

MASTERARBEIT

AUSWAHLKRITERIEN FÜR AGILE TIMEBOXED UND NON-TIMEBOXED ANSÄTZE

ausgeführt am



FACHHOCHSCHULE DER WIRTSCHAFT

Studiengang

Informationstechnologien und Wirtschaftsinformatik

Von: Akram El-Kormany

Personenkennzeichen: 1810320003

Graz, am 10. Dezember 2019

.....

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

Agile Entwicklungsmethoden haben schon seit vielen Jahren ihren Einsatz in der Wirtschaft gefunden. Unternehmen haben den Bedarf von solchen agilen Entwicklungsprozessen entdeckt und haben schnelle Maßnahmen getroffen, um weiter am Markt zu existieren und die Kundenwünsche mit Hilfe von agilen Prozessen zu erfüllen. Jedoch gibt es einen wesentlichen Faktor, der alle agilen Entwicklungsprozesse betrifft, dieser Faktor ist die Zeit. Die Zeit spielt eine wesentliche Rolle für alle Auftraggeber und Auftragnehmer. Unter den agilen Entwicklungsprozessen gibt es zwei unterschiedliche Ansätze, timeboxed und non-timeboxed. Da stellt sich die interessante Frage: „Welche Auswahlkriterien sind entscheidend für agile Timeboxed und Non-timeboxed Ansätze?“, mit der sich diese Arbeit beschäftigt. Aufgrund dessen, dass es in der vorhandenen Literatur kaum Antworten auf diese Fragestellung gibt, wurden im Rahmen dieser Masterarbeit mithilfe der qualitativen Inhaltsanalyse „Interview“, einige Experten Interviews durchgeführt, um diese Frage zu beantworten. Die Ergebnisse dieser Arbeit stellen einen Software-Manager einige Auswahlkriterien zur Verfügung, anhand dessen entschieden werden kann, ob man einen Timeboxing oder einen Non-Timeboxing Ansatz bevorzugt.

ABSTRACT

Agile development methods have been used in business for many years. Companies have discovered the need for such agile development processes and have taken swift action to continue to exist in the marketplace and to meet customer needs through agile processes. However, there is an essential factor that affects all agile development processes, this factor is time. Time plays an essential role for all clients and contractors. Among the agile development processes, there are some that are timeboxed and others that are non-timeboxed. This raises the interesting question: "Which selection criteria are crucial for timeboxed and non-timeboxed approaches?", which is the subject of this master's thesis. Due to the fact that there are hardly any answers to this question in the existing literature, interviews were conducted in the context of this master thesis with the help of the qualitative content analysis "Interview", in order to answer this question. The results of this master's thesis provide a software manager with a few selection criteria that can be used to decide whether to apply a timebox or a non-timeboxing approach.

INHALTSVERZEICHNIS

1	Einleitung.....	1
1.1	Motivation	1
1.2	Hypothese	2
1.3	Methodik.....	2
1.4	Fragestellung	3
1.5	Ziel der Arbeit	3
2	Grundlagen.....	4
2.1	Klassische Vorgehensmodelle.....	4
2.2	Agilität.....	5
2.2.1	Werte	7
2.2.2	Das agile Manifest.....	9
2.2.3	Agile Prinzipien.....	12
2.3	Agile Entwicklungsprozesse	15
2.3.1	Scrum.....	15
2.3.2	Extreme Programming.....	18
2.3.3	Kanban.....	18
2.4	Timeboxed versus Non-timeboxed.....	20
2.5	Zusammenfassung	21
3	Timeboxing.....	22
3.1	Meilenstein und Timebox.....	22
3.2	Timeboxing und das Pareto-Prinzip	22
3.3	Vorteile Timeboxing	23
3.4	Vorteile Non-timeboxing.....	24
3.5	Zusammenfassung	25
4	Forschungsteil: Auswahlkriterien für Timeboxed und Non-timeboxed Ansätze	26
4.1	Vorgehensweise.....	26
4.1.1	Begründung der Forschungsmethoden.....	26
4.1.2	Datenerhebung mittels Interviews	27
4.1.3	Datenauswertung mittels Qualitativer Inhaltsanalyse	30

4.1.4	Vorgehensweise bei der Kategorienbildung.....	31
4.2	Auswertung und Interpretation.....	33
4.2.1	Kategorie: Überblick über das Projekt	33
4.2.2	Kategorie: Psychische Effekte.....	34
4.2.3	Kategorie: Innovation.....	35
4.2.4	Kategorie: Teamsetting	36
4.2.5	Kategorie: Velocity und Iterationsdauer	39
4.2.6	Kategorie: Go Live	40
4.2.7	Kategorie: Projektart	41
4.2.8	Kategorie: Qualität	43
4.2.9	Auswertung und Interpretation aller Kategorien.....	44
4.3	Zusammenfassung	45
5	Zusammenfassung und Ausblick	47
6	Abbildungsverzeichnis	50
7	Tabellenverzeichnis	51
8	Abkürzungsverzeichnis	52
9	Quellenverzeichnis.....	53
10	Literaturverzeichnis	55
11	Anhang.....	I
	Interview-Transkripte	I
	Befragter 1	I
	Befragter 2	XIII
	Befragter 3	XXVI
	Befragter 4	XXXVI
	Befragter 5	XLVII
	Befragter 6	LIV
	Kodierung	LXIII
	Kategorie-Bildung Interview 1	LXIII
	Kategorie-Bildung Interview 2	LXXVI
	Kategorie-Bildung Interview 3	LXXXII
	Kategorie-Bildung Interview 4	LXXXVIII
	Kategorie-Bildung Interview 5	XCIII

Inhaltsverzeichnis

Kategorie-Bildung Interview 6	C
Endgültiges Kategoriensystem	CVII

1 EINLEITUNG

Viele Unternehmen haben den Bedarf von agilen Entwicklungsprozessen entdeckt und haben schnelle Maßnahmen getroffen, um weiter am Markt zu existieren und die Kundenwünsche mit Hilfe von agilen Prozessen zu erfüllen. Trotz der viel existierenden agilen Entwicklungsansätze wie „Kanban“, „Extreme Programming“, „Crystal“, „Scrum“, „Adaptive Software Development“ und vieles mehr, steht der agile Entwicklungsprozess „Scrum“ in der Software-Branche im Vordergrund. Jedoch gibt es einen wesentlichen Faktor, der all diese agilen Entwicklungsprozesse betrifft, dieser Faktor ist die Zeit. Die Zeit spielt eine wesentliche Rolle für alle Auftraggeber und Auftragnehmer, sei es eine Lieferung, ein Deployment oder ein Service. Manche Aufträge müssen in einem gewissen Zeitrahmen erledigt werden (timeboxed), andere wiederum haben keinen Zeitdruck für die Lieferung des Endprodukts oder können gar nicht einschätzen, wie lange ein solches Projekt dauern kann (non-timeboxed). Nun kommt die Frage auf: Welche Auswahlkriterien sind entscheidend für Timeboxed und Non-timeboxed Ansätze?

1.1 MOTIVATION

Es sind verschiedene Faktoren, die mich motivierten, mich mit diesem Thema auseinanderzusetzen. Zum einen habe ich durch viele Diskussionen über die derzeitigen Prozesse im Unternehmen, in dem ich tätig bin, verfolgt, wo die Frage aufgeworfen wurde, welche agile Prozesse für welche Projekte am geeignetsten sind und vor allem welche Auswahlkriterien entscheidend für Timeboxed und Non-timeboxed Ansätze sind. Zum anderen habe ich im Rahmen meines Masterstudiums „IT & Wirtschaftsinformatik“ an der FH Campus02 in Graz immer wieder in verschiedenen Veranstaltungen und Vorträge von diesem Problem gehört, dass Unternehmen meistens auf Scrum zurück greifen obwohl, Scrum gerade nicht wirklich für die Situation und die Arbeitsweise des Unternehmens am passendsten ist.

1.2 HYPOTHESE

In dieser Arbeit wird die These vertreten, dass sehr viele Unternehmen meistens nicht wissen, welche agile Entwicklungsprozesse sich für ihren Zweck eignen, bzw. sie nicht wissen unter welchen Umständen Timeboxed bzw. Non-timeboxed Ansätze sich am besten eignen.

1.3 METHODIK

Zur Überprüfung dieser These werden mit der vorliegenden Arbeit zwei Ansätze verfolgt:

1. Analyse von Timeboxed und Non-timeboxed Ansätze und anschließend Auswahlkriterien für beide Arten überlegt.
2. Untersuchung anhand von Interviews mit agilen Expertinnen und Experten, welche Auswahlkriterien es ihrer Erfahrung nach für agile Timeboxed und Non-timeboxed Ansätze gibt, um einige Kriterien auszuhändigen, die entscheidungsrelevant sind.

Entsprechend der Verfolgung von zwei aufeinander aufbauenden Ansätzen für die Überprüfung der hier vertretenen These, gliedert sich die vorliegende Arbeit in zwei Teile.

Mit dem **ersten, theoretischen Teil** wird das Ziel verfolgt, ein Grundverständnis von der Agilität und der Grundidee des agilen Entwicklungsprozesses zu geben. Danach werden die einzelnen agilen Entwicklungsprozesse in Timeboxed und Non-timeboxed eingeteilt und anschließend Auswahlkriterien für beide Arten überlegt.

Mit dem **zweiten Teil der Arbeit** wird das Ziel verfolgt mittels explorativ ausgerichteter, empirischer Untersuchungen die Realität des Einsatzes von Timeboxed und Non-timeboxed Ansätze in der Praxis herauszufinden. Dementsprechend werden Interviews mit agilen Expertinnen und Experten durchgeführt, wo die Expertinnen und Experten Auswahlkriterien für agile Timeboxed und Non-timeboxed Entwicklungsprozesse angeben.

Anhand der theoretischen Ausführungen, sowie der Ergebnisse der empirischen Untersuchung soll überprüft werden, ob die hier vertretene These, dass sehr viele Unternehmen meistens nicht wissen, unter welchen Umständen Timeboxed bzw. Non-timeboxed, am geeignetsten ist.

1.4 FRAGESTELLUNG

Dieser Masterarbeit beschäftigt sich mit der Frage: Welche Auswahlkriterien sind entscheidend für Timeboxed und Non-timeboxed Ansätze?

1.5 ZIEL DER ARBEIT

Zusammenfassend lässt sich die Zielsetzung dieser Arbeit wie folgt darstellen: Die Überprüfung der These, dass sehr viele Unternehmen meistens nicht wissen, welche agilen Entwicklungsprozesse sich für ihren Zweck eignen, bzw. sie nicht wissen unter welchen Umständen welche Methode sich am besten eignet und dementsprechend Entscheidungskriterien festzulegen anhand dessen man sich für agile Timeboxed oder Non-timeboxed Ansätze entscheiden kann.

2 GRUNDLAGEN

2.1 KLASSISCHE VORGEHENSMODELLE

In der Softwareentwicklung gibt es verschiedenen Modelle, um die Anforderungen des Kunden umzusetzen. Einer der ersten existierenden Modelle ist das so genannte klassische Vorgehensmodell, auch bekannt als Wasserfall Modell. In diesem Modell werden die typischen Entwicklungsphasen beginnend bei den Anforderungen bis zu Wartung durchgeführt. Der wesentliche Punkt beim klassischen Wasserfall Modell ist, dass die Phasen von oben nach unten abgearbeitet werden, das heißt jede Phase muss abgeschlossen werden, bevor mit einer weiteren Phase begonnen wird. Schließt man also eine Phase ab und geht man zur nächsten Phase, so kann man sehr schwer zu der vorherigen Phase wieder zurück gehen und schon gar nicht zur vorvorherigen Phase. Somit ist spätes Reagieren nicht mehr möglich und eine exakte, genaue Planung notwendig.

Andererseits kann nicht ausgeschlossen werden, dass der Kunde schlussendlich nach der Auslieferung der Software doch nicht zufrieden ist und sich die Funktionalitäten ganz anders vorgestellt hat. Dazu kommt noch, dass es eine starke Abhängigkeit der Abteilungen gibt und sich dadurch das Projekt in die Länge zieht, da die Abteilungen nicht parallel arbeiten können. Das bedeutet, dass Abteilung „Testing“ beispielsweise auf Abteilung „Implementierung“ warten muss, um mit der Arbeit beginnen zu können, denn jede Phase muss abgeschlossen werden bevor es in die nächste Phase geht. Durch diesen Nachteil bekommt der Kunde zu einem sehr späteren Zeitpunkt, nämlich nach Abschluss des Projekts, die Endlösung präsentiert.

Jedoch bringt dieses Modell einen Vorteil mit sich, die Planungssicherheit. Durch eine genaue Struktur und eine präzise Planung kann bei gewissen Projekten die Umsetzung sehr gut funktionieren. Jedoch nur wenn die Projektanforderungen konstant sind und keine regelmäßigen Feedbackschleifen benötigt werden, was bei den meisten Projekten nicht der Fall ist. (vgl. Royce, 1970, S. 1-9)

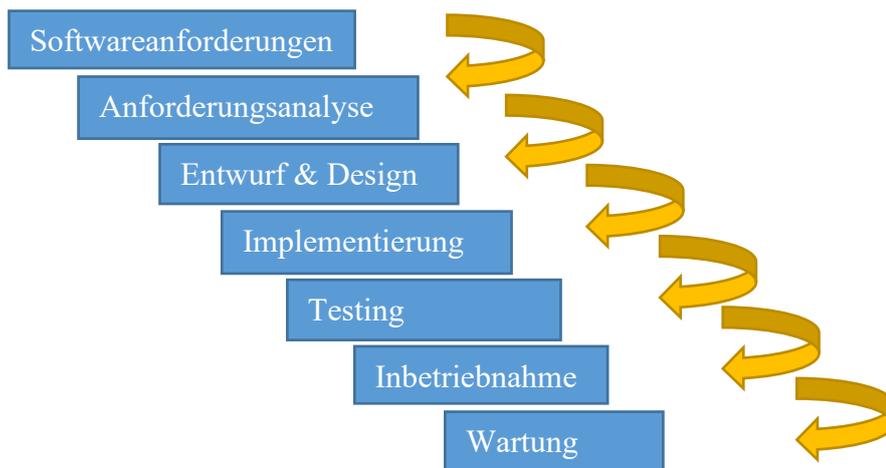


Figure 1: Phasen des klassischen Wasserfall Modell (in Anlehnung an Riehle, 2006)

2.2 AGILITÄT

Sieht man sich das klassische Wasserfall Modell an, so erkennt man, dass es einige Nachteile mit sich bringt. Deswegen hat man versucht durch die Entwicklung von agilen Vorgehensmodellen diese Nachteile zu beseitigen. Aber was genau bedeutet agil?

Agil kommt vom lateinischen „Agilis“ (lat.)“ und bedeutet so viel wie beweglich, tätig, rührend. (vgl. Diehl, o. J.).

Der Begriff „agil“ hat schon in den früheren Jahren 1967 Gebrauch gefunden. Allerdings wurde der Begriff Ende des 20. Jahrhunderts in der Produktion sehr häufig verwendet, um auf erwartete und unerwartete Marktänderungen und Bedürfnisse schnell reagieren zu können. Dieser Begriff wurde allerdings in der Informatik, speziell im Software-Entwicklungsbereich in den letzten Jahren verwendet um flexibel und darüber hinaus proaktiv, antizipativ und initiativ zu agieren und notwendige Veränderungen bei der Softwareentwicklung einzuführen.

„Agilität ist die Fähigkeit der Informationsverarbeitungsfunktion eines Unternehmens auf wechselnde Kapazitätsansprüche sowie veränderte funktionale Anforderungen sehr schnell (möglichst in Echtzeit) zu reagieren sowie die Möglichkeiten der Informationstechnologie derart nutzen zu können, dass der fachliche Handlungsspielraum des Unternehmens erweitert oder sogar neugestaltet wird.“ (vgl. Nissen 2013, S. 3)

Agile Vorgehensmodelle sind speziell dafür gedacht, flexibel auf Veränderungen und Change-Requests seitens der Kunden zu reagieren. Dabei werden dem Kunden zu jedem beliebigen Zeitpunkt gewünschte Anforderungen geliefert. Eines der wichtigsten Ziele der agilen Prozesse ist, alle Arten von möglichen Risiken, welche jederzeit auftreten können, frühzeitig zu erkennen und diesen entgegenzuwirken. Agile Prozesse sind deshalb auch so besonders, weil der Ansatz sowohl iterativ, als auch inkrementell ist. Es bietet einen geschützten Rahmen um Feedback möglichst früh einzubringen und dadurch die Projektsituation aus dem Gelernten zu verbessern. Durch die kurzen Entwicklungszyklen bilden sich die regelmäßigen Feedbackschleifen am Ende jedes Zyklus.

Die bekannten klassischen Phasen aus dem sequenziellen Wasserfall Modell wie beispielsweise: Design, Implementierung und Testing, existieren genauso im inkrementellen Ansatz. Dabei unterscheidet sich der klassische Ansatz vom agilen Ansatz dadurch, dass die Phasen nicht sequenziell, sondern parallel ausgeführt werden. Dadurch kann parallel zur Implementierung, getestet werden, um die erlangten Fehler zu bearbeiten und so kann auch, falls die Implementierung Auswirkungen auf das Design hat, das Design unmittelbar adaptiert und verbessert werden. (vgl. Eckstein 2004, S. 11-13)

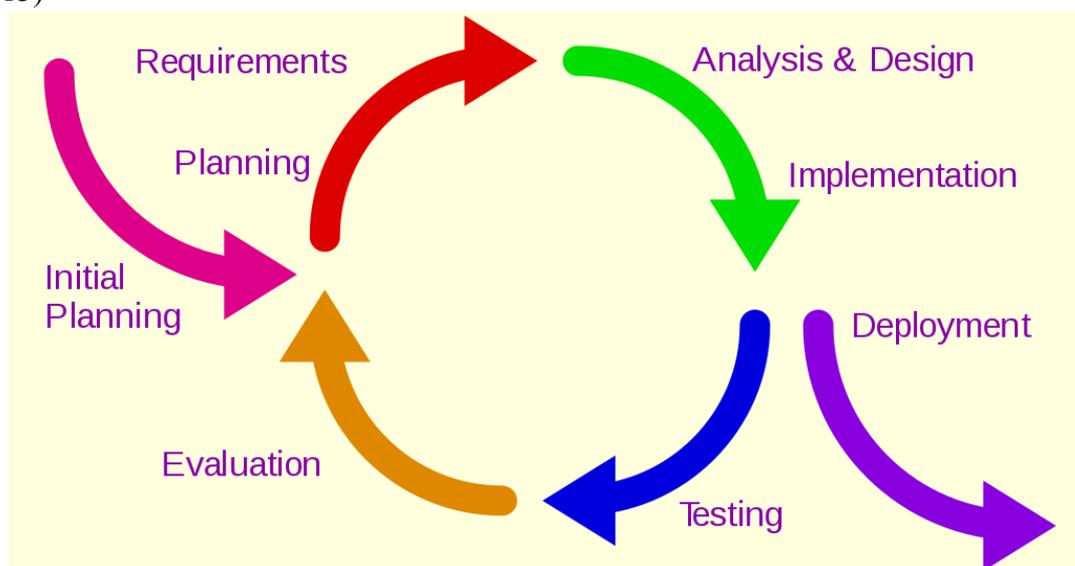


Figure 2: Agiler Entwicklungsprozess (o. V., 2018)

2.2.1 WERTE

2.2.1.1 KOMMUNIKATION

Die Kommunikation ist ein wesentlicher Wert des agilen Modells. Dieser Wert ist deshalb so wichtig, weil durch die Menge an Dokumentationen, Artefakten wie Lastenheften oder Pflichtenheften manchen in den Sinn kommt, dass der Verzicht auf die zwischenmenschliche Kommunikation ein Weg wäre. Agile Entwicklungsmethoden sehen die direkte Kommunikation zwischen allen Projektbeteiligten als notwendig. Somit sollen sowohl Anwenderinnen und Anwender ihre Bedürfnisse kommunizieren und sich untereinander über diese Bedürfnisse einig sein, aber auch die Kundenmanagerinnen und Kundenmanager müssen mit den Anwenderinnen und Anwendern über die Ziele sprechen und sich auf diese einigen. Schlussendlich müssen auch die Entwicklerinnen und Entwickler mit den Anwenderinnen und Anwendern und den Kundenmanagerinnen und Kundenmanagern in Absprache sein, um über Entscheidungen, Probleme und gewünschte Lösungen einen Überblick zu haben. Aber genau so müssen auch die Entwicklerinnen und Entwickler unter sich viel kommunizieren, um die Zusammenarbeit bei der Entwicklung der gewünschten Software mit den entsprechenden Zielen zu vereinfachen. Mit der Kommunikation ist hierbei eine direkte, offene und ehrliche Kommunikation, in der über alles gesprochen wird, ohne unangenehme Themen auszulassen.

2.2.1.2 EINFACHHEIT

Ein weiterer Wert des agilen Vorgehens ist die Komplexität der Softwareentwicklung, die dadurch entsteht, dass verschiedene Berufsgruppen zusammenarbeiten, dass Teammitglieder unterschiedliche Qualifikationen haben, dass Teams unterschiedliche Größen haben, dass die Anforderungen, das System und die zu verwendenden Komponenten zu komplex sind, dass die Teamziele und Teamspezialisierung unterschiedlich sind, zu vereinfachen. Genau weil die Softwareentwicklung in sich schon so eine Komplexität mit sich bringt, versucht man mit agilen Modellen, die Komplexität auf technische, organisatorische und methodische Ebene zu vereinfachen.

Somit soll einerseits, um die technische Komplexität zu vereinfachen, nur das eingesetzt werden, was wirklich gebraucht und benötigt wird. Die Architektur bzw. der Lösungsansatz sollen so einfach wie möglich gehalten werden, um später Erweiterungen und Änderungen durchführen zu können. Genauso sollen in der Organisation nicht viele unnötige Rollen vorhanden sein, die den ganzen Prozess komplexer machen. So sollen die Entwicklerinnen und Entwickler bzw. die Anwenderinnen und Anwender oder die Kundenmanagerinnen und Kundenmanger, das organisieren was sie können, ohne dafür unnötige Rollen zu erschaffen. Methodisch soll das Vorgehen so einfach wie möglich gehalten werden, sodass jede Projektbeteiligte und jeder Projektbeteiligter genau weiß wie der Prozess aussieht, es versteht und auch lebt. Die unterschiedlichen Methoden im agilen Modell sollen nicht für unnötige Komplexität sorgen, sondern wenn die Projektsituation es anbietet, so soll man die agile Methode noch einfacher leben als sie eigentlich vorgesehen ist.

2.2.1.3 FEEDBACK (RÜCKKOPPLUNG)

Dieser wichtige agile Wert ist in Zusammenhang mit der Kommunikation zu verstehen und zu leben. Darüber hinaus setzen unterschiedliche agile Methoden auf Feedbackschleifen, um die zeitnahe Projektsituation zu bewerten bzw. zu evaluieren und um die Kommunikation zwischen den Projektbeteiligten zu verbessern.

2.2.1.4 MUT

Ein weiterer Wert ist Mut, der ziemlich oft in der Softwareentwicklung benötigt wird, vor allem, wenn das Projekt, der Prozess oder jegliches in die falsche Richtung geht. Die Softwareentwicklung braucht mutige Menschen, die aufstehen und offen, ehrlich und direkt sagen, so können wir nicht weiterarbeiten, es benötigt eine Änderung. Denn nicht nur positives Feedback ist gefragt im agilen Vorgehensmodell, sondern genauso negatives Feedback, um zeitnah Probleme zu verbessern.

