/de/manifesto.html>
- Kuckartz, U. (2018). *Qualitative Inhaltsanalyse: Methoden, Praxis, Computerunterstützung* (4. Aufl.). *Grundlagentexte Methoden*. Beltz Juventa. <http://www.beltz.de/de/nc/verlagsgruppe-beltz/gesamtprogramm.html?isbn=978-3-7799-3682-4>
- Kupiainen, E., Mäntylä, M. V. & Itkonen, J. (2015). Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies. *Information and software technology*, 62, 163. <https://doi.org/10.1016/j.infsof.2015.02.005>
- Mayring, P. (2015). *Qualitative Inhaltsanalyse: Grundlagen und Techniken* (12., überarbeitete Auflage). Beltz Verlag. <http://d-nb.info/1063369835/04>
- Meyer, A. N., Fritz, T., Murphy, G. C. & Zimmermann, T. (2014). Software developers' perceptions of productivity. In S.-C. Cheung, A. Orso & M.-A. Storey (Hrsg.), *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (S. 19–29). ACM. <https://doi.org/10.1145/2635868.2635892>
- Niederberger, M. & Wassermann, S. (Hrsg.). (2014). *Methoden der experten- und stakeholdereinbindung in der sozialwissenschaftlichen forschung*. Springer VS. <https://doi.org/10.1007/978-3-658-01687-6>
- Olsson, H. H. & Bosch, J. (2014). Climbing the “Stairway to Heaven”: Evolving From Agile Development to Continuous Deployment of Software. In J. Bosch (Hrsg.), *Continuous Software Engineering* (S. 15–27). Springer. https://doi.org/10.1007/978-3-319-11283-1_2
- Padmini, K. V. J., Dilum Bandara, H. M. N. & Perera, I. (2015). Use of software metrics in agile software development process. In A. G. B. P. Jayasekara (Hrsg.), *2015 Moratuwa Engineering Research Conference (MERCon 2015): Moratuwa, Sri Lanka, 7-8 April 2015* (S. 312–317). IEEE. <https://doi.org/10.1109/MERCon.2015.7112365>
- Rädiker, S. & Kuckartz, U. (2019). *Analyse qualitativer Daten mit MAXQDA: Text, Audio und Video. Lehrbuch*. Springer VS. <http://www.springer.com/>
- Rempel, P. & Mäder, P. (2015). Estimating the Implementation Risk of Requirements in Agile Software Development Projects with Traceability Metrics. In S. A. Fricker & K. Schneider (Hrsg.), *Lecture notes in computer science: Bd. 9013. Requirements engineering: foundation for software quality: 21st international working conference, REFSQ 2015, Essen, Germany, March 23 - 26, 2015 ; proceedings* (Bd. 9013, S. 81–97). Springer. https://doi.org/10.1007/978-3-319-16101-3_6
- Sadowski, C. & Zimmermann, T. (2019). *Rethinking Productivity in Software Engineering*. Apress; Imprint, Apress.

- Shah, S. M. A., Papatheocharous, E. & Nyfjord, J. (2015). Measuring Productivity in Agile Software Development Process: A Scoping Study. In *ICSSP 2015, Proceedings of the 2015 International Conference on Software and System Process* (S. 102–106). Association for Computing Machinery. <https://doi.org/10.1145/2785592.2785618>
- Shahin, M., Ali Babar, M. & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*, 5, 3909–3943. <https://doi.org/10.1109/ACCESS.2017.2685629>
- Sink, D. S. & Tuttle, T. C. (1995). *Planning and measurement in your organization of the future* (4. Dr). Industrial Engineering and Management Press.
- SonarSource S.A (Hrsg.). (2021). *sonarqube: Firmenhomepage*. <https://www.sonarqube.org/>
- Tangen, S. (2005). Demystifying productivity and performance. *International Journal of Productivity and Performance Management*, 54(1), 34–46. <https://doi.org/10.1108/17410400510571437>
- Tomasdottir, K. F., Aniche, M. & van Deursen, A. (2020). The Adoption of JavaScript Linters in Practice: A Case Study on ESLint. *IEEE Transactions on Software Engineering*, 46(8), 863–891. <https://doi.org/10.1109/TSE.2018.2871058>
- Wassermann, S. (2014). Das qualitative Experteninterview. In M. Niederberger & S. Wassermann (Hrsg.), *Methoden der experten- und stakeholdereinbindung in der sozialwissenschaftlichen forschung* (S. 51–67). Springer VS. https://doi.org/10.1007/978-3-658-01687-6_4
- Yingling Li, Junjie Wang, Yun Yang & Qing Wang (2020). An extensive study of class-level and method-level test case selection for continuous integration. *Journal of Systems and Software*, 167, 110614. <https://doi.org/10.1016/j.jss.2020.110614>
- Zampetti, F., Vassallo, C., Panichella, S., Canfora, G., Gall, H. & Di Penta, M. (2020). An empirical characterization of bad practices in continuous integration. *Empirical Software Engineering*, 25(2), 1095–1135. <https://doi.org/10.1007/s10664-019-09785-8>