2.2.1.5 RESPEKT

Der letzte wichtige Wert des agiles Vorgehensmodell ist Respekt. Ein stetiger respektvoller Umgang zwischen allen Projektbeteiligten ist nicht nur vorteilhaft, um ein

sauberes und angenehmes Arbeitsklima zu gewährleisten, sondern ist genauso notwendig, um ein erfolgreiches Projekt auf die Beine zu stellen. Ohne einen respektvollen Umgang zwischen allen Beteiligten ist eine offene und ehrliche Kommunikation gar nicht möglich. Somit müssen die Anwenderinnen und Anwender, die Kundenmanagerinnen und Kundenmanager und die Entwicklerinnen und Entwickler bzw. das Entwicklungsteam sich gegenseitig respektieren, aber auch Team intern. So müssen beispielweise die Entwicklerinnen und Entwickler sich untereinander respektieren, um produktiv zusammenarbeiten zu können, in dem keiner ausgeschlossen wird und zum Außenseiter erklärt wird. Zusätzlich muss man sein Gegenüber verstehen, bevor man Feedback gibt, um das Feedback so objektiv und passend wie möglich zu gestalten und weiterzugeben. (vgl. Wolf-Gideon 2008, S. 10-12)

2.2.2 DAS AGILE MANIFEST

Im Jahr 2001 trafen sich viele Vertreterinnen und Vertreter von verschiedenen agilen Methoden mit dem Ziel ein Manifest für alle agilen Prozesse und Methoden aufzustellen, der auf Praxiserfahrungen beruht. Unter ihnen waren: Kent Beck, Mike Beedle, Martin Fowler, Jim Highsmith und Kent Schwaber, Vertreter von agilen Methoden wie Scrum, Extreme Programming, Crystal und Feature Driven Development. Das Besondere an diesem Manifest ist, dass es von Praktikerinnen und Praktiker erstellt wurde und von Praktikerinnen und Praktiker verwendet werden soll. So ist es nicht nur eine Theorie, die sich in einem Buch niedergeschrieben befindet, sondern praxisorientierte Erfahrungswerte.

Mit dem Titel: „Wir entdecken bessere Wege, Software zu entwickeln, indem wir Software entwickeln und anderen dabei helfen“ haben die Vertreterinnen und Vertreter 4 Leitsätze entwickelt:

1. **Individuen und Interaktionen** stehen über Prozessen und Werkzeugen
2. **Funktionierende Software** steht über einer umfassenden Dokumentation

3. **Zusammenarbeit mit dem Kunden** steht über der Vertragsverhandlung
4. **Reagieren auf Veränderung** steht über dem Befolgen eines Plans

Das heißt, obwohl die Werte auf der rechten Seite wichtig sind, werden die Werte auf der linken Seite höher geschätzt.

Individuen und Interaktionen stehen über Prozessen und Werkzeugen bedeutet, dass die Individuen, die in einem Projekt beteiligt sind und ihre Interaktionen und Zusammenarbeit im Team wichtiger sind oder höher bewertet werden, als Werkzeuge oder Prozesse, seien diese Prozesse auch agile Prozesse. Im Fokus steht der Mensch, der das Softwareprojekt nach vorne bringt und das Projekt zu einem Erfolg werden lässt. Dabei sind Prozesse und Werkzeuge nur ein Hilfsmittel, die Menschen zum Ziel verhelfen. Nicht umgekehrt, dass die Menschen nur dafür da sind, um Prozesse einzuhalten und Werkzeuge anzuwenden.

Funktionierende Software steht über einer umfassenden Dokumentation bedeutet, dass eine umfangreiche umfassende Dokumentation dem Kunden im Endeffekt nichts bringt, wenn die eigentliche Software nicht funktioniert. Deshalb steht die Software stets im Vordergrund. Nur anhand laufender Software wird der Projektfortschritt nachgewiesen. Es ist eine viel stärkere Aussage für den Kunden zu behaupten, dass 20% der Features am System implementiert sind und zur Verwendung stehen, anstatt zu behaupten, dass 80% der Features in den Dokumentationen niedergeschrieben sind. Somit dienen alle Aktivitäten, auch die Dokumentation dazu lauffähige Software zu entwickeln und diese zu liefern.

Zusammenarbeit mit dem Kunden steht über der Vertragsverhandlung heißt, dass die Zusammenarbeit mit dem Kunden und die Zufriedenheit des Kunden wichtiger ist, als die Vertragsverhandlungen. Im Endeffekt ist die Zufriedenheit des Kunden und die Erfüllung seiner Bedürfnisse ein Maßstab für den Projekterfolg. Es nutzt einem nichts, die Anforderungen des Kunden beim Vertragsabschluss so umzusetzen wie gewollt, ohne den Kunden im laufenden Prozess einzubinden. Denn im Endeffekt weiß der Kunde erst dann was er haben möchte und ob seine Anforderungen seine Probleme wirklich lösen,

wenn er das Produkt oder in dem Fall die Software ausprobiert. Deshalb sind ein ständiger Austausch und eine ständige Zusammenarbeit mit dem Kunden und den Anwenderinnen und Anwendern sehr erforderlich für den Projekterfolg. Es ist auch für niemanden vorteilhaft, weder für die Projektmanagerinnen und Projektmanager, Entwicklerinnen und Entwickler, noch für den Kunden und den Anwenderinnen und Anwender, wenn der Auftrag nach Vertrag zwar läuft aber schlussendlich eine unbrauchbare Software entsteht, die das ursprüngliche Problem oder Bedürfnis nicht erfüllt.

Durch ein großes Maß an Flexibilität kann man kurzfristig auf Kundenwünsche und Veränderungen reagieren. Vor allem zu der Zeit der Digitalisierung und des stetigen Wandels im Geschäftsalltag, erwarten sich Kunden eine kontinuierliche Anpassungsfähigkeit. Somit sind Anforderungen nicht in Stein gemeißelt und können jederzeit vom Kunden geändert oder ergänzt werden.

Reagieren auf Veränderung steht über dem Befolgen eines Plans besagt, dass die agile Software Entwicklung dazu da ist, um problemlos geänderte Anforderungen zu akzeptieren. Die Natur von Softwareprojekten bestätigt, dass im Laufe der Zeit sich ständig Veränderungen ergeben. Beispielweise werden zusätzliche Funktionalitäten benötigt, mithilfe deren, das Geschäftsziel besser erreicht werden kann. Im Laufe des Projekts lernen alle Beteiligten Individuen dazu und können somit auch bessere und effizientere Lösungsansätze vorschlagen, welche zum Projekterfolg beitragen. Aber genauso werden etablierte Prozesse sich verändern, da man im Laufe der Zeit erkennt, dass man mehr kommunizieren muss, oder mehr zusammenarbeiten muss, und genauso werden dadurch Prozesse angepasst und etabliert. Es geht aber nicht darum, Pläne, Prozesse und Deadlines komplett aufzugeben, sondern eher zu erkennen, wann ein Plan nicht mehr funktionieren kann und dementsprechend flexibel zu reagieren und Veränderungen zu schaffen. (vgl. Wolf-Gideon 2008, S. 13-15)

2.2.3 AGILE PRINZIPIEN

Das davor genannte agile Manifest basiert auf die folgenden zwölf Prinzipien:

- **Stelle den Kunden, durch frühzeitige und regelmäßige Lieferungen nützlicher Software, zufrieden:** Sehr viele Teams oder Firmen tendieren dazu, die Software einmal, und zwar am Ende des Projekts bzw. am Ende der Entwicklung zu liefern. Diese Vorgehensweise stellt allerdings den Kunden nicht zufrieden, da der Kunde regelmäßig die Features bzw. Funktionalitäten seiner Software verwenden möchte.
- **Stehe geänderten Anforderungen positiv gegenüber, selbst wenn sie bei der Entwicklung erst spät auftreten:** Oft führt komplexe Verwaltung von Anforderungen dazu, dass die Projektverantwortlichen der Meinung sind, dass alle Anforderungen seitens des Kunden im Vorhinein definiert sein müssen. Jedoch ist die Natur von Softwareprojekten eine solche, die mit Änderungen umgehen soll, da ständig Verbesserungsvorschläge und neue Technologien entwickelt werden.
- **Liefere lauffähige Software häufig aus:** Der Kunde ist nur an ein lauffähiges Produkt bzw. an eine lauffähige Software interessiert. Sehr viele Teams machen oft den Fehler, dass sie die Software nicht als Ganzheit in einem Gesamtsystem integrieren, sondern dass vereinzelte Funktionalitäten bzw. Features an kleinen Subsystemen laufen, die meist allein keinen Sinn ergeben. Dies passiert deshalb, weil die Integration als eine schwierige Aufgabe gesehen wird.
- **Fachexperten und Entwicklerinnen und Entwickler arbeiten zusammen:** Die Zusammenarbeit zwischen Entwicklerinnen und Entwickler und Fachexperten ist von großer Bedeutung. In sehr vielen Firmen sind die Abteilungen so voneinander getrennt, dass die Entwicklerinnen und Entwickler kaum mit den Fachexperten zusammenarbeiten. Dies führt zum Problem, dass die Informationen über die Anforderungen verloren gehen und dass kein direktes Feedback stattfindet.

- **Vertraue motivierten Individuen:** Gib den Individuen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die aufgeforderten Aufgaben fristgerecht und akkurat erledigen.
- **Kommunikation von Angesicht zu Angesicht:** Persönliche und direkte Kommunikation ist ein wesentlicher Bestandteil der agilen Prozesse. Vor allem könnte es in großen Teams passieren, dass man selten mit jedem Teammitglied zu einem persönlichen direkten Gespräch kommt. Dies führt dazu, dass das Vertrauen geschwächt wird.
- **Lauffähige Software ist das wichtigste Fortschrittsmaß:** Die lauffähige Software ist der wichtigste Maßstab, um den Projekterfolg zu messen und dadurch auch den Projektfortschritt zu bewerten. Lauffähige Software muss über die umfassende Dokumentation gestellt werden.
- **Fördere nachhaltige Entwicklung:** kontinuierlicher und gleichbleibender Rhythmus bei der Entwicklung der Software ist gesünder und produktiver und liefert bessere und nachhaltige Ergebnisse als die hektische Entwicklung. Agile Prozesse sind streng gegen die Überbelastung der Mitarbeiterinnen und Mitarbeiter und gegen unnötigen Arbeitsdruck.
- **Ständiges Augenmerk auf technische Güte und gutes Design:** Oft werden Software Projekte so umgesetzt, dass die Designer am Anfang des Projekts das Design festlegen, die Abteilung für die Qualität und das Controlling nur zu gewissen Zeitpunkten zum Einsatz kommen und dass die Software Tester gegen Ende vom Projekt die Software testen. Diese Vorgehensweise führt dazu, dass die zu liefernde Software keine Qualität mit sich bringt. Aus dem Grund, weil z.B. der Designer sein erstelltes Design testen muss, und genau so müssen auch alle Rollen, welche im Projekt arbeiten, in allen Projektphasen zusammenarbeiten. Nur so kann das Gesamtverständnis des Teams wachsen und die Software besser werden.

- **Einfachheit ist entscheidend:** Ein grundsätzliches Agiles Prinzip ist die Einfachheit. Es sollten unnötigen Prozessschritte und unnötige Arbeiten vermieden werden, die keinen Vorteil mit sich bringen. In sehr vielen Fällen wird die Tradition einer Firma gefolgt, ohne wirklich zu hinterfragen ob es in diesem einen Projekt so Sinn macht und ob man einige Zwischenschritte nicht vermeiden soll.
- **Selbstorganisierende Teams:** Agile Prozesse befolgen das Gedankengut des selbstorganisierten Teams. Jemanden die Kontrolle über das Team zu übergeben, erzeugt im Team ein Gefühl des Misstrauens und führt dazu, dass die Teammitglieder frustriert und deprimiert sind.
- **Teamreflektion und -anpassung:** Die Agilität ist kein Fan davon, Teams das Leben schwer zu machen, indem Prozesse vorgeschrieben werden, die nicht angepasst werden dürfen. Agile Prozesse geben den Menschen und den Teams eine Vorlage zum Verwenden. Teams sollen über ihre Prozesse reflektieren und falls notwendig diese Prozesse auch anpassen. Es existiert keine globale Prozessdefinition, die für jede Teamkonstellation und in jeder Situation mit jeder Projektgröße passt. (vgl. Michl, 2018)

2.3 AGILE ENTWICKLUNGSPROZESSE

2.3.1 SCRUM

Scrum ist ein englisches Wort und bedeutet auf Deutsch „Gedränge“. Der Begriff stammt ursprünglich aus der Rugby Sportart. Bei einem Rugby Spiel, sind jeweils acht Spieler der beiden Mannschaften auf dem Spielfeld. Die Spieler der beiden Mannschaften stehen am Spielfeld sehr eng nebeneinander und unterstützen sich gegenseitig sehr stark.



Figure 3: Scrum in der Rugby Sportart (Rugby, 2011)

Aus diesem Hintergrund der starken Teamarbeit und der Zusammenhalt im Team hat sich der Begriff „Scrum“ in der Softwareentwicklung etabliert. Sowohl beim Rugby als auch bei der Softwareentwicklung gibt es nur einige wenige Regeln die zu beachten sind, viel mehr steht bei beiden, sowohl beim Rugby also auch bei der Softwareentwicklung mit dem Framework „Scrum“ die kooperative, selbstverantwortliche und selbstorganisierte Zusammenarbeit des Teams im Mittelpunkt.

Somit beschreibt das Framework „Scrum“ ein sich entwickelnder Prozess, dass dazu beiträgt, dass die Teammitgliederinnen und Teammitglieder durch ihre Eigenverantwortlichkeit und ihre Selbstorganisation alle Aufgaben, die für die Produktentwicklung im Team notwendig sind, übernehmen. (vgl. Pichler 2008, S. 1-4)

Da das Framework „Scrum“ auf eine empirische Prozesssteuerung basiert, was bedeutet, dass das Framework auf die gesammelten Erfahrungen aufbaut, gibt es drei wesentliche Säulen: Transparenz, Überprüfung und Anpassung.

Transparenz: die wesentlichen Aspekte der Prozesse müssen für das gesamte Team klar sein. Genauso, dass diese Prozessaspekte gemeinsam im Team standardisiert werden, damit jede und jeder im Team ein gemeinsames Verständnis davon hat.

Überprüfung: ständige Überprüfung des Fortschritts des Sprintziels, um Abweichungen möglichst früh zu erkennen und dagegen wirken zu können. Dennoch soll die Überprüfung nicht den Teammitgliedern an das weiterarbeiten an das Sprintziels verhindern.

Anpassung: erkennt man bei der Überprüfung, dass es Mängel gibt, z.B. dass Prozessaspekte nicht funktionieren oder nicht so akzeptiert werden können, dann müssen schnelle Anpassungen erfolgen, um weitere Abweichungen zu vermeiden. (vgl. Schwaber 2016)

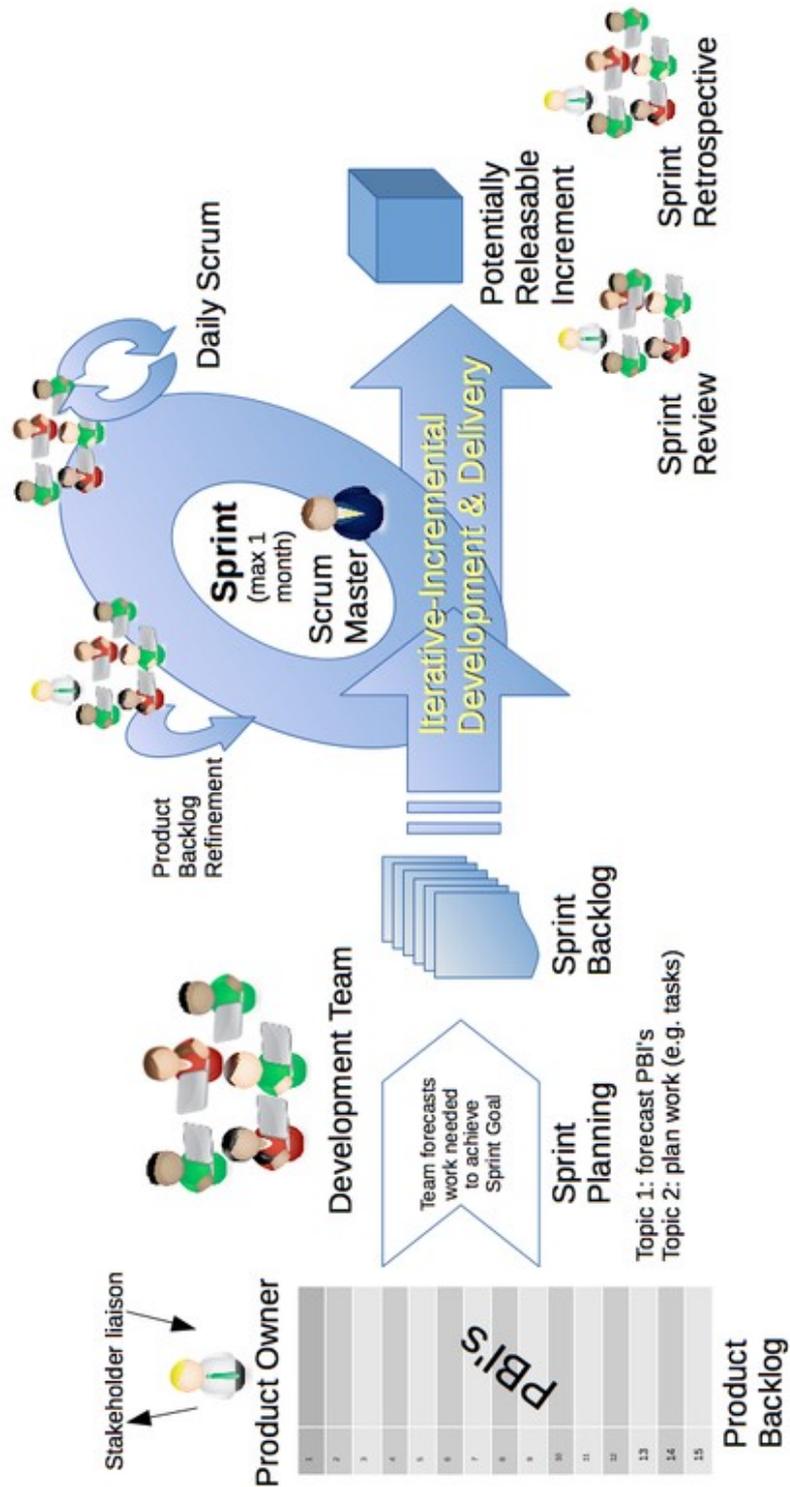


Figure 4: Übersicht Scrum Entwicklungsprozess (Mitchell, 2015)

2.3.2 EXTREME PROGRAMMING

Extreme Programming ist ein weiteres agiles Vorgehensmodell, welches auf das agile Manifest und die agilen Grundprinzipien aufbaut und das von einigen Unternehmen bei der Softwareentwicklung eingesetzt wird. Das agile Vorgehensmodell Extreme Programming, auch XP genannt, basiert auf die fünf Grundwerte der Agilität.

Das leichtgewichtige Vorgehensmodell Extreme Programming basiert auf die folgenden vier Phasen: Planung, Design, Implementierung und Testing. Hierbei werden diese Phasen in so genannte Iterationen durchgeführt. Dadurch entsteht eine iterative inkrementelle Software, die vom Kunden nach jeder Iteration auf die geforderten Funktionalitäten getestet wird. Jede Iteration dauert in der Regel zwei Wochen, aber das ist nicht im Stein gemeißelt, Teams können selbst entscheiden welche Dauer die Iteration haben soll und welche Dauer am meisten Sinn macht. In jeder Iteration sollen die Erfahrungswerte der vorigen Iteration miteinbezogen werden, um stets eine Verbesserung des Prozesses zu gewährleisten. (vgl. Pöpel 2014, S. 10)

2.3.3 KANBAN

Der agile Entwicklungsprozess „Kanban“ basiert auf die Philosophie von Kaizen. Kaizen bedeutet auf Japanisch: die Veränderung zum Besseren. Hierbei geht es also um die ständige Verbesserung, dies kann Faktoren wie die Qualität, die Produktivität und die Kundenzufriedenheit betreffen. Kanban dient vor allem sehr gut dafür, die Prozesse bzw. die Abläufe zu koordinieren und die Visualisierung der Prozesse zu verbessern. (vgl. Cron 2013, S. 11)

Kanban wird nach David J. Anderson als „komplexes, adaptives System“ erklärt, welches auf die folgenden fünf Prinzipien aufgebaut wird:

1. Visualisiere den Workflow

Das erste Kernprinzip von Kanban ist die Visualisierung der Prozesse. Gerade bei der Softwareentwicklung ist die Visualisierung der Prozesse enorm wichtig. Nicht wie bei einer Produktionsfabrik, wo die Arbeitsschritte greifbar und klar sind. So ist bei der Softwareentwicklung, wo die Arbeitsschritte nicht greifbar sind, wichtig diese Prozesse

zu visualisieren. Dadurch steigt auch die Transparenz zwischen den einzelnen Teammitgliederinnen und Teammitglieder und auch zwischen dem Entwicklungsteam und dem Kunden. So kann beispielsweise der Kunde den Fortschritt der Entwicklung betrachten und notfalls bei Engpässen eingreifen.

2. Begrenze den Work in Progress (Menge an begonnener Arbeit)

Kanban schreibt im zweiten Kernprinzip vor, dass nur eine gewisse Anzahl an Tasks, in der Spalte „Work in Progress“ bzw. WIP sein dürfen. Die Limitierung bzw. „WIP-Limits“ wird je nach Teammitgliederanzahl angepasst. Der Grund weshalb Kanban dies vorschreibt, ist folgender: Je mehr Tasks in der Spalte „Work in Progress“ ist, umso langsamer werden, die Tasks fertig, da eine Person nicht parallel an zwei Tasks arbeiten kann. Dadurch senkt der Durchfluss der Tasks, und außerdem ist das Arbeiten an verschiedenen Tasks gleichzeitig oder dass hin und her switchen zwischen den Tasks kontraproduktiv. Des Weiteren wirkt das Prinzip auch ziemlich gut auf die gegenseitigen Blockaden der Tasks, denn durch die Limitierung der WIP Spalte, werden weniger Tasks gleichzeitig entwickelt und mehr abgeschlossen, was zu Verringerung der Blockaden und Abhängigkeiten beiträgt

3. Führe Messungen durch und kontrolliere den Prozess

Jeder Prozess benötigt stets eine Verbesserung. Um Verbesserungspotenzial erkennen zu können, müssen Messungen durchgeführt werden. Genau dies besagt das dritte Prinzip. Dass anhand von Beobachtungen und Bewertungen gesagt werden kann, ob die Limitierung der WIP Spalte funktioniert, ob die Visualisierung der Arbeitsschritte eingehalten wird und falls nötig Maßnahmen getroffen werden, wird durch dieses Prinzip ermöglicht. Somit verbessert sich der ganze Arbeitsprozess und die Produktivität steigt.

4. Mache die Regeln für den Prozess explizit

Das Kanban System besteht auf gewisse Regeln bzw. Praktiken, für die sich das Team bekennt. Deshalb müssen die Regeln für das komplette Team transparent gemacht

werden, um sie auch einhalten zu können. Wenn die Regeln vom Team eingehalten werden, hat es zwei sehr gute Effekte auf das Team:

- Wenn sich die Teammitgliederinnen und Teammitglieder an die Regeln halten, erkennen sie dadurch welche Regeln angepasst oder verändert werden müssen.
- Durch die Einhaltung der Regeln findet bei den Diskussionen keine subjektive Schuldzuweisung statt, sondern eher eine objektive Betrachtung des Problems und dadurch eine Chance zur Verbesserung.

5. Implementiere Feedback-Mechanismen

Ein Grundprinzip der Agilität ist das Feedback geben und das Feedback nehmen. Bei Kanban ist es nicht anders, da Kanban auch „die Veränderung zum Besseren“ bedeutet. Deshalb ist die kontinuierliche Verbesserung durch die Rückkopplung entscheidend. Dafür setzen verschiedene Unternehmen auf verschiedene Methoden, um diese Feedback Schleifen zu gewährleisten. Eine Methode ist die Anwendung des Daily-Scrum Meetings, um einmal pro Tag zum Austausch zu kommen und Feedback über die eigene Arbeitssituation zugeben. Eine andere Methode ist das Retrospektive Meeting, wo sich die Teammitgliederinnen und Teammitglieder treffen, um Probleme anzusprechen und Verbesserungsvorschläge zu generieren. (vgl. Leopold 2013, S. 17-21)

2.4 TIMEBOXED VERSUS NON-TIMEBOXED

In den verschiedenen Projekten spielt der Zeitfaktor eine wesentliche Rolle. Sowohl für die Kunden ist es wichtig, dass sie möglichst schnell bzw. zeitnah eine lauffähige Software zum Laufen bekommen sowie für das Image des Unternehmens der die Software produziert. Somit lässt sich die agilen Softwareentwicklungs-Methoden in zwei Kategorien teilen: Timeboxed und Non-timeboxed Methoden.

Timeboxed ist eben, ein Zeitabschnitt, der nicht überschritten werden darf. Diese Technik gibt einem die Gelegenheit, die Zeit, die für eine bestimmte Aktivität genutzt werden soll, im Voraus zu begrenzen. So werden zum Beispiel Meetings, Entwicklungsabschnitte und

sämtliche Termine zeitlich eingegrenzt, damit sie nicht in die Länge gezogen werden. Timeboxed kann sich auch auf ein ganzes Projekt oder auch auf ein Subprojekt beziehen, dass z.B. bis Ende des Jahres fertig werden muss, da das Budget nur bis Ende des Jahres reicht. Genauso gibt es aber Projekte bzw. Subprojekte oder Pilotprojekte, die vom Kunden und vom Entwicklungsteam nicht in einem festgelegten Zeitfenster aufgrund verschiedener Faktoren das Prinzip vom Timeboxing nicht erfüllen können. Aufgrund dessen entstand in der agilen Softwareentwicklung die zwei Begrifflichkeiten: Timeboxed und Non-Timeboxed. (vgl. o. V. 2019)

Sieht man sich die Methoden Scrum und Extreme Programming an, so erkennt man, dass diese Ansätze von der Theorie her Timeboxed Ansätze sind. Scrum zum Beispiel basiert auf Iterationen in denen Tasks bzw. Aufgaben in einem festgelegten Zeitpunkt erledigt werden. Sieht man sich aber den Ansatz von Kanban an, erkennt man sofort, dass dieser Entwicklungsprozess einen Non-timeboxed Ansatz verfolgt. Deshalb findet man in Kanban keine zeitlichen Vorgaben, wann geplant, entwickelt oder geliefert werden muss. (vgl. Botta 2019)

2.5 ZUSAMMENFASSUNG

In diesem Kapitel wurde ein Überblick über die Grundlagen der Agilität in der Softwareentwicklung verschafft. Dabei wurden die Werte und die Prinzipien der agilen Softwareentwicklung und das agile Manifest wissenschaftlich beleuchtet und die Unterschiede zwischen dem klassischen Vorgehensmodell und dem agilen Vorgehensmodell näher dargestellt. Schlussendlich wurden die bekanntesten agilen Entwicklungsprozesse erläutert und die Begrifflichkeiten „Timeboxed und Non-timeboxed“ in Zusammenhang mit der Softwareentwicklung erklärt, die dann für den weiteren Verlauf der Masterarbeit ziemlich von Bedeutung sind.

3 TIMEBOXING

3.1 MEILENSTEIN UND TIMEBOX

In der Zeitmanagement redet man oft von den zwei Begrifflichkeiten Meilenstein und Timebox, die ähnlich klingen jedoch unterschiedlich sind.

Ein Meilenstein ist ein Konzept, wo es drum geht einen in der Zukunft zutreffendes Ereignis mit einem Termin zu verbinden. Dieses verbreitete Konzept dient, um beispielweise den Abschluss einer Phase in einem Projekt anzuzeigen. Meilensteine können aber auch in Submeilensteine unterteilt werden, die wiederum dieselbe Aufgabe haben, den Abschluss einer Phase und der Beginn einer anderen Phase zu demonstrieren. Kommt der Fall vor, dass die definieren Aufgaben in einem Meilenstein zu dem ausgemachten Termin nicht erledigt sind, so wird der Termin verschoben bis die Aufgaben für diesen Meilenstein erledigt sind.

Anders ist es beim Timebox. Timebox ist eine Technik, die ebenso dafür verwendet wird, um Aufgaben mit einem Termin zu versehen. Der große Unterschied zum Meilenstein ist, dass der Termin nicht nach hinten verschoben wird, falls der Umfang der vordefinierten Aufgaben nicht erledigt ist. Somit ist Timebox ein Zeitraum, indem die geplanten Aufgaben erledigt werden sollen und dieser nicht verändert wird, auch bei nicht vollständigem Erledigen der Aufgaben. Somit hat die Einhaltung des Termins eine viel höhere Priorität als die fertigen Aufgaben. (vgl. Vogenschow 2015, S. 4)

3.2 TIMEBOXING UND DAS PARETO-PRINZIP

Das Pareto Prinzip stammt vom italienischen Ökonom Vilfredo Pareto. Dieses Prinzip besagt, dass 20 Prozent der Gesamtbevölkerung 80 Prozent des Volksvermögens besetzen. Die gleiche Gleichung lässt sich auf die Zeit anwenden, so wird in 20 Prozent der Zeit 80 Prozent des Erfolgs erzielt. Timeboxing hat vor allem die Vision, dieses Prinzip anzuwenden und das Verhältnis zwischen Leistung und Zeit bestmöglich zu steigern. (vgl. Strasser 2012, S. 51-52)

3.3 VORTEILE TIMEBOXING

Verbesserter Fokus: Dank Timeboxing wird das fokussierte Erledigen der Aufgaben einfacher gemacht. Timeboxing gibt der Projektmanagerin bzw. dem Projektmanager den Eindruck das das Entwicklungsteam mental drauf eingestellt ist, die Aufgaben fokussiert abzuarbeiten, ohne sich von anderen Nebenaufgaben ablenken zu lassen. So werden die Tasks mit voller Aufmerksamkeit angegangen, um sie zeitgerecht zu erledigen.

Scope: Mithilfe eines Projektzieles kann Timeboxing dazu beitragen, den Projektzeitplan eines Projekts festzulegen. Somit kann das Projekt mit einem gewissen Puffer nach Plan verlaufen und vorausschauend das Enddatum der Entwicklungszeit des Projekts bestimmen.

Klarheit: Timeboxing unterstützt einen Projektmanagern dabei Klarheit über den Projektablauf zu schaffen. (vgl. Lucidchart 2019)

Neubewertung: Timeboxing bietet eine Möglichkeit zur Neubewertung es Projekts. Im alltäglichen Privatleben als auch im Berufsleben kommt man selten dazu die eigene Leistung, sowohl die Projektleistung zu bewerten. Mithilfe von Timeboxing, weißt man am Ende der festgelegten Zeit ob man, die Aufgaben fristgerecht erledigt hat oder nicht. Anhand dessen kann man sich Verbesserungsvorschläge überlegen oder den Grund für den Fortschritt analysieren, um weiterhin erfolgreich zu sein.

Erfolgserebnis: Bei vielen großen Projekten, die man vorwärtsbewegen muss, benötigt man eine sehr lange Periode, um einen Projekterfolg zu erleben. Diese einschüchternden, demotivierenden Gefühle gibt es bei Timeboxing nicht, da man nach jeder Timeboxing Periode diesen Erfolgseffekt erlebt. (vgl. Boograard 2019)

Tasks priorisieren: In jedem Projekt haben die verschiedensten Tasks bzw. Aufgaben unterschiedliche Prioritäten für den Kunden. Timeboxing zwingt das Entwicklungsteam die Aufgaben mit der höheren Priorität zuerst zu erledigen. Wenn die geschäftskritischen Aufgaben fristgerecht geliefert werden können, ist der Kunde automatisch zufriedener, was auf das Team wiederum positiv wirkt, da dadurch das Eigenengagement und die Eigenverantwortlichkeit steigt.

Fortschritt verfolgen: Das Timeboxing spiegelt den Fortschritt der einzelnen Schritte des Projekts wider. Dadurch hält man alle Projektbeteiligten wie Stakeholder, Sponsoren usw. am Laufenden und benötigt keine Projektstatusmeetings. Somit ist es ein zeitlicher Prüfraster für den Projektfortschritt und unterstützt uns bei der Fortschrittskontrolle.

Perfektionismus vermeiden: Timeboxing zwingt die Teammitgliederinnen und Teammitglieder die geplanten Tasks in der vordefinierten Zeitspanne fertig zu stellen. Wodurch auch eine gute Lösung entsteht, die wahrscheinlich nicht perfekt ist. Man hat durch Timeboxing automatisch den Vorteil, dass unnötiger Perfektionismus vermieden wird, denn dies bringt oft mehr Schaden als Nutzen mit sich.

Vorhersehbarkeit fördern: Ein weiterer Vorteil von Timeboxed Ansätze ist, dass es die Vorhersehbarkeit des Teams fördert. So müssen Teams vor dem Beginn der Timeboxed Zeitperiode vorhersagen wie viele Tasks sie in diesem Zeitabschnitt schaffen können. (vgl. Inamdar 2018)

3.4 VORTEILE NON-TIMEBOXING

Kreativitätssteigerung: Mithilfe von Non-timeboxed Ansätze hat das Team mehr Zeitraum. So kann Non-timeboxing dann eingesetzt werden, wenn die Kreativität der Mitarbeiterinnen und Mitarbeiter gefragt ist.

Fertige Produkte: Der Nachteil von Timeboxing ist, dass man nicht gewährleisten kann, dass alle Tasks am Ende des Zeitabschnitts fertiggestellt werden. Somit ist es ein Vorteil im Non-timeboxed Ansatz, dass die Tasks nicht eine ganze Periode warten müssen, bis sie geliefert werden.

Mehr Diskussionsrahmen: Wichtige Meetings und sämtliche wichtige Diskussionen, die für das Team vom Vorteil sind, werden nicht automatisch unterbrochen sobald die vorgesehene Zeit eintrifft. Es kann bei Timeboxing vorkommen, dass die Angelegenheit ungenau diskutiert werden, weil die Zeiteinschätzung möglicherweise nicht korrekt war. Dies kommt bei Non-timeboxing nicht vor. (vgl. o. V. o. J.)

3.5 ZUSAMMENFASSUNG

In diesem Kapitel wurde der Begriff Timeboxing in Kombination mit der Softwareentwicklung näher erläutert, dabei wurden einige Vorteile von Timeboxed Ansätze und einige Vorteile von Non-timeboxed Ansätze vorgestellt, welche man in der Literatur wiederfindet. Da es in der Literatur keine Quellen für das Thema „Auswahlkriterien für Timeboxed und Non-timeboxed Ansätze“ gibt, wird diese Frage nur im Forschungsbereich behandelt.

4 FORSCHUNGSTEIL: AUSWAHLKRITERIEN FÜR TIMEBOXED UND NON-TIMEBOXED ANSÄTZE

In diesem Kapitel wird der Forschungsteil dieser Masterarbeit beschrieben. Als allererstes wird die Herangehensweise bei der Durchführung der Forschung näher dargelegt, danach werden die Ergebnisse dieser Forschung vorgestellt und schlussendlich folgt die Auswertung und die Interpretation.

Dieser Masterarbeit beschäftigt sich mit der Frage: „Welche Auswahlkriterien sind entscheidend für Timeboxed und Non-timeboxed Ansätze?“

Hiermit wird in diesem Kapitel überprüft, unter welchen Umständen welcher Ansatz (Timeboxed oder Non-timeboxed) sich am besten eignet und dementsprechend Entscheidungskriterien festzulegen anhand dessen man sich für agile Timeboxed oder Non-timeboxed Ansätze entscheiden kann.

4.1 VORGEHENSWEISE

4.1.1 BEGRÜNDUNG DER FORSCHUNGSMETHODEN

Da über dieses Thema, Entscheidungskriterien für Timeboxed und Non-Timeboxed Ansätze, kaum erforscht wurde, wird hier bei dieser Masterarbeit eine qualitative Forschungsmethode angewandt, da Laut Flick (vgl. Flick 2000, S. 14-17), die qualitative Forschung sich gut für Anwendungsfelder mit wenig untersuchtem Forschungsgebieten eignet. Zusätzlich lässt sich mithilfe von qualitativen Forschungsmethoden wie die Durchführung von Interviews mit offenen Fragen und einen Diskussionsraum, erste neue Kenntnisse gewonnen werden. Außerdem bekommt man vollständige Informationen von den Interviewpartnern durch die gestellten offenen Fragen und die Diskussion.

Aufgrund dessen wurde hier bei der Forschungsfrage dieser Masterarbeit, die qualitative Forschungsmethode Interview herangezogen.

4.1.2 DATENERHEBUNG MITTELS INTERVIEWS

Im Rahmen der Durchführung der Forschungsmethode wurden sechs Interviews mit Expertinnen und Experten durchgeführt. Es wurden diese sechs Personen ausgesucht, weil die meisten von ihnen Erfahrungen in der Software-Entwicklung in verschiedenen Rollen (in der Entwicklung als auch im Management Bereich) mitbringen. Somit werden ihre Aussagen nicht rein von der Entwicklungsseite, noch rein von der Managementseite sein, sondern werden beide Bereiche abdecken. Die folgende Tabelle soll einen Überblick über die durchgeführten Interviews schaffen.

ID	Alter	Position	Geschlecht	Abschluss	Dauer
B1	24 Jahre	Service Delivery Manager	Männlich	Bachelor of Science	29 Minuten
B2	34 Jahre	SDQA- Manager	Männlich	Diplomingenieur	26 Minuten
B3	40 Jahre	Scrum Master	Weiblich	MBA in IT-Management	29 Minuten
B4	26 Jahre	Gründer und Geschäftsführer	Männlich	Bachelor of Science	24 Minuten
B5	35 Jahre	Software-Entwickler	Männlich	Master of Science	26 Minuten
B6	29 Jahre	Scrum Master	Weiblich	Master of Science	23 Minuten

Tabelle 1: Übersicht über die interviewten Personen

Interviewablauf

- Kurze Vorstellung der Person: Alter, Titel, Geschlecht
- Bedanken für die Teilnahme am Interview

- Forschungsfrage und -ziel vorstellen
- Information über die anonyme Behandlung der Daten
- Einverständnis holen & fragen ob es offene Fragen gibt
- Interview startet
- Interview endet, bedanken für die Zeit und alles Gute wünschen

Interviewfragen

1. Was ist deine höchst abgeschlossene Ausbildung und welche Berufserfahrung bringst du im Bereich Software-Entwicklung mit?
2. Benutzt du in deinen Softwareprojekten agilen Entwicklungsmethoden? Wenn ja, welche benutzt du?
 - Hast du Timeboxed Ansätze (z.B. Scrum) und Non-Timeboxed Ansätze (z.B. Kanban) angewandt?
 - Falls Scrum angewandt wurde, wurde Scrum Timeboxed oder eher nicht? Und falls nicht warum nicht?
3. Bezogen auf die Zeit, also Timeboxing, was ist deine Meinung nach die Auswahlkriterien, warum ein Projektmanager Timeboxed bzw. Non-timeboxed Ansätze anwenden soll?
4. Inwiefern kann die Rolle der Timeboxing bzw. Non-timeboxing einen Einfluss auf die die Psyche der Mitarbeiterinnen und Mitarbeiter haben? Psychologische Effekte?
 - Motivation
 - Druck bzw. Stress

5. Inwiefern ist die Struktur des Teams ein Einflussfaktor um sich für Timeboxed bzw. Non-timeboxed Ansätze zu entscheiden?
 - Fixe Mitarbeiter oder Aushilfsmitarbeiter?
 - Nebenjob oder Vollzeitjob?
 - Welches Setting im Team?
 - Welche Berufserfahrungen im Team?
 - Zusätzliche Firmenaufgaben? (z.B. Lead Aufgaben)
6. Inwiefern kann die Team-Velocity und die Sprintdauer (Sprintumfang) ein Einflussfaktor sein um sich für Timeboxing oder Non-timeboxing zu entscheiden.
 - Wenn Team-Velocity bekannt, dann Timeboxed oder Non-Timeboxed? und warum?
 - Ist es überhaupt möglich die Team-Velocity vorauszusagen?
 - Macht es einen Unterschied ob der Sprint länger als zwei Wochen oder weniger als zwei Wochen dauert für diese Entscheidung?
7. Inwiefern kann die Entscheidung davon abhängen, ob man in der Go Live Phase ist und man die Themen fertigstellen muss.
8. Welches Projektsetting ist entscheidend für Timeboxing bzw. Non-timeboxing?
 - Pilotprojekte
 - Software oder Support- & Wartungsprojekte
 - Projekt
 - Produkt

9. Haben die zwei Ansätze „Timeboxing“ und „Non-timeboxing“ einen Einfluss auf die Qualität der Software?

Die ersten 2 Fragen des Interviews sind Eingangsfragen. Sie dienen, um den Einstieg in das Interview zu erleichtern. Somit beschäftigen sich die Fragen 3 bis 9 mit der eigentlichen Forschungsfrage. Bei einigen Fragen sind einige Unterpunkte bzw. Unterfragen zu finden, welche eine Hilfestellung dem Befragten dienen soll, aber seine Antwort trotzdem nicht einschränken soll.

4.1.3 DATENAUSWERTUNG MITTELS QUALITATIVER INHALTSANALYSE

Um die Daten der durchgeführten Interviews am besten auszuwerten, eignet sich die Inhaltsanalyse von Phillip Mayring. Die Inhaltsanalyse von Mayring gibt eine Systematik und einige Analyseschritte vor, um die Inhaltsanalyse qualitativ durchführen zu können. (vgl. Mayring 2008, S. 469)

In dieser Masterarbeit wird der Ansatz von Mayring angewandt, um die Inhaltsanalyse durchzuführen. Das Material, welches mithilfe der Inhaltsanalyse von Mayring bearbeitet wird, sind die sechs transkribierten Interviews. Beim Transkribieren wurden folgende Regeln gehalten:

- Eindeutige Kürzel für die beteiligten Personen wie I = Interviewer und B1 = erster Befragter usw.
- Sprechwechsel durch Leerzeile zwischen den Sprechbeiträgen
- Wortwörtliches Transkribieren, d.h. keine zusammenfassende Wiedergabe
- Lautäußerungen der Befragten, wie lachen, seufzen oder „Ähham“ wurden ausgelassen
- Längere Pausen wurden ausgelassen
- Leichte Glättung der Sprache an Schriftsprache

- die Befragten wurden anonym gehalten

Die sechs transkribierten Interviews wurden im nächsten Schritt dann herangezogen, um die Kategorienbildung durchzuführen. Im Anhang sind die sechs transkribierten Interviews zu finden.

4.1.4 VORGEHENSWEISE BEI DER KATEGORIENBILDUNG

Um die Kategorienbildung durchzuführen wurde die induktive Kategorie-Entwicklung angewandt. Aus diesem Grund wurde die induktive und nicht die deduktive Kategorienbildung herangezogen, weil es kaum Literatur über die Auswahlkriterien von Timeboxed und Non-Timeboxed Ansätze gibt. Die induktive Kategorienbildung nach Mayring besagt nämlich, dass man die Kategorien direkt aus dem vorliegenden Material extrahiert, ohne sich auf Theoriekonzepte zu beziehen. (vgl. Mayring 2008, S. 75).

Im Anhang sind die Kategorienbildungen der einzelnen Interviews zu finden. In der nachfolgenden Tabelle ist ein Gesamtüberblick über das endgültige Kategoriensystem zu finden.

Nr.	Kategorie	Unterkategorie
1	Überblick über das Projekt	
2	Psychische Effekte	2.1 zeitlicher Druck 2.2 Motivation
3	Innovation	3.1 Kreativität 3.2 Weiterentwicklung
4	Teamsetting	4.1 Erfahrung

		<p>4.2 Anstellungsart</p> <p>4.3 100% in einem Projekt</p> <p>4.4 Teamgröße</p> <p>4.5 Sub-Teams</p>
5	Velocity und Iterationsdauer	<p>5.1 Velocity</p> <p>5.2 Iterationsdauer</p>
6	Go Live	
7	Projektart	<p>7.1 bekannte Projekte</p> <p>7.2 Pilot-, Forschungsprojekte und Prototypen Entwicklung</p> <p>7.3 Support-, Wartungs- und Betriebsprojekte</p> <p>7.4 Produkte und Projekte</p>
8	Qualität	

Tabelle 2: endgültiges Kategoriensystem

4.2 AUSWERTUNG UND INTERPRETATION

Aus allen Interviews geht hervor, dass die Befragten in ihren Software-Projekten agile Softwareentwicklungsmethoden verwenden. Nur eine, der befragten Personen, benutzt Kanban, einen Non-Timeboxed Ansatz, ansonsten benutzen die restlichen fünf befragten Personen Scrum als Timeboxed Ansatz. Alle sechs Interview-Teilnehmerinnen haben die Frage nach den Auswahlkriterien für Timeboxed und Non-Timeboxed Ansätze für sinnvoll empfunden und die demnächst folgenden Kategorien als Gründe dafür gegeben. Die Beschreibung der Kategorien resultiert einerseits aus Aussagen der befragten Personen, und andererseits aus zusätzlichen Erklärungen, um die Aussagen der Befragten zu erläutern.

4.2.1 KATEGORIE: ÜBERBLICK ÜBER DAS PROJEKT

BESCHREIBUNG

Die erste Kategorie „Überblick über das Projekt“ beschäftigt sich damit, dass der Kunde, der das Projekt in Auftrag gibt, einen Überblick über das Projekt hat. Diesen Überblick soll dem Kunden nicht nur dabei helfen einen Zeitplan für die Projektzukunft erstellen zu können, sondern auch um für beide Parteien, sowohl Auftraggeber als auch Auftragnehmer, eine Klarheit über die Kosten-technischen Fragen zu schaffen.

INTERPRETATION

Befragter 1 (B1) und B6 sind der Meinung, dass Timeboxing diesen Vorteil mit sich bringt, dass man einen Überblick über das Projekt hat. Somit kann der Kunde für sich einen zeitlichen Überblick schaffen, indem der Kunde weiß, welche Tasks in welcher Zeit ca. fertig werden. Somit kann der Kunde auch seine Geldressourcen planen und weiß welche Kosten, bis wann anfallen würden. Andererseits ist Timeboxing sofern vorteilhaft für den Auftragnehmer des Projekts, indem der Auftragnehmer weiß, welche Ressourcen in welcher Zeit mit welchen Anforderungen an Tasks er hat und somit der Auftragnehmer genauso einen Überblick über die Erwartungen hat, um mit dem Team andere Projekte zu planen. Im Gegensatz dazu sieht B1, dass Non-Timeboxing keinen Überblick für beide Parteien schafft. B6 meint, wäre der Aspekt der Übersicht dem Kunden egal, das heißt,

wäre es dem Kunden egal, welche Tasks gerade in Arbeit sind, welche Features bald fertig sind, welche Iterationen abgeschlossen werden und der Kunden interessiert sich nur für das Endprodukt zu einer bestimmten Deadline, dann würden auch Non-Timeboxing Ansätze in Frage kommen. Wobei B1 der Meinung ist, dass man mit Timeboxing, dem Kunden eher damit garantiert, dass die Arbeit erledigt wird und dass das Projekt zu der bestimmen Deadline fertiggestellt wird, was natürlich auch das Vertrauen erhöht.

4.2.2 KATEGORIE: PSYCHISCHE EFFEKTE

BESCHREIBUNG

Unter der Kategorie „Psychische Effekte“ sind jene psychischen Effekte gemeint, die den Mitarbeiterinnen und Mitarbeitern bei der Durchführung von Projekten im Wege stehen bzw. aufmuntern.

Diese Kategorie lässt sich in zwei Unterkategorien unterteilen. Die erste Unterkategorie wäre „zeitlicher Druck“. Mit zeitlichem Druck ist der zeitliche Druck für das Entwicklungsteam bei der Ausführung von Software Projekten gemeint.

Die zweite Unterkategorie nennt sich „Motivation“ und beschreibt im Wesentlichen, den Aspekt der Motivation für das Entwicklungsteam bei der Durchführung von Software Projekten.

INTERPRETATION

Bei der Unterkategorie „zeitlicher Druck“, sind sich alle befragten Personen einig, dass ein zeitlicher Druck niemals bei Non-Timeboxing Ansätze entsteht, sondern eher wahrscheinlicher bei Timeboxing Ansätze. Somit wäre die Rede bei dieser Unterkategorie über Timeboxing Ansätze. Wenn man die Iterationen bei Timeboxing Ansätze überplant und viel mehr Tasks bzw. Story Points aufnimmt, die das Team nicht schaffen könnte, dann wird dieser zeitliche Druck sich mit Sicherheit ergeben. Es kann auch in eine Richtung gehen, dass die Mitarbeiterinnen und Mitarbeiter des Teams nicht mehr in die Arbeit gehen möchten und teilweise kommt es auch manchmal aufgrund von zeitlichem Druck und Stress zu Kündigungen. Andererseits ist das Team gefragt, sich aktiv

in die Iterationsplanung einzubringen und solche Situationen zu vermeiden, dass Iterationen überplant werden. So kann das Team eben bei der Iterationsplanung einen gewissen Puffer miteinberechnen, damit die Tasks nicht auf den letzten Drücker erledigt werden und sich dieser zeitliche Druck bzw. diese psychische Belastung behebt.

Bei der Unterkategorie „Motivation“ haben alle befragten Personen kein einziges Mal drauf hingewiesen, dass Non-Timeboxing motivierend sein könnte. Vielmehr geht es bei dieser Unterkategorie wieder um Timeboxing Ansätze. Falls man bei Timeboxing einen zeitlichen Druck hat, ergibt sich automatisch, dass das Team eher belastet und nicht motiviert ist. Somit sind beide Unterkategorien „zeitlicher Druck“ und „Motivation“ voneinander abhängig. So ergibt sich wie oben bei der Unterkategorie „zeitlicher Druck“ erwähnt, dass bei einem zeitlichen Druck unter anderem die Motivation zu arbeiten oder Tasks abzuschließen, nicht gegeben ist. Andererseits, wenn der Zeitdruck nicht vorhanden ist und das Team einen gewissen Puffer in die Iteration miteinbezieht, ist es sehr motivierend für das Team, die vordefinierten Story Points bzw. Tasks in den vereinbarten Zeitraum abzuschließen. Denn es ist sehr motivierend zu sehen, dass man als Team einen Erfolg erzielt hat, und dadurch wird das Team mit voller Motivation die nächste Iteration starten. Zusätzlich sieht B3, dass Timeboxing vom Haus aus motivierend ist, weil es vorgibt, dass die Planung der nächsten Iteration im Vorfeld fix steht, und sich das Team drauf einstellen kann, welche Aufgaben zu erledigen sind und dass das auf das Stresslevel des Teams sich positiv auswirkt.

4.2.3 KATEGORIE: INNOVATION

BESCHREIBUNG

Die Kategorie „Innovation“ lässt sich in zwei Unterkategorien „Kreativität“ und „Weiterentwicklung“ unterteilen. „Kreativität“ bedeutet in diesem Zusammenhang, dass die Kreativität der Mitarbeiterinnen und Mitarbeitern bei Software-Entwicklungsprojekten mit einem zeitlichen Druck sehr blockiert wird. Aufgrund des zeitlichen Drucks finden die Mitarbeiter die nötige Zeit nicht um neue kreative Lösungen vorzuschlagen und auszuprobieren.

Die zweite Unterkategorie „Weiterentwicklung“ bezieht sich ebenso auf die Mitarbeiterinnen und Mitarbeitern des Entwicklungsteams, und zwar, dass die nötige Zeit, um sich in den Bereich zu vertiefen und weiter zu bilden, aufgrund des zeitlichen Drucks nicht gegeben ist.

INTERPRETATION

Diese Kategorie „Innovation“ ist sehr stark von der vorigen Kategorie „psychische Effekte“ abhängig. Sobald der zeitliche Druck, welcher sich bei Timeboxing Ansätze ergibt, vorhanden ist, beschränkt dies auf der einen Seite, die Kreativität der Mitarbeiterinnen und Mitarbeitern des Teams bei der Entwicklung von Lösungsansätzen eines Features. Dementsprechend hat man aufgrund des Stress-Faktors nicht die Freiheit sich mit neuen Frameworks oder mit neuen Technologien zu beschäftigen und neue kreative Ideen zu kreieren. Auf der anderen Seite beschränkt es ebenso die Weiterbildung, weil die Zeit in Timeboxed Iterationen mit einem Stressfaktor nicht gegeben ist.

Da bei Non-Timeboxing, der zeitliche Druck nicht vorhanden ist, hat man natürlich hier mehr die Freiheit, seine Kreativität einen freien Lauf zu lassen neue Ideen bei der Implementierung eines Features mit neuen Technologien auszuprobieren. Genauso bietet es sich im Non-Timeboxing sehr gut an, dass die Mitarbeiterinnen und Mitarbeiter sich in ihren Rollen und Schwerpunkten weiter- und ausbilden.

4.2.4 KATEGORIE: TEAMSETTING

BESCHREIBUNG

Die Kategorie „Teamsetting“ beschäftigt sich damit, wie das Entwicklungsteam eines Softwareprojektes aufgebaut ist. Diese Kategorie lässt sich in fünf Unterkategorien unterteilen: Erfahrung, Anstellungsart, 100% in einem Projekt, Teamgröße, Sub-Teams.

Unter der Unterkategorie „Erfahrung“ versteht man nicht nur die Berufserfahrung der Mitarbeiterinnen und Mitarbeiter des Entwicklungsteams, sondern auch die Teamerfahrung bzw. die Reife des Teams und wie gut sie untereinander arbeiten können.

Die Unterkategorie „Anstellungsart“ beschäftigt sich mit der Anstellungsart der Mitarbeiterinnen und Mitarbeiter des Entwicklungsteams. Das heißt ob man ein fixer Mitarbeiter im Team ist oder ob man nur eine Aushilfskraft für eine kurze Zeit ist. Aber auch ob man Vollzeit angestellt ist oder eben Teilzeit angestellt ist.

Die nächste Unterkategorie „100% in einem Projekt“ beschäftigt sich damit, dass die Mitarbeiterinnen und Mitarbeiter des Entwicklungsteams nicht parallel zu den Aufgaben im Projekt, noch weitere zusätzliche Aufgaben, wie Teamleader, Chapterleader usw. bekommen, ohne dass man die Zeit für die zusätzlichen Aufgaben wegrechnet.

Unter der nächsten Unterkategorie „Teamgröße“, ist die Größe des Entwicklungsteams gemeint, ob das Team nur aus ein paar Mitarbeiterinnen und Mitarbeiter besteht oder aus vielen.

Die letzte Unterkategorie „Sub-Teams“ beschäftigt sich mit dem Fall, dass ein Softwareprojekt von mehreren Sub-Teams oder Abteilungen gleichzeitig betreut bzw. entwickelt wird.

INTERPRETATION

Die Teamkonstellation bzw. das Teamsetting hat sich als ein entscheidendes Kriterium für alle Interviewer herauskristallisiert. Bei einer optimalen Teamstruktur, wo im Team fast bis gar keine Aushilfskräfte angestellt sind, wo alle Mitarbeiterinnen und Mitarbeiter des Teams sich zu 100% auf ihre Aufgaben und ihr Projekt konzentrieren können, würde es Sinn ergeben Timeboxing Ansätze zu verwenden, da man mit solchen Bedingungen optimal planen kann. Andersrum, wenn man ein sehr instabiles Team hat, mit dem man nicht wirklich planen oder rechnen kann, macht es durchaus Sinn einen Non-Timeboxed Ansatz zu befolgen.

Wenn wir über die erste Unterkategorie „Erfahrung“ sprechen, so haben alle Befragten eines ausgeschlossen, und zwar, dass die Berufserfahrung nicht damit zu tun hat, ob man Timeboxing oder Non-Timeboxing Ansätze anwendet. Viel mehr sprachen die befragten Interviewer über die Erfahrung im Team und über den Reifegrad des Teams. Dementsprechend waren B3 und B6 (beide Scrum-Master) der Meinung, dass im Falle

ein hoher Reifegrad des Teams auch Non-Timeboxing Ansätze in Frage kommen würden, da bei einem hohen Reifegrad des Teams der Informationsfluss zwischen den Mitarbeiterinnen und Mitarbeitern des Teams automatisch stattfindet.

Bei der nächsten Unterkategorie „Anstellungsart“ waren sich alle Befragten einig, dass man mit Aushilfskräften, vor allem wenn es mehrere solche Anstellungen im Team gibt, sprich Mitarbeiterinnen und Mitarbeiter, die nicht fix im Team sind, sondern nur vorübergehend beschäftigt sind und je nach Bedarf Aufgaben übernehmen, es nicht klug wäre Timeboxing anzuwenden, sondern eher Non-Timeboxing, weil man sehr schwer mit solchen Anstellungen rechnen und planen kann. Der Optimalfall wäre, wenn man im Team rein fixe Vollzeit Beschäftigte hat mit zusätzlich paar Teilzeit Angestellte, dann würde Timeboxing gut funktionieren, weil man mit solchen Mitarbeiterinnen und Mitarbeitern besser planen und rechnen kann.

Bei der dritten Unterkategorie „100% in einem Projekt“ haben viele der Befragten sich dazu geäußert, dass sie viele Erfahrungen damit gemacht haben, dass Personen für verschiedenen Aufgaben im Team, aber auch Firmenweit zuständig sind, diese aber kaum bzw. sehr wenig bei der Planung im Projekt berücksichtigt werden. So werden Personen die zusätzliche Aufgaben, wie Chapterleader oder Teamleader haben, trotzdem im Projekt zu fast 100% miteinkalkuliert. Diese Belastung führt natürlich in eine Richtung, wo die Arbeitskraft ihre Aufgaben nicht vollständig in der einberechneten Zeit ausführen kann, und ebenso belastet wird. Mit solchen Mitarbeiterinnen und Mitarbeitern ist es natürlich schwer zu planen, daher ist es besser in solchen Fällen Non-Timeboxing Ansätze anzuwenden, da solche Personen nicht wie bei Timeboxed Ansätze konstant Leistung erbringen können.

Zur Unterkategorie „Teamgröße“ gaben Interviewer B4 und B6 an, dass ab einer bestimmten Größe, es Sinn macht Timeboxing anzuwenden, weil unter anderem die Kommunikation und der Informationsfluss strukturiert gehandhabt wird. Hat man ein Team, das aus wenigen (< 5) Mitarbeiterinnen und Mitarbeiter besteht, so kann man eben auch Non-Timeboxing Ansätze anwenden.

Bei der letzten Unterkategorie „Sub-Teams“ waren B4 und B6 der Meinung, dass Teams, die aus Sub-Teams oder Abteilungen bestehen, die an einem Gesamtprodukt zusammenarbeiten, auf jeden Fall Timeboxing Ansätze verwenden müssen. Es macht nur Sinn Timeboxing zu verwenden, da man Tasks hat, die in verschiedenen Sub-Teams parallel implementiert werden, die vielleicht voneinander abhängig sind. Aufgrund dessen muss man in solchen Fällen Time-boxen, ansonsten muss ein Sub-Team auf die Fertigstellung einer Aufgabe warten, es aber nicht bekannt ist wann die Aufgabe ca. fertig wird, um mit der anderen Aufgabe beginnen zu können. Ein Beispiel, das hier erwähnt wurde, war die Abhängigkeit von Frontend- und Backend-Tasks in Frontend- und Backend-Sub-Teams.

4.2.5 KATEGORIE: VELOCITY UND ITERATIONSDAUER

BESCHREIBUNG

Die Kategorie „Velocity und Iterationsdauer“ beschäftigt sich einerseits mit der Velocity des Teams und inwiefern die Velocity einen Einfluss auf die Entscheidung Timeboxing oder Non-Timeboxing haben kann. Die Velocity ist die Entwicklungsgeschwindigkeit des Teams und beschreibt die Anzahl an Story Points, die ein Team während einer Iteration umsetzt. Andererseits behandelt diese Kategorie den Aspekt der „Iterationsdauer“, sprich welche Iterationsdauer vernünftig für Timeboxing und Non-Timeboxing wäre.

INTERPRETATION

Im Grunde sind alle Interviewer der Meinung, dass beide Unterkategorien „Velocity“ und „Iterationsdauer“, primäre Entscheidungskriterien sind. Desweiteren sind sich alle Befragten ebenso einig, dass wenn man die Velocity ermitteln kann und eine genau definierte Iterationsdauer von bis zu zwei Wochen hat, die Entscheidung primär für Timeboxing Ansätze gefallen würde.

Die Entscheidung über die Velocity hängt sehr stark mit dem Aspekt des Teamsettings zusammen. Sieht man aber die Erfahrungen der Interviewer an, so erkennt man, dass die Velocity in sehr vielen Fällen nicht einfach zu tracken ist. Vor allem dann nicht, wenn man ein instabiles Team hat mit sehr vielen Aushilfskräften und mit Rahmenbedingungen

zu tun hat, die dazu beitragen, dass die Velocity schwer vorausszusagen ist. Einige Beispiele für diese Rahmenbedingungen sind: unerwartete Kündigungen, öfter die Teamgröße in kurzen Abständen zu reduzieren oder zu erhöhen, oder auch fehlende Kommunikation zwischen den Teammitgliedern. Bei solchen Rahmenbedingungen tendieren die Befragten eher zur Non-Timeboxing Ansätze.

Bei der zweiten Unterkategorie „Iterationsdauer“, gibt es keine Meinungsverschiedenheit zwischen den interviewten Personen, dass je kürzer die Iteration desto planbarer und übersichtlicher kann man arbeiten. Sprich eine Iteration von bis zu zwei Wochen, spricht für einen Timeboxing Ansatz. Je länger die Iterationen werden, desto unüberschaubar und schwerer kann man sie planen, weil man einen längeren Zeitraum schwerer genau vorausplanen kann als ein kurzer Zeitraum. Somit würden die Interviewer bei Iterationen die länger als zwei Wochen dauern, dazu tendieren Non-Timeboxing Ansätze zu verwenden.

4.2.6 KATEGORIE: GO LIVE

BESCHREIBUNG

Die Go-Live Phase ist die Phase kurz vor dem Go-Live, wo noch einige Features fertig implementiert werden müssen, Bugs gefixt werden müssen und ausgiebig getestet werden müssen. Diese Phase ist ziemlich stressig, weil meistens der Go-Live Termin nicht verschiebbar ist, da es meistens mit weiteren Maßnahmen wie Marketing usw. verbunden ist.

INTERPRETATION

Die Go-Live Phase ist für alle Interviewpartnerinnen und Interviewpartner ein Begriff für eine stressige und anstrengende Phase, wo sehr viel geleistet werden muss, aber auch unerwartete Aufforderungen seitens Kunden eingelastet werden. In dieser Phase neigen sich die Meinungen für Non-Timeboxing Ansätze. Dies aus dem Grund, weil sehr viel in kurzer Zeit erwartet wird und man mit Timeboxing nicht alles im Griff bekommen würde. Deshalb tendieren sehr viele Teams dazu, in diese Phase, wo sehr viele Defekts gelöst werden müssen, wo die letzten Features programmiert und auch ausgiebig getestet

werden müssen, zu Non-Timeboxing Ansätze zu wechseln, und sich als Ziel setzen alle diese Aufgaben bis zur vorgegebenen Deadline abzuarbeiten.

4.2.7 KATEGORIE: PROJEKTART

BESCHREIBUNG

Die Kategorie „Projektart“ beschäftigt sich mit der Art des Softwareprojekts. Diese Kategorie lässt sich in vier Projektarten unterteilen:

1. Bekannte Projekte: Projektarten, die dem Team gängig und bekannt sind, wo keine großartige Forschung für die Herangehensweise benötigt wird und wo das Team auch an ähnlichen Projekten gearbeitet hat.
2. Pilot-, Forschungsprojekte und Prototypen Entwicklung: Pilot- und Forschungsprojekte sind Projekte, die ziemlich neu sind, wo neue Technologien ausprobiert werden und getestet wird, ob es Sinn macht sie einzusetzen oder nicht. Prototypenentwicklung ist die Entwicklung von einem vereinfachten Versuchsmodell eines geplanten Produktes.
3. Support-, Wartungs- und Betriebsprojekte: sind Projekte, die sich mit allen Maßnahmen für die Instandhaltung der Software beschäftigen.
4. Produkte und Projekte: Ein Produkt ist ein firmeninternes Produkt, dass von der Firma finanziert wird, und dann verkauft, oder als „Software as a Service“ (SaaS) angeboten wird. Projekte werden meistens im Namen eines Auftraggebers entwickelt werden.

INTERPRETATION

Viele der Interviewer haben schon bei der initialen Frage nach Auswahlkriterien für Timeboxing und Non-Timeboxing Ansätze angeführt, dass die Projektart bzw. das Projektsetting ein Hauptentscheidungskriterium für die Beantwortung dieser Frage ist. Desweiteren waren sich alle interviewten Personen einig, dass man die Projektart für diese Frage auf keinen Fall ausschließen könnte.

Sieht man sich die erste Unterkategorie „bekannte Projekte“ an, sprich Projekte, die öfters auf dieser Art und Weise vom Team implementiert wurden, wird Timeboxing als vorzuzuziehende Lösung von den Befragten gesehen. Denn das Team kennt die Rahmenbedingungen solcher gängigen Projekte, kennt die Prozesse und die anzuwendenden Frameworks und muss nicht nach neuen Lösungsansätzen forschen, welche unberechnete Zeit mit sich bringen könnte.

Bei der zweiten Unterkategorie „Pilot-, Forschungsprojekte und Prototypen Entwicklung“ ist die meist vertretende Meinung der befragten Personen, dass es besser wäre Non-Timeboxing Ansätze anzuwenden. Dies aus dem Grund, weil bei Forschungs- und Pilotprojekten, sowie bei Prototypen Entwicklung, sehr viel geforscht und getestet wird. Man probiert neue Technologien aus, generiert neue Ideen, entwickelt und verfeinert diese. Aufgrund dessen kann man schwer solche Projekte in Timeboxing Ansätze angehen, da man sie schwer vorausplanen kann. Der Interviewer B2 war allerdings der Meinung, dass man bei solchen Projekten ebenso Timeboxing Ansätze anwenden sollte, damit man recht schnell einen ersten Prototyp auf die Beine stellt und es sich nicht in die Länge zieht. Allerdings ist Interviewer B6 der Meinung, dass man einen Non-Timeboxing Ansatz benutzen sollte und sich Meilensteine setzen sollte, damit man in absehbarer Zeit ein erstes Ergebnis erzielt.

Bei der dritten Unterkategorie „Support-, Wartungs- und Betriebsprojekte“, sind sich alle Interviewbeteiligten einig, dass die Non-Timeboxing die sinnvollste Variante wäre. Bei solchen Projekten werden eben viele Tickets von Kunden und Dritt-Partner eingelastet und daher kann man schwer vorausplanen. Es wird ebenso erwartet, die aufgetretenen Probleme möglichst schnell zu lösen und den Kunden zu supporten, was auch schwer zum Vorausplanen wäre.

Bei der letzten Unterkategorie „Produkte und Projekte“ waren sich alle Befragten ebenso einig, dass ein Timeboxing Ansatz die sinnvollste Lösung wäre. Sobald man einen Zeitplan, einen Ablauf des Projekts oder des Produkts hat und alle Rahmenbedingungen des Vorausplanes gegeben sind, macht es Sinn Timeboxing Ansätze zu verwenden, um die Vorteile von Timeboxing zu genießen. Vor allem macht Timeboxing auch Sinn, bei

Förderprojekte, wo gewisse Phasen erreicht werden müssen oder bei Projekten, wo es eine Deadline gibt, die auf keinen Fall überschritten werden darf.

4.2.8 KATEGORIE: QUALITÄT

BESCHREIBUNG

Die Kategorie „Qualität“ behandelt die Qualität der Software, sprich Aspekte wie das Testing und das Refactoring, die die Qualität der Software senken oder steigern können.

INTERPRETATION

Für fünf von den sechs Interviewpartnern ist die Qualität ein Entscheidungsmerkmal, um sich für Timeboxing oder Non-Timeboxing zu entscheiden. Dies aus dem Grund, dass mehr Zeit, mehr Qualität bedeutet. Sprich man kann sich mehr Zeit lassen für die Implementierung und man achtet nicht nur darauf, dass es irgendwie funktioniert, sondern, dass man die Features ausgiebig testet, dass man sich Zeit für das Refactoring lässt und dass man vor allem Probleme tiefgründiger analysiert und nicht nur Workarounds programmiert, die die Probleme nur kurzfristig lösen. Somit ist es bei Timeboxing Ansätzen aufgrund des Zeitdrucks kaum möglich, sich diesen Aufgaben zu widmen, wodurch die Qualität drunter leiden kann. Deswegen würden 5 von den 6 Interviewpartner Non-Timeboxing bevor ziehen.

Nach der Meinung von B3 kann man dies nicht pauschal behaupten, weil man sich bei Non-Timeboxing auch nicht ewig für die Qualität Zeit lassen kann.

4.2.9 AUSWERTUNG UND INTERPRETATION ALLER KATEGORIEN

Nachdem jede Kategorie einzeln interpretiert und erläutert wurde, wird in diesem Unterkapitel versucht, ein Gesamtbild der Ergebnisse aller Kategorien zu geben, um vor allem die Forschungsfrage, „Welche Auswahlkriterien sind entscheidend für Timeboxed und Non-timeboxed Ansätze?“, zu beantworten. Die Auswahlkriterien entsprechen hier den Kategorien.

Die erste Kategorie „Überblick über das Projekt“ spricht rein für Timeboxing, da durch Timeboxing beide Parteien (Auftraggeber und Auftragnehmer) einen genauen Überblick über das Projekt haben und besser ihre Ressourcen planen können.

Bei der zweiten Kategorie „Psychische Effekte“ erkennt man ganz schnell, dass alle Interviewpartner nur von Timeboxing Ansätze sprechen und einerseits den Nachteil des zeitlichen Drucks bei Timeboxing Ansätzen sehen, wenn die Iterationen überplant werden. Allerdings kann dieser negative Effekt durch die aktive Teilnahme an den Iterationsplanungen beseitigt werden, indem man als Team einen gewissen Puffer miteinberechnet, um diesen Druck Faktor zu beheben. Dieser zeitliche Druck ist bei Non-Timeboxing Ansätzen nicht gegeben. Andererseits wird der Motivationsfaktor erwähnt, der sich auch nur bei Timeboxing (nicht bei Non-Timeboxing) spüren lässt. Dies aber unter der Bedingung, dass der erste negative Aspekt des zeitlichen Drucks nicht existiert. Somit wird das Team motiviert indem es den Erfolg der letzten Iterationen wahrnimmt und motivierend in die nächsten Iterationen geht.

Die Kategorie „Innovation“ spricht eher für Non-Timeboxing Ansätze, weil man in einem Non-Timeboxing Ansatz keinen zeitlichen Druck hat, der einen einschränkt, innovative Ideen und Lösungen vorzuschlagen und auszuprobieren. Somit wird einerseits die Kreativität des Teams gefördert und andererseits hat man die Zeit sich in den eigenen Schwerpunkten weiter und auszubilden.

Die Kernaussage der vierten Kategorie „Teamsetting“ ist, dass Timeboxing für ein stabiles Team, mit dem man planen kann, vorteilhaft wäre und Non-Timeboxing eher dann besser wäre, wenn das Team instabil ist. Instabil ist das Team, wenn viele Mitarbeiterinnen und Mitarbeiter des Teams nicht 100% im Projekt sind und viele weitere

Aufgaben haben, die in der Planung nicht berücksichtigt werden und wenn das Team aus vielen Aushilfskräften besteht, die die Planung schwer machen. Genauso waren einige Interviewer der Meinung, dass die Zusammenarbeit von Sub-Teams für einen Timeboxing Ansatz spricht, da man sonst voneinander abhängige Aufgaben nicht managen kann.

Die Kategorie „Velocity und Iterationsdauer“ spricht für Timeboxing, wenn man die Velocity ermitteln kann und man eine genau definierte Iterationsdauer von bis zu zwei Wochen hat. Kann man allerdings die Velocity aufgrund der Teamsituation kaum ermitteln, weil das Team sehr instabil ist, oder man hat Iterationen, die länger als zwei Wochen dauern, dann würden Non-Timeboxing Ansätze aufgrund der Nichtplanbarkeit bevorzugt werden.

Die Kategorie „Go-Live“ ist die Phase, die mit sehr viel Stress verbunden ist. Für diese kritische Phase neigen sich die Meinungen der Interviewpartnern dazu, einen Non-Timeboxing Ansatz anzuwenden.

Bei der Kategorie „Projektart“ sind die Aussagen eindeutig, dass für die Unterkategorien „bekannte Projekte“ und „Produkte und Projekte“ Timeboxing Ansätze bevorzugt werden und dass für die Unterkategorien „Pilot-, Forschungsprojekte und Prototypen Entwicklung“ und „Support-, Wartungs- und Betriebsprojekte“ Non-Timeboxing Ansätze bevorzugt werden.

Die letzte Kategorie „Qualität“ sprach für die meisten Interviewpartnern eher für Non-Timeboxing Ansätze, mit der Argumentation „mehr Zeit bedeutet mehr Qualität“.

4.3 ZUSAMMENFASSUNG

In diesem Kapitel wurde der Forschungsteil dieser Masterarbeit vollständig dokumentiert. Begonnen wurde mit der Vorstellung der Forschungsfrage, danach wurde die Vorgehensweise genauer erläutert und die Auswahl der qualitativen Forschungsmethode „Interview“ begründet. Daraufhin wurden die Interviews mit den Interviewpartnern durchgeführt und ausgewertet. Als Ergebnis hat sich die das endgültige Kategoriensystem

gebildet, welches dann mithilfe der Aussagen der Interviewpartnern interpretiert wurde, um die Forschungsfrage zu beantworten.

5 ZUSAMMENFASSUNG UND AUSBLICK

In dieser Masterarbeit wurde das Thema der Zeit, nämlich das Timeboxing und das Non-Timeboxing in agilen Software Projekten aufgegriffen und behandelt. Die Zeit ist ein wichtiger Faktor, welcher in der Unternehmenswelt sowohl für den Auftraggeber, als auch für den Auftragnehmer immer mehr an Bedeutung gewinnt. Egal ob es eine Lieferung, ein Deployment oder ein Service eines Softwareprodukts ist, die Zeit spielt in allen geschäftlichen Geschehen eine wichtige Rolle. Manche Softwareaufträge müssen in einem gewissen Zeitrahmen erledigt werden (timeboxed), andere wiederum haben keinen Zeitdruck für die Lieferung des Endprodukts oder können gar nicht eingeschätzt werden (non-timeboxed). Und hier kommt die Forschungsfrage dieser Masterarbeit auf: „Welche Auswahlkriterien sind entscheidend für Timeboxed und Non-timeboxed Ansätze?“.

Im ersten inhaltlichen Kapitel dieser Arbeit wurde versucht, einen allgemeinen Überblick über das Thema „Softwareprozesse“ zu geben. Dabei wurde auf die beiden bekannten Ansätze, nämlich den klassischen und den agilen Ansatz näher erläutert und die Unterschiede gezeigt. Daraufhin wurden die Werte, das agile Manifest und einige Prinzipien der Agilität verständlich gemacht und einige agile Prozesse kurz angesprochen und die Unterschiede zwischen Timeboxing und Non-Timeboxing in Bezug auf agile Softwareprojekten verdeutlicht.

Im nächsten Kapitel wurde der Begriff „Timeboxing“ und mit dem Begriff verbundenen Themen, wie „Meilensteine“ und das „Pareto-Prinzip“ näher beschrieben. Des Weiteren wurden mithilfe von Literaturwerken einige Vorteile für die beiden Ansätze „Timeboxing und Non-Timeboxing“ definiert. Aufgrund dessen, dass sehr wenig Literatur über das Thema „Timeboxing und Non-Timeboxing in Softwareprojekten“ vorhanden ist, hatte man nicht die Möglichkeit, die Forschungsfrage, nämlich „Welche Auswahlkriterien sind entscheidend für Timeboxed und Non-timeboxed Ansätze?“, durch Literaturwerken näher zu erläutern, weshalb diese Frage im nächsten Kapitel „Forschungsteil“ behandelt wurde.

Im letzten Kapitel dieser Masterarbeit hat der Forschungsteil stattgefunden. Hier wurde die Vorgehensweise bei der Durchführung der qualitativen Forschungsmethode „Interview“ dokumentiert und auch das Forschungsdesign begründet. Nachdem die Interviews mit den Interviewpartnern durchgeführt wurden und nach der Auswertung der Interviews nach dem Vorgehensmodell von Mayring bei der Durchführung von qualitativen Inhaltsanalysen, wurden die Ergebnisse interpretiert und beschrieben. Durch die Interpretation und die Beschreibung der Ergebnisse, wurde die Forschungsfrage: „Welche Auswahlkriterien sind entscheidend für Timeboxed und Non-timeboxed Ansätze?“ beantwortet. Die Auswahlkriterien sind nochmal in der unteren Tabelle zu sehen. Die Kategorien der Tabelle entsprechen den Auswahlkriterien.

Nr.	Kategorie	Unterkategorie
1	Überblick über das Projekt	
2	Psychische Effekte	2.1 zeitlicher Druck 2.2 Motivation
3	Innovation	3.1 Kreativität 3.2 Weiterentwicklung
4	Teamsetting	4.1 Erfahrung 4.2 Anstellungsart 4.3 100% in einem Projekt 4.4 Teamgröße 4.5 Sub-Teams
5	Velocity und Iterationsdauer	5.1 Velocity

		5.2 Iterationsdauer
6	Go Live	
7	Projektart	7.1 bekannte Projekte 7.2 Pilot-, Forschungsprojekte und Prototypen Entwicklung 7.3 Support-, Wartungs- und Betriebsprojekte 7.4 Produkte und Projekte
8	Qualität	

Tabelle 3: endgültiges Kategoriensystem

Die Zuordnung der Auswahlkriterien zu Timeboxing und Non-Timeboxing Ansätzen befindet sich in dem Unterkapitel „4.2.9 Auswertung aller Kategorien“.

6 **ABBILDUNGSVERZEICHNIS**

Figure 1: Phasen des klassischen Wasserfall Modell (in Anlehnung an Riehle, 2006)....	5
Figure 2: Agiler Entwicklungsprozess (o. V., 2018)	6
Figure 3: Scrum in der Rugby Sportart (Rugby, 2011).....	15
Figure 4: Übersicht Scrum Entwicklungsprozess (Mitchell, 2015).....	17

7 **TABELLENVERZEICHNIS**

Tabelle 1: Übersicht über die interviewten Personen	27
Tabelle 2: endgültiges Kategoriensystem	32
Tabelle 3: endgültiges Kategoriensystem	49

8 **ABKÜRZUNGSVERZEICHNIS**

Aufl.	Auflage
bzw.	beziehungsweise
etc.	et cetera
max.	maximal
MBA	Master of Business Administration
min	minimal
o. J.	ohne Jahr
o. V.	ohne Verfasser
S.	Seite
usw.	und so weiter
vgl.	vergleiche
z.B.	zum Beispiel
PC	Private Computer
PO	Product Owner
SaaS	Software as a Service
SDQA	Software Delivery and Quality Assurance
SM	Scrum Master
US	User Story
XP	Extreme Programming

9 QUELLENVERZEICHNIS

Behrmann, Pascal (2008): Konzeption und Evaluierung eines agilen Entwicklungsprozesses für Einzelentwickler am Beispiel eines Unterstützungssystems für Autoren, [online] http://edoc.sub.uni-hamburg.de/haw/volltexte/2008/660/pdf/Behrmann_SoDa.pdf [01.07.2019].

Boograard, Kat (2019): Was ist Timeboxing?, [online] <https://blog.toggl.com/timeboxing/> [25.09.2019].

Botta, Christian, Bahr Ines (2019): Kanban vs Scrum: Welche Methode ist besser?, [online] <https://www.capterra.com.de/blog/558/kanban-vs-scrum> [24.09.2019].

Cron, Tobias (2013): Agiles Projektmanagement in der IT mit Kanban, [online] https://monami.hs-mittweida.de/frontdoor/deliver/index/docId/4030/file/BA_Tobias_Cron_MK07w2.pdf [01.07.2019].

Diehl, Andreas (o. J.): Agiles Arbeiten – Du kannst es nicht erklären, Du musst es erleben, [online] <https://digitaleneuordnung.de/blog/agiles-arbeiten/> [20.05.2019].

Inamdar, Anand (2018): The Advantages of Timeboxing to Increase Productivity?, [online] <https://upraise.io/blog/timeboxing-increases-productivity/> [25.09.2019].

Lucidchart, Content Team (2019): A Guide to Timeboxing, [online] <https://www.lucidchart.com/blog/what-is-timeboxing> [25.09.2019].

Mitchell, Ian, (2015): File:Scrum Framework.png, [online] https://commons.wikimedia.org/wiki/File:Scrum_Framework.png [02.06.2019].

Michl, Thomas (2018): Agiles Manifest: 12 Prinzipien für eine agile Verwaltung, [online] <https://www.borisgloger.com/blog/2018/08/23/agiles-manifest-12-prinzipien-fuer-eine-agile-verwaltung/> [29.05.2019].

- Nissen, Volker (2013): IT-Agilität als strategische Ressource im Wettbewerb, [online] https://www.researchgate.net/profile/Volker_Nissen/publication/259082894_IT-Agilitat_als_strategische_Ressource_im_Wettbewerb/links/55db556108aeb38e8a8b84f6/IT-Agilitaet-als-strategische-Ressource-im-Wettbewerb.pdf [20.05.2019].
- o. V. (2018): Inkrementelles Vorgehensmodell, [online] https://de.wikipedia.org/wiki/Inkrementelles_Vorgehensmodell [20.05.2019].
- o. V. (2019): Was ist Timeboxing?, [online] <https://t2informatik.de/wissen-kompakt/timeboxing/> [01.07.2019].
- . o. V. (o. J.): What is Timeboxing in Scrum?, [online] <https://www.visual-paradigm.com/scrum/what-is-time-boxing-in-scrum/> [25.09.2019].
- Pöpel, Benjamin (2014): Agile Softwareentwicklung, [online] https://hps.vi4io.org/_media/teaching/sommersemester_2014/siw2014_poepel_agile_arbeit.pdf [01.07.2019].
- Riehle, Jan-Philip (2006): Klassiker des Projektmanagements: Wasserfall-Modell vs. agiles Vorgehen, [online] <https://www.pinuts.de/blog/webstrategie/projektmanagement-wasserfall-gegen-scrum> [20.05.2019].
- Rugby, Army (2011): West Point - The U.S. Military Academy, [online] https://www.flickr.com/photos/west_point/5506094415 [01.06.2019].
- Schönfeld, Patrick (2017): Die 5 Werte von Scrum, [online] <https://chaosverbesserer.de/blog/2017/02/12/die-5-werte-von-scrum/> [01.06.2019].
- Schwaber, Kent, Sutherland Jeff (2016): The Scrum Guide, [online] <https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-German.pdf> [01.06.2019].

10 LITERATURVERZEICHNIS

- Eckstein, Jutta (2004): *Agile Software Entwicklung im Grossen*, 1. Aufl., Heidelberg: dpunkt.verlang GmbH.
- Flick, Uwe, Kardorff Ernst und Steinke Ines (2000): *Qualitative Forschung. Ein Handbuch*, 13. Auflage, Reinbek bei Hamburg: Rowohlt.
- Gloger, Boris, Margetich Jürgen (2014): *Das Scrum-Prinzip Agile Organisationen aufbauen und gestalten*, 1. Aufl., Stuttgart: Schäffer-Poesschel Verlag.
- Gloger, Boris (2016): *Scrum Produkte zuverlässig und schnell entwickelt*, 5. Aufl., München: Carl Hanser Verlag.
- Leopold, Klaus, Kaltenecker Siegfried (2013): *Kanban in der IT*, 2. Aufl., München: Carl Hanser Verlag.
- Mayring, Philipp (2008): *Qualitative Inhaltsanalyse. Grundlagen und Techniken*, 10. Auflage, Weinheim/Basel: Beltz Verlag.
- Pichler, Roman (2008): *Scrum Agiles Projektmanagement erfolgreich einsetzen*, 1. Aufl., Heidelberg: dpunkt.verlang GmbH.
- Royce, Winston W. (1970): *Managing the Development of large Software Systems*, In: Ieee Wescon, Nr. August.
- Strasser, Shinja (2012): *Teamarbeit und Teamentwicklung im Umfeld des agilen Projektmanagements*, Simbach.
- Wolf-Gideon, Bleek, Henning Wolf (2008): *Agile Software-Entwicklung Werte Konzepte und Methoden*, 1. Aufl., Heidelberg: dpunkt.verlang GmbH.
- Vigenschow, Uwe (2015): *Der APM-Guide zu »APM – Agiles Projektmanagement«*, 1. Aufl., Heidelberg: dpunkt.verlang GmbH.

11 ANHANG

INTERVIEW-TRANSKRIPTE

BEFRAGTER 1

1 -00:00:01

2 -I: Guten Abend Herr Actors möchten sie sich kurz vorstellen.

3 -00:00:05

4 -B1: Guten Abend Danke für die Einladung. Mein Name ist Actors. Ich bin derzeit als Service Delivery Manager für die Departments Customer Service and Operations beschäftigt, bin seit drei Jahren in dieser Branche. Davor habe ich auf der FH Joanneum Internet Technik studiert und erfolgreich abgeschlossen. Und Derzeit studiere ich Wirtschaftsinformatik berufsbegleitend auf der Campus 02. Ich bin 24 Jahre alt und wie gesagt seit drei Jahren in einem E-Commerce-Softwareunternehmen tätig.

5 -00:00:59

6 -I: Vielen Dank für die Teilnahme am Interview und für die Vorstellung. Ich möchte Ihnen kurz die Forschungsfrage präsentieren warum ich sie hier interviewte. Und zwar geht es bei meiner Masterarbeit eben darum, dass ich danach forsche wann man Timeboxing bzw. Non-Timeboxing Ansätze einsetzt. Also welche Auswahlkriterien sind entscheidend für Timeboxed und Non-timeboxed Ansätze in der Softwareentwicklung?

7 -00:01:22

8 -I: Ich möchte Ihnen noch kurz darüber informieren über die anonyme Behandlung der Daten. Natürlich werden die Daten nicht an Dritte weitergegeben und möchte dementsprechend auch ihr Einverständnis einholen.

9 -00:01:48

10 -B1: Ich bin damit einverstanden.

11 -00:01:50

12 -I: Haben Sie noch Fragen bevor wir starten?

13 -00:01:52

14 -B1: Nein ich freue mich auf Interviewfragen beginnen wir mit der ersten Frage.

15 -00:02:00

16 -I: Die erste Frage lautet Was ist Ihr höchst abgeschlossener Ausbildung und welche Berufserfahrung bringen sie im Softwarebereich mit.

17 -00:02:08

18 -B1: Ich habe wie vorhin erwähnt Bachelor of Science bereits abgeschlossen und bin derzeit mit meiner Masterarbeit beschäftigt. Wenn alles gut läuft werde ich ab Februar nächsten Jahres mein Diplom abschließen. Bringe mit mir drei Jahren Berufserfahrung in einem Softwareunternehmen in der Softwareentwicklung und auch Software Wartung und Software Betrieb.

19 -00:02:49

20 -I: Gut, dann zur nächsten Frage. Nutzen Sie in ihre Software Projekt agile Entwicklungsmethoden.

21 -00:02:55

22 -B1: Ja selbstverständlich.

23 -00:02:58

24 -I: Und welche sind diese Software-Entwicklungsprozesse?

25 -00:03:01

26 -B1: Während Software Projekte gibt habe ich schon während der FH-Phase betreut und war eben auch im Software Projekten tätig. Da haben wir sehr stark Scrum benutzt. In meinem heutigen Job seit drei Jahren bin ich im Kanban Prozess entwickelt und nutzen eben sehr stark Kanban.

27 -00:03:26

28 -I: Alles klar. Gut gleich zur nächsten Frage. Vielleicht jetzt auch auf die Forschungsfrage bezogen, bezogen auf die Zeit, also Time Boxing. Was sind Ihrer Meinung nach die Auswahlkriterien warum ein Projektmanager wie Sie jetzt zum Beispiel für ein Projekt Timeboxing oder Non-Timeboxing Ansätze bevorzugen soll.

29 -00:03:55

30 -B1: Gut, gibt es einige Gründe warum das eine oder das andere. Ich denke, für Timeboxing entspricht, dass wir dem Kunden gegenüber einem zeitlichen Überblick schaffen und dass auch garantieren so gut wie möglich, dass somit auch dem Kunden die Sicherheit gibt auch die Kosten-technischen Fragen für Kunde schon am Anfang des Projektes eine Klarheit schafft. Das ist einmal ein wichtiger Faktor für die Kunden damit er überhaupt weiß wie lange das Projekt dauern wird und mit welchem Aufwand er rechnen muss. Das ist einmal einer der Hauptverteiler, die es gibt. Andererseits kann auch auf der anderen Seite der Auftragnehmer in dem Fall, das Unternehmen die Software entwickelt auch mit ihren eigenen Ressourcen rechnen. In dieser Phase habe ich so viele Leute eingestellt und somit ist die Ressource Planung des Unternehmens auch leicht handzuhaben. Was das Non-Timeboxing angeht gibt es wiederum auf der anderen Seite den Nachteil für Kunden, dass er vielleicht hier keinen Überblick hat. Aber in diesem Fall ist es wichtig für das Unternehmen, dass sie hier eine gewisse Freiheit hat auch die Mitarbeiter eine gewisse Freiheit haben die das Projekt ein Produkt zu entwickeln. Es hängt aber sehr stark davon ab was für ein Projekt, das ist, ob das ein Projekt ist Software Projekt ist das üblicherweise im Unternehmen immer wieder entwickelt wird oder dass es mit neuen Features mit neuen Projekten Stichwort Pilotprojekten zu konfrontieren ist. Da müsste man dann noch einmal anschauen welche Methode Sinn macht. Es kommt auf

das Projekt Art an. Es kommt auf die Kundenwünsche an, auf die zeitliche Ressourcen, auf die personelle Ressourcen, die hier angewendet werden sollen.

31 -00:06:12

32 -I: Alles klar vielen Dank.

33 -00:06:15

34 -I: Vielleicht gleich zur nächsten Frage Inwiefern kann die Rolle der Timeboxing bzw. Non-Timeboxing ein Einfluss auf die Psyche der Mitarbeiter haben. Dieser psychische Effekt. Gibt es Gründe wo Sie sagen, Okey ich nimm jetzt das eine oder das andere. Welches der beiden bringt Vorteile mit sich für die Psyche des Mitarbeiters?

35 -00:06:34

36 -B1: Wie vorhin gesagt Timeboxing hat sicher den Vorteil, dass es vielleicht kostengünstiger ist vielleicht auch einen Überblick schafft. Aber auf der anderen Seite muss man schon sagen, wenn man von Timeboxing spricht meistens ist es so dass man immer einen engen Zeitraum definiert. Das ist für die Mitarbeiter oder für das gesamte Team das Projektteam, Ist es ein zeitlicher Druck das die Kreativität der Mitarbeiter einschränken könnte. Es ist ein zeitlicher Druck, dass die Motivation der Mitarbeiter sinken könnte. Es ist ein zeitlicher Druck, dass auch die Freiheit des Mitarbeiters auf die Umsetzung des Features einschränken wird. Der Mitarbeiter wird vielleicht nie auf die Idee kommen neues bzw. neue Technologien anzuwenden, weil er diese Zeit vielleicht nicht hat. Dementsprechend hat man diese Freiheit nicht und das ist schon ein großer Faktor. Und vor allem in den kritischen Projekten, die dann auch eskaliert werden an das Managementteam ist generell auch der Wunsch in die Arbeit zu gehen manchmal sogar nicht da. Das Mitarbeiter sagen das ist es für mich schon zu viel und ich würde ehrlich gesagt am besten jetzt aufhören. Das kommt auch sehr oft vor. Auch in der Firma wo ich arbeite habe ich das öfters gehört. Man hat einen Go-Live Termin definiert und es müsste bis zu den Go-Live wirklich alles fix und fertig sein. Aber man müsste ja dann schauen, Ich kann ich. Ich habe da Beispiele wo auch die Mitarbeiterinnen und Mitarbeiter bis in

der Nacht gearbeitet haben. Aber dann in dem Team schon zu Kündigungen kam, freiwillige Kündigungen, weil sie einfach nicht mehr mit diesem Druck leben konnten. Auf der Non-Timeboxing hat man natürlich diese Freiheit und kann man sich ein wenig selbst Spielraum schaffen das schon für die Selbstentwicklung des Mitarbeiters eine wichtige Rolle spielt.

37 -00:08:50

38 -I: Vielleicht noch dazu eine Frage Was denken Sie über die Motivation. Wir haben jetzt gehört, dass die Motivation auch senken kann, wenn das Projekt schlecht läuft. Aber wenn man darüber nachdenkt das Projekt läuft gut und man hat eben eine Zeit wo man entwickelst und man sieht auch den Erfolg davon. Denken Sie, dass es nicht eine Motivation bzw. ein Motivationsfaktor für die Mitarbeiter sein kann, das Timeboxing Ansatz

39 -00:09:17

40 -B1: Wenn alles wirklich gut läuft Ja, aber die bisherigen dreijährigen Erfahrungen haben schon gezeigt, dass es in wirklich sehr seltenen Projekten der Fall ist. Was passiert am Ende des Tages Du schlägst der Kunde zum Beispiel fünf Wochen vor für einen Mini Projekt und der Kunde kommt her und sagt Ich will drei Wochen mehr zahle ich nicht und das war's und dann handelt man sich auf spätestens vier Wochen haben. Somit hat man schon eine Woche verloren. Und das zeigt, Es wird nie der Rahmen in vielen Projekten nie den Rahmen geschaffen, dass der Mitarbeiter ohne den extremen Druck hier durchkommen kann. Selbstverständlich wenn diese extremer Druck nicht da ist, ist es natürlich ein Motivationsgrund, hat man gewisse stolz als Team aber auch als Einzelperson. Dass wir in dem vereinbarten Zeitraum das geschafft haben und dazu gibt es immer wieder Zuckerln auch von der Kundenseite dafür, dass man sich dafür einhält. Aber wie gesagt es ist leider sehr selten der Fall im realen Leben.

41 -00:10:26

42 -I: Vielen Dank, zur nächsten Frage gleich. Inwiefern ist die Struktur des Teams wie das Team aufgebaut ist ein Einflussfaktor, um sich für Timeboxing oder Non-Timeboxing Ansätze zu entscheiden.

43 -00:10:39

44 -B1: Die Struktur folgendermaßen Je nachdem welche Leute im Team sitzen, hat man Leute die bereits Senioren sind also Senior EntwicklerInnen Senior sind. Da hat man einen gewissen Flow schon entdeckt auch in der Firma schon. Und man geht diese Phase nach und deswegen sind auch in dem Fall auch Timeboxing Methoden für sie auch einfacher handzuhaben. Das heißt wenn du Seniorinnen EntwicklerInnen Senior Projektleiter hast. Des Weiteren ist es wichtig wie viel kann der Mitarbeiter geben. Ist er wirklich Vollzeit für das Projekt da. Manchmal hat man drei Projekte zu betreuen gleichzeitig und überall zu 50 Prozent zu 65 Prozent. Das spielt auch für den Mitarbeiter einen Faktor, dass er sich immer umsteigen muss von einem Projekt zum anderen er die anderen Features da hat er die anderen Regelungen da hat er die anderen Probleme. Das spielt auch einen Faktor, wenn der Mitarbeiter Vollzeit also wirklich ganz voll dabei ist bei dem Projekt ist es wichtig auch ein verlässlicher Mitarbeiter, wenn er Teilzeit da ist gewisse Tage ist er da gewisse Tage ist er nicht da ist er wiederum problematisch mit ihm in Timeboxing zu arbeiten. Wenn wie gesagt die Berufserfahrung spielt hier eine Rolle wie viele Jahre hat man schon Projekt entwickelt. Seit wie viel Jahren sind man überhaupt in der Firma. Es kann sein, dass er schon seit zehn Jahren Software Projekte entwickelt. Aber es ist halt neu in der Firma und ist neu konfrontiert mit dieser Art und Weise wie die Firma arbeitet. Deswegen spielt auch die Erfahrung in dieser Firma selbst und die nächste Stufe ist dann seit wann ist der Mitarbeiter im Projekt. Manchmal gibt es Projekte wo dann Quereinsteiger gibt also Leute waren von Anfang an nicht bei der Projektentwicklung dabei. Das Software Projekt hat schon begonnen und man hat schon vieles implementiert. Man hat viele schon Released aber dann braucht man externe Ressourcen und nimmt sie mit auf bis diese Leute dann on-boarded werden. Ist diese Onboarding Phase ist für das Projekt nimmt es auch sehr viel Zeit Anspruch und das könnte auch für Timeboxed Ansätze problematisch werden.

45 -00:13:03

46 -B1: Genau, des Weiteren man müsste schauen, sind das fixe Mitarbeiter im Unternehmen oder sind das so externe Mitarbeiter, die für einen bestimmten Zeitraum hier sind. Bei solchen Fällen verlässt man sich nicht voll und ganz auf die Mitarbeiter. Man gibt auch nicht sehr viel denen man weiß Irgendwann werden sie auch weg gehen. Und in vielen Fällen sind sie auch sehr teuer. Externe Hilfe zu holen. Und deswegen ist es dann auch eine Frage der Finanzierung. Und da könnte Timeboxed problematisch werden. Und auch wenn man die Mitarbeiter anschaut, man müsste auch die Ressourcen anschauen was die Mitarbeiter hergeben kann, wenn der Mitarbeiter in der Firma bereits auch andere Rollen hat, wie zum Beispiel Team Lead, Entwicklungslinie, Development LEAD, und so weiter und so fort. Dann kann er auch nicht alles hergeben oder wenn der Projektmanager fünf Projekte leitet und sich nicht auf einem Projekt konzentrieren kann könnte es hier in Timeboxed zu Problemen kommen.

47 -00:14:15

48 -I: Also sie sehen hier, dass bei manchen Strukturen des Teams also wenn nicht optimal strukturiert ist, das dann Non-Timeboxed besser funktionieren würde als Timeboxed.

49 -00:14:28

50 -B1: selbstverständlich, weil hier eben diese zeitliche Freiheit schon noch gerade gibt. Es ist nichts sehr locker, aber es ist auch nicht so streng wie es in Timeboxed ist,

51 -00:14:39

52 -I: Vielen Dank, zur nächsten Frage: Inwiefern kann die Team-Velocity und die Sprint Dauer also Sprint-Umfang Einfluss auf die Entscheidung ob man Timeboxed oder Non-Timeboxed nimmt.

53 -00:14:50

54 -B1: Grundsätzlich muss man darüber reden wie einfach wie schwer ist es zu messen die Geschwindigkeit Velocity des Teams. Das heißt wenn man wirklich geschaffen hat was auch schwierig ist, die Velocity definiert hat und dann auch eine gewisse Sprint-Dauer hat dann kann man sich schon die Timeboxed Methode entscheiden, weil man weiß Man kann den Task mit so und so viele Story Points in so vielen Tagen erledigen. Dann hat man die Timeboxed Methode, die einfach anzuwenden ist. Jetzt ist es aber so dass es selbstverständlich im Projekt zu unerwarteten Vorfällen kommen könnte. Was die Ressourcen angeht. Das würde heißen Die Kunden haben nicht mehr das Geld und senken die Anzahl der Mitarbeiter für das Team. Die Mitarbeiterinnen und Mitarbeiter im Team sind ausgefallen aus verschiedenen Gründen Krankheit, Pflegeurlaub, etc. die man nicht damit rechnet und auch im nächsten Fall, ich mein, man kann schon in einem gewissen Raum rechnen wann der Mitarbeiter Urlaub geht. Man kann die Planung des Urlaubs vielleicht drei Monate vorher auch schon einkalkulieren, aber manchmal kommt es auch dass der Mitarbeiter spontan Urlaub nehmen musste selbst drei Wochen vorher. Und das hat auch einen Einfluss darauf auf den Team Velocity deswegen ist es sehr schwer vorherzusagen was das Team Velocity wirklich sein kann. Man kann ungefähr sagen was aber wiederum nicht genug ist, um Timeboxed richtig einzusetzen. Deswegen ist auch die Länge des Sprints sehr wichtig. Derzeit ist es in vielen Projekten so dass man mit zwei Wochen rechnet und zwei Wochen sind durchaus unschaubare Tage, also sehr viele Tage, die schwer zu handhaben sind. Wenn man heute von einem Sprint redet in drei Tage Zyklus dann kann man das leichter handhaben und leichter definieren okay in drei Tagen kann ich das und das, umsetzen aber wenn du zwei Wochen hast und in zwei Wochen kann vieles passieren kann vieles auf dich zukommen. Ist es schon schwer erst mal die Velocity zu tracken und auch zu festlegen und dann im nächsten Schritt anhand der Ergebnisse vom Velocity Tracking, dass man sagt ich brauche so und so viele Tage und nehme mir diesen Timeboxed Ansatz. Deswegen nochmal zusammengefasst, eher schwierig das Team Velocity zu definieren in erster Linie. Wenn wir wirklich einen überschaubaren Zeitrahmen haben kann man dann hier auch die Timeboxed Methode anwenden was dann auch vorteilhaft ist.

55 -00:18:08

56 -I: Vielen Dank für diese ausführliche Erklärung, zur nächsten Frage: Inwiefern kann die Entscheidung davon abhängen die Entscheidung Timeboxed oder Non- Timeboxed ob man in einer Go-Live Phase ist. Sie haben auch das Wort vorhin auch erwähnt. Was ist Ihre Meinung dazu.

57 -00:18:30

58 -B1: Es ist schwierig, weil von Anfang an hast du am Beginn des Projektes wahrscheinlich definiert welchen Ansatz Du nimmst. Du sagst du hast am Anfang des Projektes dich für Timeboxed Einsatz entschieden und will das auch durchsetzen. Das hast du auch so den Kunden kommuniziert und hast auch einen gewissen Road-Map eine gewisse Timeline definiert. Deswegen in der Go-Live Phase dann, das heißt je nachdem wie viel die Entwicklung gedauert hat. Diese Flexibilität zu haben zu entscheiden nehme ich jetzt Timeboxed oder Non-Timeboxed ist auch wirklich sehr Kunden abhängig bzw. auch Team abhängig.

59 -00:19:28

60 -B1: Optimal ist es das in der Go-Live Phase, das heißt kurz vor der Stabilitätsphase, es hat einen internen Go-Live gegeben und jetzt bin ich am Fixieren von Defekt etc. etc. Macht es Sinn, dass man hier einen Non-Timeboxed Ansatz verwendet, weil man auch nicht unbedingt damit rechnen kann wie viel Defekts zum Beispiel auf mich zukommen oder was muss noch schnell angepasst werden, was sind noch schnell die Wünsche von Kunden. Weil in dem Fall in der agilen Entwicklungsansatz. Man muss ja auch eine gewisse Flexibilität dem Kunde gegenüber und der Kunde vor allem in der Phase spielt sehr oft mit verschiedenen Tools und spielt sehr oft mit verschiedenen neuen Anforderungen.

61 -00:20:17

62 -B1: Deswegen wenn es die Möglichkeit besteht macht es schon Sinn zum Beispiel in der Go-Live Phase mit Kanban zu arbeiten weil wie gesagt da hast du dann deine Tasks

die reinkommen die gewissen Ressourcen die dafür stehen und die fixen dann dass damit dann die Go-Live Phase stabiler läuft.

63 -00:20:35

64 -I: Du hast gemeint, dass in der Phase Defekte und Bugs reinkommen könnten. Aber es ist ja nicht nur so dass Bugs reinkommen, sondern die Go-Live Phase ist, wenn man davon spricht ist es ein Monat oder vielleicht zwei Monate vor dem Go-Live und da ist das Team eben auch dabei die Features noch fertigzustellen. Kann das auch ein Problem sein. Bei der Entwicklung der Features, dass man sagt Soll ich jetzt Timeboxed oder eher Non- Timeboxed nehmen. Also die Phase vor dem Go-Live, kurz davor.

65 -00:21:02

66 -B1: Es ist sehr schwer in einer Timeboxed Variante zu arbeiten, wenn es kritisch wird. Kurz vor der Umsetzung der Features. Das bedeutet Es hat Probleme gegeben und es wird kritisch mit der Zeit. Dann müsste man mehr tun als Timeboxed. Man müsste viel mehr geben und hier müsste man schauen, Timeboxed schafft ja auch eine Möglichkeit, dass der Mitarbeiter selbst weiß wie viel Zeit geben muss und wie viel er wie viele Stunden er jetzt noch geben muss. Aber kurz vor der Umsetzung eines Features das kritisch ist schafft man das nicht was auch vertraglich definiert ist. Manchmal muss man über den Rahmen gehen, um den Kunden das zu geben was auch versprochen wurde. Deswegen ist es sowieso sehr schwierig überhaupt zu sagen Wir arbeiten hundertprozentig agil. Ich glaube das ist in vielen Projekten nicht möglich. Man muss dann die alten Ansätze hernehmen und richtig hakeln damit man wirklich noch gerade diese Zeit entspricht und da wirds schwierig mit Non-Timeboxed da wird schwierig mit Timeboxed. Da müsste man einfach viel mehr geben. Und ich glaube in der agilen Entwicklung hat man auch ein bisschen offenen Raum dafür

67 -00:22:45

68 -I: Vielen Dank, zur letzten Frage dann gleich, welches Projekt Setting ist entscheidend für Timeboxing oder Non-Timeboxing. Also Pilot-Projekt haben Sie mal

angesprochen, ob es ein Produkt, ein Projekt ist ob es ein Firmen Projekt ist oder ein Kundenprojekt ist. Macht es einen Unterschied?

69 -00:23:04

70 -B1: Selbstverständlich. Wenn es zum Beispiel ein Projekt ist was das Entwicklungsteam bisher immer wieder gemacht hat und sich auskennt, wenn gewisse Frameworks angewendet wurden etc., etc. dann hat man eine gewisse Flow und Erfahrung und dann ist man schon mittendrin. Und das läuft. Das heißt Projekte die bereits öfters in der Art und Weise entwickelt wurden. Wie sie angesprochen haben Pilotprojekte sind kritische Fälle das bedeutet Es gibt jetzt weltweit neue Features neue Art und Weise wie es zum Beispiel Micro Services und die nicht mehr den alten Entwicklungsansatz vor allem sondern neu definiert wurden. Und man arbeitet schon mit Micro Services zum ersten Mal etc. Dann hat man hier auch eine Schwierigkeit zu sagen was nehme ich. Deswegen musste man sich auch nochmal anschauen da in dem Fall ist Timeboxing halt sehr schwer. Man kann Timeboxing ansetzen, aber man muss sich klar sein, man muss einen gewissen Puffer schaffen für die Leute, dass sie sich einmal einlesen können, dass sie Knowhow aufbauen in dem Pilotprojekt. Wie gesagt in üblichen Projekten, die man bis jetzt hatte, dann ist es möglich die Timeboxing Methode anzuwenden. Wenn du aber zum Beispiel Projekte hast wie es in meinem alltäglichen Job der Fall ist, Support Projekte, Betriebs Projekte wo man auch sehr viel davon abhängt was der Kunde oder was den Dritt-Partner jetzt einlastet, was für Blockers, Inzidenz etc. einlastet werden. Damit kannst du ja nicht rechnen. Das sind Vorfälle, die kommen auf dich zu. Du kannst maximal ein paar Schätzungen machen wie viel auf mich zukommen wird. Aber in solchen Support und Betriebs und Wartungs-Projekten ist eine Timeboxing Ansatz unvorstellbar deswegen wird auch in vielen Support Teams und betrieb Teams Operations-Teams auch Kanban angewendet. Des Weiteren wenn du ein Produkt Entwicklungsunternehmen bist und ein Produkt für dich selbst entwickelt, hast du auf der einen Seite keine Kunde vielleicht der den zeitlichen Druck macht, aber du hast wahrscheinlich das Management-Team, das sagt Ich will zu dem Zeitraum fertig sein und ich will dieses Tool jetzt haben. Das kann schon problematisch werden. Andererseits muss zu schauen, dass du so wenig wie möglich an Ressourcen reinsteckst, weil das wird

ja nicht von Externen bezahlt. Das wird von internen Finanzen Topf vom Budget vom internen Budget finanziert. Gerade bei solchen Projekten schaut man so wenig wie möglich Geld auszugeben. Zusammengefasst Pilotprojekte sehe ich kritisch da müsste man sehr kritisch, kritisch für Timeboxing. Da müsste man im besten Fall auch im Grunde kommunizieren einen Puffer zu schaffen, dass die Mitarbeiter die Mitarbeiter sich einlesen können, Knowhow aufbauen können. Support Betriebs Projekte eher Non-Timeboxing, Produktentwicklung hängt sehr stark davon ab was die Vorgaben sind für dieses Produkt aber eher auch Timeboxing würde ich sagen und üblichen Projektentwicklungen wo man bereits Expertise hat macht es durchaus Sinn Timeboxing anzuwenden.

71 -00:26:52

72 -B1: Vielleicht noch eine Frage, die mir im Nachhinein eingefallen ist, was die Qualität betrifft, spielt die Qualität damit ob mit Timeboxing oder Non-Timeboxing nimmt. Wenn wir jetzt davon reden, dass wir jetzt in einer Phase Go-Live Phase Non-Timeboxing verwenden sollen. Nimmt die Qualität damit ab oder nimmt sie zu. Ist es sinnvoller, wenn man jetzt rein nur Qualität-technisch redet, dass man Timeboxing oder eher nicht.

73 -00:27:14

74 -I: Der zeitliche Druck kann schon sehr viel mitspielen wird schon einen ziemlichen Einfluss haben auf die Qualität, selbstverständlich man kalkuliert Quality Assurance Zeit. Und es wird einen Test-Agent geben, der das testet, eine gewisse Qualität Checklisten durchführt. Aber wenn hier der Druck da ist, konzentriert man darauf, dass es mal funktioniert egal wie es auch funktioniert. Man kann hier Implementierungen treffen, die vielleicht nicht unbedingt sauber sind. Die auf eine langfristige Zeit dann doch nicht unbedingt gescheit implementiert wurden. Hat man in der Timeboxing Ansatz überhaupt Zeit für zum Beispiel Refactoring genommen. Dass man noch einmal durchgeht und durchschaut was man besser implementieren. Das man dann nur mehr kleine Workaround implementiert und nicht wirklich gescheit implementiert und dann kriegt irgendeinen Defekt rein den löst man auch nicht wirklich, sondern man schaut es irgendwie noch

funktioniert aber das dahinterliegende Problem des Defekts analysiert man auch nicht tiefgründig. Weil es gibt keine Ressourcen, es gibt keine Zeit. Deswegen hat das Timeboxing aus meiner Sicht schon einen gewissen Einfluss darauf, dass dann die Qualität doch schlechter ist.

75 -00:28:51

76 -I: Also sehen Sie das man da Non- Timeboxing Ansätze mehr Qualität mit sich bringen könnten als mit Timeboxing.

77 -00:28:59

78 -B1: Ja durchaus wie gesagt da hat man halt den zeitlichen Druck nicht.

79 -00:29:06

80 -I: Perfekt, vielen Dank für das Interview hat mich sehr gefreut. Dankeschön und alles Gute.

81 -00:29:13

82 -B1: Ja vielen Dank.

BEFRAGTER 2

1 -00:00:05

2 -I: Lieber Matthias mag du dich kurz vorstellen.

3 -00:00:07

4 -B2: Alles klar. Mein Name ist Matthias. Ich habe Software-Entwicklung-Wirtschaft studiert, bis zum Master rauf. Danach habe ich selber eine eigene Firma gegründet die mittlerweile schon abgegeben. Und jetzt bin ich schon gute drei Jahre bei Netconomy. Und Ich bin mittlerweile 34 Jahre alt.

5 -00:00:51

6 -I: Super, Ich möchte mich bedanken für die Teilnahme am Interview.

7 -00:00:55

8 -I: Kurz zur meiner Forschungsfrage, die lautet eben: Welche Auswahlkriterien sind entscheidend für Timeboxing bzw. Non-Timeboxing Ansätze. Hast du im Vorhinein noch Fragen?

9 -00:01:15

10 -B2: nein, eigentlich nicht.

11 -00:01:17

12 -I: Dann möchte ich dich noch informieren über die anonyme Behandlung der Daten und möchte damit deinen Einverständnis einholen.

13 -00:01:27

14 -B2: Ja ja, ist gut!

15 -00:01:30

16 -I: Die erste Frage lautet Was ist deine höchst abgeschlossene Ausbildung und welche Berufserfahrung bringst du im Bereich Softwareentwicklung mit.

17 -00:01:41

18 -B2: Ein bisschen was habe ich schon vorweggenommen am Anfang, also die höchste Ausbildung wäre auf der Uni, der Diplomingenieur Softwareentwicklung Wirtschaft. Berufserfahrung ist relativ breit gefächert. Ich habe mittlerweile alle Rollen, die mir irgendwie machen kann habe ich mittlerweile durchgemacht Vom Entwickeln. selber bis hin Produktversion und am Ende SDQA- Manager, also verstärkt im Qualitätsbereich. Ja, bis auf Scrum-Master haben ich alles durch.

19 -00:02:15

20 -I: Alles klar, Zur zweiten Frage gleich, benutzt du in deinen Entwicklungsprojekten agile Entwicklungsmethoden.

21 -00:02:28

22 -B2: Ja, verwenden wir. Das wäre dann Scrum.

23 -00:02:32

24 -I: Falls es so ist wie bei deinem Fall, wurde oder wird Scrum bei euch getimeboxed oder eher nicht.

25 -00:02:37

26 -B2: Das wird bei uns time-geboxt wobei man sich natürlich ein bisschen darüber streiten kann ob es denn wirklich time-geboxt ist oder nicht. Aber im Großen und Ganzen würde ich sagen Ja.

27 -00:02:48

28 -I: gut dann bezogen eben auf die Zeit, also Timeboxing. Was ist deine Meinung nach die Auswahlkriterien warum ein Projektmanagertime-boxt bzw. Non-getimeboxt auswählen soll.

29 -00:03:07

30 -B2: Also ich meine ich bin ja eher ein Verfechter Fan von Timeboxing. Wenn man dann einfach etwas fixer für dieses Projekt was man abschließt einfach damit markiert. Das heißt dass man auf das Projekt gesehen deshalb auch ein bisschen anfangen zu unterteilen und danach auch Erfolgserlebnisse dazwischen hat.

31 -00:03:30

32 -B2: Wenn man das Ganze wie einen großen Teig quasi ansieht, hat man nie einen Erfolg dazwischen, sondern man arbeitet einfach kontinuierlich weiter. Ich würde sagen, dass es was Psychologisches hat, wenn man es Timeboxed macht. Das heißt man hat immer einen Abschnitt betrachtet dem geht es gut oder schlecht beziehungsweise müssen wir zum Beispiel Gas geben oder hätten wir gut gemacht zum Beispiel, um die Qualität zu erhöhen oder ist es vielleicht wichtiger, dass man Arbeitspakete schneller umsetzt. Das sind eher die Sachen, die mich Richtung Timeboxing bringen würde.

33 -00:04:09

34 -I: Und diese Sachen findest du nicht in Non-Timeboxing Ansätze.

35 -00:04:14

36 -B2: Nicht unbedingt. Wobei man dazu sagen muss, es hat ja wahrscheinlich, es gibt gewisse Fälle wo es einfach besser ist, wenn man z.B. direkt an Kanban denkt. Es gibt Situationen wo es durchaus mal etwas mehr Sinn macht in diese Richtung zu gehen. Das meiner Meinung nach gut überlegen ist. Ich würde sagen. Egal welches Framework man jetzt nimmt es muss angepasst sein auf das was wir gemacht.

37 -00:04:42

38 -I: Alles klar.

39 -00:04:46

40 -I: Vorweggenommen von der nächsten Frage Inwiefern kann die Rolle von Timeboxing oder Non- Timeboxing ein Einfluss auf die Psyche der Mitarbeiter haben. Ich meine hier die Effekte wie Motivation Druck Stress. Diese ganzen Aspekte.

41 -00:05:02

42 -B2: Ich glaube ich glaube, weil du grad Motivation gesagt hast. Ich glaube es ist sehr motivierend, wenn man sieht was man geschafft hat. Wenn man das immer wieder kontinuierlich anfangt es zu betrachten. Das funktioniert meiner Meinung nach nur wenn

du dir quasi Scheiben von den Software Projekt runter schneidest und diesen Abschnitt anfängst zu betrachten. Es kann jetzt natürlich auch einen gewissen Druck ausüben, weil wenn du sagst okay ich muss schon so viele schaffen in diesem Abschnitt erzeugt das automatisch bzw. natürlich auch ein Stress im Team. Weil vielleicht schaffst du es nicht, weil es zu viel ist usw. Ich glaube aber, wenn es gut gewählt ist hat man viele Vorteile, wenn man einfach sagt super. Ich habe es geschafft und geht danach besser hinein. Im Vergleich zu Non-Timeboxed.

43 -00:06:06

44 -B2: Da würde ich sagen ist der Stress nicht zu groß. Du hast nicht was Fixes was du eigentlich schaffen muss. Natürlich gibt es aber oft Eine Deadline ist klar, aber auf die Strecke dorthin ist es okay.

45 -00:06:24

46 -I: Also siehst du das eher die Motivation bei Timeboxing liegt und der Stressfaktor weniger bei Non-Timeboxed?

47 -00:06:33

48 -B2: Ja genau, Ich glaube, dass der Druck. Ich glaube, dass der Druck bzw. der Stress ist, der liegt. Da liegt eher bei Timeboxed aber auch sehr motivierend sein kann. Man hängt natürlich immer ein bisschen davon ab was für ein Typ Mensch man ist. Bei manche funktionieren sehr gut. Manche eher nicht.

49 -00:06:53

50 -I: Gut danke für diese Antwort. Zur nächsten Frage: Inwiefern ist die Struktur des Teams wieder wie das Team-Setup aufgebaut ist ein Einflussfaktor, um sich für einer der beiden Ansätze zu entscheiden.

51 -00:07:21

52 -B2: Ich würde sagen es ist für Timeboxing. Ist es immer besser Du hast natürlich fixen Mitarbeitern mit Vollzeit. Kannst du viel besser planen. Wenn du jetzt zieht es speziell auch das Arbeitsumfeld wo ich Arbeit.

53 -00:07:33

54 -B2: Vor einigen Jahren war es noch so dass viele Studenten einfach dabei waren gerade beim Testen von Arbeitspaketen. Und das hat immer dazu geführt, dass man vieles nicht abschließen können, weil du nicht richtig damit planen hast können. Es hängt davon ab Du hast ja Beispiel Teilzeit arbeitet 20 Stunden dann wirst du es immer haben, wenn er mal krank ist und das macht sehr schwierigen einen Arbeitspaket abzuschließen.

55 -00:08:00

56 -I: Also Wer bei einem gemischten Team Setting wäre Non-Timeboxed sinnvoller? Und hängt das von der Berufserfahrung auch ab?

57 -00:08:15

58 -B2: Ja genau, auf jeden Fall. Berufserfahrungen. Ja ich bin ich bin ein sehr großer Vertreter davon Ich glaube, dass Erfahrung gar nicht so eine große Rolle spielt, sondern dass vom Team davon abhängt wie gut es mit diesen Situationen umgehen kann. Erfahrung ist natürlich super. Wenn du wirklich ein Gebiet ein Genie bist. Aber wenn man jetzt so Standard Menschen nimmt und Senior und Junior dann hängt es davon ab meiner Meinung nach wie gut das Team miteinander interagiert. Umso besser die quasi ihre Aufgaben weitergeben können und so besser miteinander kommunizieren können. Umso leichter können Sie das noch abschließen. Ich glaube nicht, dass es so viel davon abhängt wie die Berufserfahrungen ist. Ich meine sicher Es ist schön, wenn du nur Senior sitzen hast. Aber es gibt verschiedene Untersuchungen die eigentlich gezeigt haben, dass es mehr davon abhängt wie das Team miteinander kommuniziert und wie gut sie miteinander umgehen.

59 -00:09:30

60 -I: Spielt es vielleicht auf eine Rolle, wenn man in der Firma auch viele verschiedene Aufgaben, das heißt vielleicht Lead Aufgaben oder andere Aufgaben, die Firmenweit sind und nicht Projektspezifisch sind. Das heißt ich kann nicht 100 Prozent geben, sondern ich muss immer wieder außerhalb des Projekts auch was tun.

61 -00:10:07

62 -I: Das heißt man muss sich entscheiden. Natürlich gibt es manche Sachen wo mehr Priorität ist, aber inwiefern kann das denn Einfluss darauf haben ob ich jetzt ob jetzt ein Projektmanager sagt Entwickler haben teilweise auch Firmen-Aufgaben, die ich nicht so gut abmessen kann. Wie viel Aufwand das sein wird und dadurch, die Entscheidung fällt Timeboxed oder Non-Timeboxed.

63 -00:10:31

64 -B2: Interessant, ist das ist sicher ein schwieriges Kapitel. Je nachdem wie du die Prioritäten setzt werden sich die Leute entscheiden. Wenn ich jetzt sage du hast einen Lead und der muss sich um seine Lead-Aufgaben kümmern hat aber zusätzlich auch noch eine Rolle in diesem Team. Dann ist immer die Frage wie er dies setzt. Wahrscheinlich kannst du auf das Projekt irgendwie sagen Okay 20 Prozent keine Ahnung und 80 Prozent Projekt. Aber das ist immer schwierig. Ich würde sagen es wäre sicher ein Vorteil, wenn man im Non-Timeboxed gehen wird, aber dort hat man nicht so den Stress. Ich glaube bei Timeboxed das ist sehr schwierig, weil du musst Du musst einfach konstant Leistung zeigen und sagen okay ja ich bin jetzt mal einen Monat weg. Ich muss meine Lead-Aufgaben machen. Das stelle ich mir sehr schwierig vor.

65 -00:11:35

66 -I: Alles Klar, danke. Inwiefern können die Team-Velocity und die Sprint-Dauer ob es drei oder vier Wochen ist, ein Einfluss auf die Entscheidung haben.

67 -00:11:51

68 -B2: Naja die Velocity würde ja quasi den Schnitt wiedergeben. Wäre wenn man jetzt zwei Wochen oder vier Wochen wäre das okay. Aber es gibt ja es gibt ja manches. Bei manchen Firmen gibts Sprints, die ein halbes Jahr dauern. Da wird es dann schwer Timeboxed zu verwenden wahrscheinlich, da wärest du Automatisch in Non- Timeboxed. Wobei dann immer die Frage ist kannst du schon nicht mehr, du kannst nicht mehr definieren, weil, es ist so lange, wenn da irgendwo Verzögerungen sind. Sind Sie schon so einen großen Unterschied ausmachen kann kein Teamleiter oder Projektmanager sagen wo sie grad stehen. Insofern wird das auf jeden Fall auf jeden Fall etwas machen. Aber das wäre jetzt halt nur für ganz lange. Ich glaube, dass bei vier Wochen du schon eher dazu tendiert in Richtung Non- Timeboxed zu gehen, weil das schon so lange ist.

69 -00:13:01

70 -I: Ja auch schwer die Krankenstände die Urlaube die Pflege Urlaube mit einzukalkulieren.

71 -00:13:08

72 -B2: Wobei es eigentlich wieder egal sein sollte, weil ich nicht über einen langen Zeitraum sagen wir ja oder so bleibt es ja konstant. Du hast zwar im Normalfall ist es ja so Du weißt okay. Der Winter kommt, die Leute werden eher krank.

73 -00:13:26

74 -I: Also, dass man diese Aspekte auch in die Velocity mit einberechnet.

75 -00:13:30

76 -B2: Genau. Normalerweise würdest du ja tun. Du schaust deine Sprints an, du schaust du okay zu viel Velocity haben wir dagehakt und nimmst danach einen Durchschnitt. Du hast sicher so dass manche Sprints nicht abschließt. Weil das natürlich viele Leute krank werden. Aber im Schnitt soll Du im Schnitt solltest du es gut vorauszusagen.

77 -00:14:07

78 -I: Vielleicht auch dazu, ist es überhaupt möglich die Velocity so gut voraussagen. Was eben auch schwer macht, wenn man Zwei Wochen Sprint, dass man sagt ich habe 15 Story Points, die gemacht werden sollen, weil die Velocity das sagt. Aber kann man das eigentlich voraussagen, wenn man jetzt noch diesen Aspekt des nicht fixen Teamsetting hernimmt.

79 -00:14:35

80 -I: Also bezogen auf Teilzeit Krankenstände, Urlaube. Also alles zusammen. Also von der Theorie das ist es ja eigentlich so aber in der Realität, wenn man sieht die meisten Teams eigentlich auch so aufgebaut.

81 -00:14:54

82 -B2: Ich glaube was da Gefährliches ist. Du hast meistens jemanden der die Velocity ausrechnet und dann hast du meistens irgendwo einen Projektmanager kommt und sagt: Warum habt ihr das nicht geschafft. Ja die Velocity ist glaube ich nicht ein fixer Wert, den wir jetzt unbedingt immer erreichen muss. Sondern über die Zeit ist er konstant. Ich glaube, dass das auch etwas wo sich das entscheiden. Ich glaube die kann man relativ genau vorhersagen. Kleiner Gedanke dazu. Wir haben jetzt letztens bei der Retro angeschaut. Und die war in den letzten drei oder vier Sprints exakt der Output was geliefert worden ist.

83 -00:16:06

84 -I: Vielleicht gleich zur nächsten Frage, wenn man sich jetzt in einer GO-Live Phase befindet. In einer Phase wo es ziemlich stressig teilweise auch von Kundenseite selbst wird. Was macht das Sinn in solche Phasen, was für einen Ansatz zu verwenden.

85 -00:16:18

86 -B2: Da ist es schwierig. Ich bin eigentlich ein Verfechter, dass man trotzdem bei Timeboxed bleibt. Auch wenn mir persönlich vorkommt, dass es halt sehr schnell immer in dieses Setting hineingeht, dass man quasi man war lange im Timeboxed drinnen. Also

speziell jetzt auch Scrum und Kanban betrachtet. Aber dann Switch du dann meistens hinüber, weil du denkst dir, es ist jetzt sowieso eine Kransh Phase. Wir müssen alle Überstunden machen und schauen, dass das weitergeht. Man versucht Zeit zu sparen. Ich glaube aber, dass das eigentlich ein sehr schlechter Gedanke ist, weil man sich dadurch nicht zeitspart.

87 -00:17:01

88 -B2: Die Frage ist aber was im Hintergrund steht auch. Was ist das für ein Projekt. In meinem Fall ist es ein Dienstleister das ist sowieso bissel was anderes. Immer. Wenn du ein fixes Produkt hast. Glaube ich Wer Timeboxed im Moment das ideale Mittel, das ich nehme. Aber das hängt immer davon ab was das Goal ist was dahinter liegt. Also würde ich sagen standardmäßig bei solchen Fällen Non-Timeboxed und schaut, dass man dieses Projekt fertig macht und diese Phase ist oder so anstrengend.

89 -00:17:45

90 -I: Also eigentlich schon die nächste Frage welches Projekt Setting ist entscheidend dafür. Wenn ich jetzt ein Pilotprojekt habe, wenn ich jetzt ein Wartungsprojekt habe, ein Produkt oder ein Projekt, Wo nehme ich welchen Ansatz?

91 -00:18:13

92 -B2: Also ich würde sagen bei Produkt wäre ich immer bei Timeboxed und Projekt ist es unterschiedlich hängt immer ein bisschen davon ab Tendenz zu Timeboxed, aber es gibt sicher Phasen wo es Non- Timeboxed ist bzw. Projekte wo es einfach mehr Sinn macht. Aber ich denke das muss man sich mal im Detail anschauen was man damit erreichen möchte.

93 -00:18:49

94 -I: Bei Wartungsprojekten Support-Projekten wo jetzt viele auch viele Defekte eingelastet werden. Was macht hier mehr Sinn?

95 -00:19:00

96 -B2: Die Frage ist hat es so viel damit zu tun. Ich glaube das hat immer ein bisschen was damit zu tun wie wir deine Strategie zu der Qualität ist und wie viel Zeit du zur Verfügung hast. Du wenig Zeit zur Verfügung hast ist die Qualität des Projektes fast immer schlecht. Und das hängt auch davon ab was du was du quasi machen möchtest. Und welche Strategie Du hast. Es wäre durchaus möglich, dass man zum Beispiel eine Zero-Bug-Policy. Ich kenne einen PO der sagt er möchte Bugs gar nicht zehn sondern die müssen einfach gefixt werden direkt also gar nicht. Also im Grunde gar nicht getimeboxed.

97 -00:19:54

98 -B2: Das hängt ein bisschen davon ab. Ich würde sagen ich glaube, es macht keinen Unterschied ob du zu Timeboxed oder Non- Timeboxed gehst. Ich glaube es macht einfach einen Unterschied wie du dazu stehst. Also speziell in meinem Fall ist es zum Beispiel so Wenn dem der Kunde egal ist wie viele Bugs er hat und wichtig ist es einfach Story-Points gemacht werden. Dann wird das so passieren.

99 -00:20:25

100 -I: Vielleicht zu Pilotprojekten noch, was würdest du sagen, wenn jetzt wo man jetzt ein Projekt hat wo man nicht wirklich weiß wie man da herangeht oder so probiert wird. Was würde man da am besten nehmen.

101 -00:20:49

102 -B2: Das ist eine sehr gute Frage. Ich denke bei Pilot-Projekten ist es würde ich sagen Es ist sehr wichtig, dass du schnell quasi etwas hin stellst damit muss man auch kann. Ich denke es ist einfach wichtig, dass man das sieht auch wenn es vielleicht am Anfang irgendwie so stellst du mal einfach nur ein Dreirad hin. Und nicht wirklich den Jet. Nur mal schauen ob das Sinn macht. Insofern würde ich quasi sagen. Auch da. Könnte man Richtung Timeboxed gehen.

103 -00:21:30

104 -I: denkt jetzt ein Pilotprojekt was sehr schwierig ist oder auch neue Technologien man kann eigentlich alles unter Pilot-Projekten verstehen. Auch neue Technologien, die man vielleicht noch ausprobieren muss damit man schaut ob sie überhaupt passen oder ob man sie überhaupt anwenden soll.

105 -00:21:53

106 -B2: Okey, ist es etwas, dass eine fixe Timeline hat. Oder ist es so etwas wie probiert.

107 -00:21:59

108 -I: Etwas das man einfach probiert und man schaut sich an ob es jetzt passt oder nicht. Und wenn es mal passt dann wendet man es vielleicht an.

109 -00:22:15

110 -B2: Würde auch dort hingehen, dass man auf jeden Fall ein Modell nimmt, wo man Releases hat, die einfach funktionieren. Ich glaube auch da ist es ein bisschen so. Irgendwann gibts in diesen Projekten dann die Phase wo jemand sagt: schauen wir uns das an was wir da tun. Da musst du drüber kommen, wenn du da auch ein Produkt hast praktisch, aber es fehlt die Hälfte. Und Deswegen funktioniert jetzt nicht richtig dann wird es negativ bewertet werden.

111 -00:22:49

112 -I: Kann ich das wirklich in einem Sprint geben, wenn man nicht weiß was sie da gemacht werden oder man hat ja nicht wirklich eine Idee. Man kann, dass dann auch schwer schätzen bei Pilotprojekten. Da ist eben auch sehr schwierig.

113 -00:23:05

114 -B2: Ja, aber das macht nichts. Das würde ja das würde ja nur schätzen die Komplexität erhöhen. Du hättest ja, wenn du unbekannte drinnen hast. Würdest du einfach nur die Komplexität höher schätzen. Auch in dem Fall also bei dem Projekt hat man ja zumindest je zumindest eine Sache das man sagt Das ist unser Ziel wo wir

hinwollen. Aber welche Technologie wird es verwenden komplett unklar. Vielleicht gibt es verschiedene Ansätze auch unklar, aber du hättest trotzdem jemanden der das Ziel hast. Das ist die Software, die wir machen wollen. Und dann muss sich jemand hier hinsetzen und überlegen okay so könnte man das erreichen und bricht das man runter Ganze. Und die müsstest du dann weiter unterbrechen bis man es schätzen kann. Also möglich wäre es. Ich hätte ein bisschen die Angst, wenn es Non-Timeboxed, dass du zu lange quasi dran sitzt. Das man sich mehr Zeit lässt.

115 -00:24:20

116 -I: Gleich zur letzten Frage, diesen Aspekt der Qualität hat es einen Einfluss darauf ob man jetzt Timeboxed oder Non- Timeboxed nimmt, sinkt die Qualität dadurch oder steigt sie steigt. Was macht mehr Sinn?

117 -00:24:38

118 -B2: Schwierig. Ich würde ein bisschen dazu tendieren. Non-Timeboxed zu nehmen. Für die Qualität war es einfach mehr Zeit hast du und mehr Zeit heißt eigentlich mehr Qualität. Aber hängt immer davon ab wie man mit der Qualität umgeht und dies auch gelebt wird.

119 -00:24:59

120 -I: Wenn man sich jetzt ein typisches Projekt anschaut wo man Sprints hat, wenn wir eben von Timeboxed redet, wäre es schwieriger zu sagen, dass man mehr Qualität als wenn man jetzt einen Non-Timeboxed Ansatz wie Kanban zum Beispiel verwendet. Das meinst du?

121 -

122 -00:25:31

123 -B2: Genau das mein ich. Wobei das natürlich nicht sein muss. Was jetzt einfach Qualität betrifft ist ich fahre das Hardline. Es muss die Qualität muss auf jeden Fall einen sehr hohen Standard haben. Dann geht das sicher auch mit Scrum. Es hängt natürlich auch

davon ab was für ein Produkt du entwickelst. Sagen wir jetzt Software für Flugzeuge wirst du ganz anders rangehen als Software für irgendein Zip Programm.

124 -00:25:57

125 -I: Das wäre es eigentlich schon mit den Fragen. Vielen Dank für das Gespräch. Danke fürs Interview und alles Gute.

126 -00:26:20

127 -B2: Danke.

BEFRAGTER 3

1 -00:00:00

2 -I: Gut, magst du dich kurz vorstellen.

3 -00:00:05

4 -B3: Also mein Name ist Eva. Ich bin derzeit tätig als Scrummaster. Ich bin mittlerweile dreieinhalb Jahre in einer Software Firma tätig. Meine höchst abgeschlossene Ausbildung in MBA in IT-Management und habe Berufserfahrung im Bereich Softwareentwicklung seit 23 Jahren. Allerdings nicht direkt in der Softwareentwicklung. Also jetzt von meiner Seite her sondern als Projektmanager. Und weil es meine ursprüngliche Tätigkeit war und bin gewechselt vor zwei Jahren in dem Bereich des Scrummasters.

5 -00:01:11

6 -I: Im Vorhinein möchte ich mich bedanken. Kurz zur Forschungsfrage die Forschungsfrage lautet: „Welche Auswahlkriterien sind entscheidend für Timeboxed und Non-Timeboxed Ansätze?“. Ich möchte dich über die anonyme Behandlung der Daten informieren. Und möchte damit dein Einverständnis damit einholen.

7 -00:01:25

8 -B3: Ja bin einverstanden.

9 - 00:01:30

10 -I: Gut, Du hast eigentlich die erste Frage schon vorweggenommen was mit der abgeschlossenen Ausbildung und Berufserfahrung. Deswegen können wir mit der zweiten Frage schon anfangen. Die zweite Frage lautet, benutzt du eben in Softwareprojekten agile Methoden und wenn ja welche?

11 -00:01:52

12 -B3: Aktuell benutze ich die hier angeführten Non-Timeboxed Ansätze mit Kanban. Auf der anderen Seite wo in der Vergangenheit war es Scrum Teams.

13 -00:02:10

14 -I: Als du Scrum verwendet hast bei den Teams, wurde Scrum getimeboxed oder eher nicht?

15 -00:02:15

16 -B3: Ja es war immer getimeboxed.

17 -00:02:20

18 -I: Das heißt man hat die Sprints respektiert und nichts raus und reingenommen wie bei Kanban zum Beispiel, wo man so zusagen sich vieles erlauben darf.

19 -00:02:36

20 -B3: Genau, Wie definierst du eigentlich den Begriff Timeboxed? Von meiner Seite aus, man hat einen gewissen abgeschlossenen Zeitrahmen sowohl bei den Meetings als auch im Ablauf.

21 -00:02:55

22 -I: Es geht darum, dass wenn ich die Entwicklungsdauer anschau, dass die getimeboxed ist oder eben nicht getimeboxed ist. Das heißt ich habe eine gewisse Dauer

von zwei Wochen Beispiel bei Scrum wo ich den ganzen Tasks durchführe und umsetze, was bei Kanban z.B. Nicht der Fall ist.

23 -00:03:17

24 -B3: Da muss ich gleich einhaken, weil indirekt ist es bei unserem Kanban System auch der Fall. Wir haben im Grunde genommen auch einen Timeboxed Ansatz im Kanban Bereich, nicht so wie Scrum von zwei Wochen. Aber indirekt schon, weil wir nämlich einen Release Zyklus von einer Woche haben. Wir planen auf einer Woche, getimeboxed. Schauen uns an von der Planung her was gehört hinein. Bei Operations ist es der Fall, dass sie in der Bereitschaft als Team auch Blocker und so lösen müssen, die auf P-Systemen vorkommen. dadurch wird diese Planung zwar unterbrochen, aber das hindert uns diese Planungsiteration einhalten.

25 -00:03:50

26 -I: Cool, gut zu wissen. Bezogen auf die Zeit also Timeboxing was ist deiner Meinung nach die Auswahlkriterien warum ein Projektmanager bzw. Timeboxing bzw. Non-Timeboxing anwenden soll.

27 - 00:04:15

28 -B3: Timeboxing bedeutet für mich eigentlich, dass ein Fokus auf einem gewissen Thema liegt. Das heißt das Meeting zum Beispiel steht unter einem bestimmten Titel und das heißt alle Teilnehmer sind auf diesen Bereich fokussiert oder sollten darauf fokussiert sein. Es hat sich in der Praxis gezeigt, dass die Leute, wenn die Teilnehmereine gewisse Zeit in Anspruch nehmen können, diese Zeit ausreichend für den Fokus nutzen können. Das heißt wenn ich es nicht Timeboxing würde dann würden sie eventuell abschweifen oder andere Themen miteinbeziehen. Ansonsten würden sie mit der Zeit nicht auskommen. Sucht man sich eine Follow-Up Termin, wo man diese Themen wieder getimeboxed fortführt.

29 -00:06:04

30 -I: Das wäre im Fall von Meetings. Aber jetzt rein von Entwicklungsprozess redet, wann würdest du als Projektmanager Timeboxing bevor ziehen und wann nicht?

31 -00:06:16

32 -B3: In diesem Fall würde ich mit einem geplanten Scope über zwei drei Iterationen Timeboxed vorziehen. Hätte ich diesen Planungshorizont nicht dann würde ich wahrscheinlich Non-Timeboxing vorziehen. Das ändert sich in der Scope in der Umsetzung bzw. Planungshorizont häufiger. Oder ich habe von der Projektsituation etwas wo man kurzzeitiger agieren muss, wie zum Beispiel Prototypen Entwicklung oder ähnliches dann würde man eher einen Non-Timeboxed Ansatz bevor ziehen. Das heißt wenn man auf die Gegebenheiten schneller reagieren muss.

33 -00:07:00

34 -I: Inwiefern hat die Rolle des Timeboxing, bzw. der Non-Timeboxing einen Einfluss auf die Psyche der Mitarbeiter haben. Da geht ebenso um Motivation Druck und Stress. Welcher Ansatz wäre besser für die Mitarbeiter.

35 -00:07:29

36 -B3: Das hängt ganz stark vom Projekt ab. Wenn ich jetzt Forschungs- und Entwicklungsbereich bin, man hat die Anforderung in kürzerer Zeit. Ich habe aber den Fokus als Forschungs- und Entwicklungsabteilung das ich forsche und die Ergebnisse immer wieder miteinbeziehen muss dann habe ich natürlich einen anderen Fokus bzw. eine andere Motivation, als wenn ich sage Ich bin ein Projektumsetzungsteam das eigentlich in der Umsetzung arbeitet ein großes Projekt umzusetzen und ständig ändert sich der Scope. Je nachdem wie man es angeht je nachdem wie die Grundanforderungen aus dem Projekt heraus sind oder um was es geht ob das Projekt ist ob das Produkt ist. Hat es einen anderen Ansatz wie es auf die Motivation beziehungsweise die Psyche bzw. auf den Stresslevel der einzelnen Personen wirken kann. Bei Projekten kommen ja solche Faktoren wie ich habe einen Go-Live Termin, ich habe eine Abgabe usw. Kann in der Produktentwicklung natürlich auch sein. Allerdings wenn ich in der Produktentwicklung

bin dann verzögert sich hat was, dann wird es halt später ausgeliefert. Da ist Impact wahrscheinlich nicht so groß wie wenn ich wirklich eine Abgabe mit dem Kunden vereinbart habe. Wechselt sich der Mitarbeiter motivieren zu arbeiten oder wenn man beispielsweise und zusätzlich wechselt sich der Scope.

37 -00:09:29

38 -I: Aber wenn wir uns jetzt auf den Mitarbeiter konzentrieren, hilft ein Timeboxed oder einen Non-Timeboxed Ansatz mehr motivierender zu arbeiten? Das heißt wenn ich abgeschlossene Sprints habe, wo ich meinen Erfolg sehe, hilft mir das mehr oder stresst mich das eher?

39 -00:09:43

40 -B3: Das kommt auf die Umgebung an was eigentlich entwickelt werden soll. Ich glaube, dass das ganz stark davon abhängig ist was ich eigentlich entwickelt ist das Produkt oder ist es Forschung und Entwicklung. Wenn man es bezieht aus dem Bereich ich hab einen Projekt das umgesetzt wird ist wahrscheinlich mit hoher Wahrscheinlichkeit sogar für alle Personen die in diesen Teams sitzen, Komfortabler wenn es über Iterationen geht, weil Scrum auch vorgibt dass die Planung steht dass was in der nächsten Iteration ist steht, dass sich die Leute darauf einstellen können was sie erwartet. Diese Transparenz ist für das Team natürlich von Vorteil. Und wirkt sich natürlich auf den Stresslevel positiv aus.

41 -00:11:00

42 -I: Das heißt nicht, dass wenn ich in einem Timeboxed Ansatz bin ich eher gestresst bin, weil ich die Aufgaben fertig machen muss? Was nicht bei einem Non-Timeboxed Ansatz der Fall ist.

43 -00:11:13

44 -B3: Das Team bringt sich in die Planung des Sprints ganz klar ein. Wenn ich weiß wie meine Velocity wie meine Kapazitäten ausschauen. Wie ich mit der technischen

Umsetzung umgehe, dann weiß ich worauf kann ich mich einlassen bzw. kann das auch entsprechend planen. Das Team ist da gefragt aktiv mitzugestalten und kann sozusagen diesen nächsten zwei Wochen richtig planen. Sie wissen was sie erwartet. Bei Non-Timeboxed hast du immer wieder das Thema die Anforderungen ändern sich, sind alle technischen Gegebenheiten und alle Abhängigkeiten abgeklärt. Ist es richtig refined? Fehlen mir einige Informationen? Wir als Team auch dem Fokus darauf, dass es sagt was hat das für einen technischen Impact, wenn das jetzt hineinkommt. Bei größeren Projekten hast du technische Abhängigkeiten. Lassen sich vielleicht auf den ersten Blick nicht so zeigen, und wenn es sich laufen ändert wie bei Non-Timeboxing dann werden manche Dinge übersehen.

45 -00:12:30

46 -I: Zur nächsten Frage gleich inwiefern ist die Struktur des Teams ein Einflussfaktor, um sich für Timeboxed oder Non-Timeboxed zu entscheiden? Also wie das Team aufgebaut ist, wie das Team-setting ist? Das heißt habe ich fixe Mitarbeiter? Aushilfskräfte? Vollzeitbeschäftigte, Teilzeitbeschäftigte usw.

47 - 00:13:00

48 -B31: Je mehr Einflüsse von außen wie Aushilfsmitarbeiter oder Nebenjobs oder Externe die in anderen Zeitzonen sitzen oder ähnliches. Man sollte die Transparenz im Team hochhalten. Man soll die Möglichkeiten geben allen Mitgliedern im Team ausreichende Informationen zukommen zu lassen. Aus dieser Perspektive her wirklich daran arbeiten zu können als gemeinsames Team ist das sicherlich von Vorteil, wenn man das getimeboxed macht.

49 -00:13:58

50 -I: Das heißt wenn ich jetzt ein Team habe, das viele Mitarbeiter hat mit dem schwer rechnen kann. Das heißt du weißt nicht der Mitarbeiter der Teil angestellt ist arbeitet vielleicht zwei Tage, das hat auch viel mit der Velocity zu tun. Wenn er jetzt man krank wird dann fehlt er die ganze Woche. Und dann kannst du eben schwer Timeboxed mit

solchen Mitarbeitern oder Mitarbeiter die auch andere Aufgaben haben wie Lead Aufgaben usw.

51 -00:14:28

52 -B3: Die Transparenz im Team ist aber eine viel Höhere also theoretisch gesehen sollte das Team an höheren Informationsfluss haben durch die Meetings durch das vorgegebene Framework. Sie stimmen sich besser ab sie haben einen besseren Informationsfluss, der vorgegeben ist. In Kanban zum Beispiel muss man das aus den Personen herausholen. Kanban oder Non-Timeboxed setzt einen höheren Reifegrad des Teams voraus als es Scrum tut.

53 -00:15:06

54 -I: Danke, nächste Frage gleich. Inwiefern hat die Team-Velocity und die Sprintdauer einen Einfluss drauf, ob ich mich für Timeboxed oder Non-Timeboxed entscheide? Also wenn ich 4 Wochen Sprints habe, kann ich das überhaupt so leicht planen? In 4 Wochen kann vieles passieren, ein Mitarbeiter muss vielleicht ins Ausland fahren, da andere muss Pflege-Urlaub nehmen usw. und das beeinflusst natürlich die Velocity.

55 -00:16:32

56 -B3: Wir haben in unseren Teams normalerweise eine Dauer von zwei Wochen und hatten auch über Weihnachten und ähnliches eine Dauer von vier Wochen und die Praxis hat gezeigt, dass es fast nicht planbar war. Je kürzer desto besser und übersichtlich. Es ist leichter für die Mitarbeiter im Team das Ganze zu fassen und selbst die Übersicht über eine Iteration zu behalten. Als über vier Wochen. Aber das ist jetzt halt sehr spezifisch auf die Teams, mit denen ich gearbeitet habe.

57 -00:17:28

58 -I: Ist es eigentlich überhaupt möglich Velocity vorauszusagen, dass man damit planen. Wenn man dran denkt das man nicht nur fixe Mitarbeiter hat.

59 -00:17:41

60 -B3: Je stabiler das Team desto eher die Velocity desto eher kann ich sie feststellen. Wenn man jetzt wirklich Aushilfskräfte, Teilzeitkräfte denn sie sollen ja fixer Bestandteil des Teams sein. Das heißt Leute die ständig wechseln die machen das ganze schwieriger. Die Idee dahinter ist, dass man Entwicklungsteam hat was gut miteinander zusammenarbeiten kann und selbst Leute eben die nur 25 oder 30 Stunden drin sind oder auch externe Kräfte die auch fixe Stunden Kontingent im Team sitzen. Das man schon nach einer gewissen Zeit eine Velocity feststellen kann.

61 -00:18:34

62 -I: Es gibt schon Teams, die das Entwicklungsteams sehr stark steht und nicht wirklich hin und her schwängt, aus meiner Erfahrung kann ich auch dass die Teams nicht wirklich bei mir meinem Projekt jede Woche sich geändert hat, weil einer ausgestiegen ist, der andere muss 2 Wochen auf Urlaub nehmen weitere werden krank usw.

63 -00:19:16

64 -B3: Diese Schwankung hast du sowieso. Urlaube und Krankenstände sind drinnen. Aber das ist ja etwas was du nicht jede Woche berechnen lässt, sondern du hast eine Tendenz. Du sammelst im Grunde genommen Daten und arbeitest mit diesen Daten. Aber dafür muss das Team zumindest stabilisiert sein. Das heißt Es soll keine personellen Fluktuationen geben über Urlaube oder über Krankenstände da kann man über eine gewisse Zeit es messen. Es ist trotzdem nur ein Wert, den man ausrechnen kann und es ist nicht in Stein gemeißelt.

65 -00:19:58

66 -B3: Zusammengefasst bei längeren Entwicklungsiterationen macht es mehr macht es mehr Sinn das man Kanban, also Non-timeboxed anwendet, weil man schwer damit rechnen kann. Aber man muss sich ganz genau anschauen, welchen Reifegrad hat mein Team. Auf der anderen Seite, was für ein Projekt, Produkt, Forschung, Entwicklung habe ich und was möchte ich damit erreichen. Ich habe mehrere Faktoren, die ich mit einbeziehen muss, um festzustellen was für eine Art Werkzeug ich dafür wähle.

67 -00:21:23

68 -I: Vielleicht gleich zu nächsten Frage, aus sehr spannend, kann es meine Entscheidung beeinflussen, wenn ich jetzt in einer Go-Live Phase bin? Das heißt ich muss jetzt die ganzen Themen fertigstellen und ich befinde mich ein bzw. Zwei Monate vor dem Go-Live.

69 -00:21:36

70 -B3: Es gibt einige Teams die in solchen Situationen vor dem Go-Live wirklich dann auf Kanban umsteigen. Weil sie sagen, sie sind damit effizient sie können schneller den Task oder Bugfixes oder ähnliches abarbeiten. Da gibt es einige Punkte zu berücksichtigen. Man erspart sich zwar die Meetings auf der einen Seite, aber man muss auf der anderen Seite gewährleisten, dass der Informationsfluss bestehen bleibt. Ansonsten Es kommt öfter vor das Teams darauf umsteigen. Wir sind jetzt in einer Abschlussphase. Wir müssen nur die restlichen Features fertig stellen. Bei uns ist alles klar und jetzt geht's wirklich nur ans Coden.

71 -00:22:50

72 -I: Vielleicht jetzt zum Thema Projekt Setting, ob es ein Produkt, ein Projekt, ein Pilotprojekt usw. ist, bei welcher Projektart würdest du welchen Ansatz bevorzugen.

73 -00:23:13

74 -B3: Bei einem Projekt abhängig von den einzelnen Parametern von den einzelnen Faktoren wie Reifegrad des Teams. Wie sie jetzt vom technischen Knowhow usw. Wenn ich es bewerten müsste würde ich sagen Pilotprojekt würde ich wahrscheinlich eher kurz ist mit Non-Timeboxing entwickeln. Support und Wartungs-Projekte sind eher auch eher nach Non-Timeboxing und ein Projekt bzw. Produkt eher Timeboxing. Abhängig von anderen Parametern. Und würde abhängen z.B. Pilotprojekte gibt es ja auch viel Längere. Das ist ja nicht unbedingt, dass man einen Piloten von einem Monat, sondern das kann ja auch eineinhalb Jahre dauern würde man anschauen vom Umfang her, wie man es einteilt und plant. Aber grundsätzlich würde ich so die Einteilung vornehmen.

75 -00:24:41

76 -I: Gleich zur Letzen Frage, haben die 2 Ansätze einen Einfluss auf die Qualität. Was macht mehr Sinn, um die Qualität zu erhöhen?

77 -00:24:46

78 -B3: Ja ich würde sagen es hat einen Unterschied, weil, es gibt einen Unterschied und der ist man muss im Fokus haben das Timeboxing Ansätze eine Hilfestellung für ein Team bieten kann. Das heißt ich habe ein Framework, dass dem Team die Möglichkeit gibt Informationen auszutauschen in strukturierten Meetings. Das heißt ich kann ein Team nehmen, dass noch nicht einen extrem hohen Reifegrad hat und kann innerhalb dieser Struktur Software entwickeln lassen. Während Non-Timeboxed voraussetzt, dass die Mitglieder des Teams untereinander Möglichkeiten schaffen müssen wie sie z.B. Know-How Transfer betreiben wie sie sich abstimmen usw.

79 -00:26:17

80 -B3: Aber da jetzt eine Aussage zu treffen das wäre zu vereinfacht, ohne diese ganzen Parameter mit einzubeziehen oder den Fokus auf einen gewissen Part zu haben, weil man es nicht so pauschal sagen kann. man hat Faktoren, die man miteinbeziehen muss, die zu dieser Entscheidung führen und die auch wiederum Rückschlüsse auf den Output schließen lassen könnten.

81 -00:26:52

82 -I: Vielleicht ein Beispiel dazu, wenn ich jetzt einen Timeboxed Ansatz nehmen würde und ich komme wie bei Scrum dazu, dass ich schnell schnell testen muss damit ich die Sachen abschließe, weil der Zeitdruck einfach gegeben ist. Ist es bei Non-Timeboxed nicht der Fall, weil ich mir da mehr Zeit lassen kann.

83 -00:27:14

84 -B3: Na, das Release muss trotzdem fertig werden. Man hat halt nicht das Ende des Sprints als Marker. Sondern man hat andere Marker, wo man fertig werden muss. Ich hab

einen gewissen Delivery Prozess, mein Release muss fertig werden, gewisse Abgaben, gewisse Abhängigkeiten zu anderen Abteilungen oder ähnliches das heißt ich habe dann andere Parameter.

85 -00:27:43

86 -B3: das Timeboxed ist nur etwas verschoben in der Ebene. Ich habe es nicht auf Teamlevel das Timeboxing ist jetzt im Sprint sondern auf einer anderen Ebene, dass ich sage ich habe eine Abgabe. Man hat immer den Faktor Zeit drinnen. Mir wäre es nicht bewusst in der Produktentwicklung das ich sage ich kann es jetzt hinauszögern oder Wir haben die Zeit so. Irgendwelche Stakeholder kommen und geben eine Abgabe. Spätestens auf einen anderen Level, holt einen das sowieso wieder ein.

87 -00:28:30

88 -I: Also würdest du nicht eine Entscheidung treffen was besser für die Qualität wäre?

89 - 00:28:30

90 -B3: Nein, würde ich nicht, nicht generell, weil Zeit und Geld immer irgendwo eine Rolle spielen. Man kann fast nicht sagen, dass man sagt wir können uns nicht Zeit lassen bis wir etwas abgeben, weil es ist Non-Timeboxed und Dadurch ist die Qualität höher. Diese Schlussfolgerung würde ich jetzt nicht ziehen.

91 -00:29:10

92 -B3: Gut dann bedanke ich mich für deine Zeit und Antworten und wünsche dir alles Gute.

BEFRAGTER 4

1 -00:00:00

2 -I: Magst du dich kurz vorstellen?

3 -00:00:10

4 -B4: Mein Name ist Mo. Ich bin Gründer und Geschäftsführer der Firma Epicore. Ich bin 26 Jahre alt und ich habe einen Bachelor of Science in Internettechnik und mache meinen Master in Wirtschaftsinformatik.

5 -00:00:28

6 -I: Alles klar. Dank ihrer Teilnahme am Interview, möchte die Forschungsfrage kurz vorstellen. Die Frage lautet Welche Auswahlkriterien sind entscheidend für Timeboxed und Non- Timeboxed Ansätze? Als nächstes möchte ich dich über die anonyme Behandlung der Daten informieren und möchte damit auch sein Einverständnis einholen.

7 -00:00:59

8 -B4: Hiermit bestätige ich

9 -00:01:01

10 -I: gut, fangen wir gleich an, die erste Frage des Interviews lautet eben, du hast vorgegriffen. Was ist eine höchst abgeschlossene Ausbildung und welche Berufserfahrung bringst du mit im Softwarebereich.

11 -00:01:18

12 -B4: In den Anfangsjahren der Firma war ich sehr stark involviert in der Softwareentwicklung. Nach und nach wurden meine Fähigkeiten im digitalen Marketing ausgebaut und im Management und die letzten zwei bis drei Jahre habe ich mich eigentlich so gut wie von der Entwicklung zurückgezogen, zumindest nicht in der direkten Entwicklung, sondern im Management im Testing und im Projektmanagement.

13 -00:02:01

14 -I: Dankeschön, in den Software Projekten, die du gehabt hast im Laufe der Firma habt ihr agile Entwicklungsmodell verwendet.

15 -00:02:07

16 -B4: Ja haben wir. Wir haben das Scrum quasi adaptiert und auf unsere Bedürfnisse zugeschnitten.

17 -00:02:22

18 -I: Dazu, habt ihr auch Scrum getimeboxed oder habt ihr es nicht getan.

19 -00:02:32

20 -B4: Also, wenn man das jetzt streng betrachtet haben wir es. Wir haben eine Lösung zwischen Timeboxing und nicht Timeboxing, das heißt es gibt Deadlines. Aber diese Deadlines werden je nach Bedürfnis abgeändert bzw. da wir sehr schnell sehr stark in Produktentwicklung fokussiert sind und es meistens Geschäftsmodelle bzw. Produktideen die vielleicht noch nicht gegeben hat kann man nicht genau definieren, wenn man auf sehr viel forscht und daher gibt es Deadlines aber die werden meistens abgeändert.

21 -00:03:22

22 -I: Das heißt Die Scrum Sprints waren jetzt nicht wirklich Timegeboxed sondern manchmal verlängert manchmal gekürzt oder wie hat das ausgeschaut?

23 -00:03:37

24 -B4: Also die Sprints waren meistens eins bis 2 Wochen. Diese Pakete, die dort vereinbart wurden, die werden es dann meistens. Das heißt wenn man auf die Sprints quasi runter setzt. Dann ja dann haben wir da meistens schon etwas abgeändert werden aber im Großen und Ganzen über das ganze Projekt ist es meistens so dass Termine abgeändert werden.

25 -00:04:05

26 -I: Danke und dann gleich zur nächsten Frage bezogen auf die Zeit also Timeboxing. Was sind deiner Meinung nach die Auswahlkriterien warum ein Projektmanager wie du zum Beispiel Timeboxing bei einem Projekt nimmt oder nicht Timeboxed.

27 -00:04:30

28 -B4: Es kommt immer ganz auf die Kriterien an. Es kommt immer auf das Projekt drauf an. Es gibt Projekte, die erfordern ein gewisses Launch oder Go-Live und das ist dann mit anderen Maßnahmen verbunden mit Marketing mit Geschäftseröffnung et cetera mit gewissen Förderungen die gewisse Phasen erreichen müssen. Und da ist es auf jeden Fall sinnvoll, dass man sich strikt an die Zeiten hält und da ist ein Timeboxing auf jeden Fall auch sinnvoll.

29 -00:04:47

30 -I: Wann wäre Non-Timeboxing sinnvoll anzuwenden?

31 -00:04:54

32 -B4: In der klassischen Prototyping bzw. Produktentwicklung Modus. wenn es darum geht eine gewisse Idee zu bauen und da ist es meistens so dass man gewisse Sachen testet dann wieder entwickelt dann verfeinert und da ist sehr viel Forschung da. Meiner Meinung nach ist das nicht sinnvoll da Time zu boxen.

33 -00:05:14

34 -I: gleich zur nächsten Frage Inwiefern kann die Rolle des Timeboxing oder des Non-Timeboxing Einfluss auf die Psyche der Mitarbeiter haben, einfach die psychischen Effekte?

35 -00:05:34

36 -B4: Ich glaube es kommt immer darauf an wie realistisch dieses Timebox gesetzt werden. Wenn man einen, wenn man einen Spielraum quasi definiert ein gewisses Polster integriert in das Timeboxing dann sollte es keine psychischen Belastungen geben. Wenn man aber ein Timeboxing anwendet wo man gewisse Requirements gibt und diese abgeschätzt werden und man keinen Polster hat dann kann es schon zu einen gewissen Stress Faktor kommen und dadurch auch einen größeren Fehler verursachen.

37 -00:06:14

38 -I: Inwiefern ist es motivierend beim Timeboxing oder Non- Timeboxing?

39 - 00:06:25

40 -B4: Ich glaub es ist motivierend gewisse Meilensteinen zu haben und diesen Meilensteinen so weit zu setzen, dass man nicht zu sehr in die Tiefe geht. Wenn es um mein Projekt geht. wenn es darum geht gewisse Requirements zu machen dann gibt es sehr viele Requirements die man einfach nicht sieht nicht an fassbar ist und sehrviel ist es meistens so dass es dann nach einer Zeit sehr demotivierend ist, wenn man etwas nicht an fassbares hat. Deswegen finde ich, dass es situationsbedingt man das anwenden sollte das Timeboxing.

41 -00:07:14

42 -I: Also meinst du das Timeboxing eher motivierend sein könnte, wenn du diese Slots hast, die du abschließt und dann auch motiviert wirst?

43 -00:07:30

44 -B4: Ganz genau!

45 -00:07:35

46 -I: Gut dann zur nächsten Frage was das Team betrifft. Inwiefern ist die Struktur des Teams oder Inwiefern hat die Struktur des Teams einen Einfluss auf diese Entscheidung?

47 -00:07:47

48 -B4: ich glaube es hat etwas mit einer gewissen Größe zu tun. Wenn es wirklich nur um die Entwicklung geht wie viele Entwickler Es gibt je größer das Team desto mehr Struktur braucht es desto mehr macht es Sinn Timeboxing anzuwenden weil gewisse Pakete von einem Teammitglied umgesetzt werden dann werden sie von einem anderen übernommen und dort weiter gemacht. Das heißt wenn ein Mitglied auf dem anderen

warten muss, weil er einen gewissen Freiraum hat dann ist das auf jeden Fall Timeboxing das Passende.

49 -00:08:18

50 -I: Spielt es auch eine Rolle ob du jetzt fixe Mitarbeiter hast oder Nebenjob Vollzeit Angestellte. Spielt das eine Rolle auf die Entscheidung ob sie die Berufserfahrung vielleicht auch ob Senioren ob Junior Entwickler sind hat das alles damit zu tun ob ich jetzt Timeboxing nehme oder nicht?

51 -00:08:44

52 -B4: Ich finde es hat schon etwas mit der Erfahrung zu tun aber das haben wir so gelöst, dass wir sagen auf jeden Junior ist ein Senior quasi als Begleiter. Als jemand den man um Rat fragen kann damit diese Deadline eingehalten werden können.

53 -00:09:13

54 -I: Wenn du jetzt Mitarbeiter hast in deinem Team die manche sind bei dir Teilzeit angestellt manche sind Vollzeit beschäftigt bei dir spielt das genauso eine Rolle, dass du jetzt sagst derjenige der nicht Vollzeit angestellt ist, ist 2-mal oder 3-mal die Woche da und wenn er jetzt mal ausfällt krank ist dann ist er die ganze Woche nicht da.

55 -00:09:44

56 -B4: Ja das ist ein ganz guter Punkt mit Vollzeit Mitarbeitern ist die Flexibilität viel höher da, wenn Ausfälle entstehen. Es sollte einfach ein Zusammenspiel sein Aus Vollzeit und freien Mitarbeitern und Teilzeit Mitarbeitern und es sollte bedacht werden, dass auch wenn man Timeboxing anwendet bei einem Teilzeit Mitarbeiter und der jetzt krank ist wird das nichts ändern. Deswegen sollte man eigentlich ein Backup haben, dass wenn ein Mitarbeiter ausfallen würde, dass ein anderer Mitarbeiter quasi übernehmen kann. Aber das wird für mich nicht die Frage. Es wäre für mich kein Faktor Timeboxing anzuwenden oder nicht.

57 -00:11:46

58 -I: Zur nächsten Frage hat vielleicht auch damit zu tun, inwiefern kann die Team Velocity, die Geschwindigkeit des Teams sozusagen beim Entwickeln und Sprint Dauer also der Umfang ein Einflussfaktor sein, um sich für Timeboxed oder Non- Timeboxed zu entscheiden

59 -00:12:52

60 -B4: Also es hat auch damit zu tun mit der anfangs Frage ob wir Timeboxing anwenden und in welcher Form. Da würde ich glaube ich gleich anschließen und sagen das beste Modell eines Sprints, sind zwei Wochen und in diesen zwei Wochen sollten sollte man eigentlich die Timeboxing soweit anwenden, dass man zwei Wochen Zeit haben diese Pakete abzuarbeiten. Und es gibt ein Zwischenspiel zwischen Meeting in dem Sprint nach einer Woche. Und da macht es auf jeden Fall weniger Sinn. Es macht keinen Sinn in diesen zwei Wochen kein Timeboxing anzuwenden, weil ich muss mit dem Projekt ja komplett fertig werden. Das heißt innerhalb von sechs Monate macht es keinen Sinn meiner Meinung nach Timeboxing zu machen. Das heißt In den zwei Wochen die Dinge, die man vereinbart haben sollten, diese Pakete, die wie wir behandelt haben sollten in der Zeit Aber es kann zu viel passieren innerhalb des Projektes, das es auf einen längeren Zeitraum nicht sinnvoll wäre.

61 - 00:14:24

62 -I: dann gleich zu Team Velocity also die Team-Geschwindigkeit. Die erste Frage, die sich stellen würde ist, kann man die Velocity überhaupt voraussagen. Ist das überhaupt möglich?

63 -00:14:33

64 -B4: Die Geschwindigkeit eines Mitarbeiters ist eigentlich ein Faktor von zwei Dingen, und zwar der erste Faktor ist das Know-How. In welchem Bereich der gut ist und in welchem Bereich er sich noch einarbeiten muss. Der zweite Punkt ist die Motivation. Das heißt wenn diese zwei Dinge sind, wenn die Motivation da ist und das nur in einem Bereich da ist dann macht es auf jeden Fall Sinn, dass ein Timeboxing anzuwenden, wenn

man auch die Geschwindigkeit voraussagen kann. Voraussagen kann man nicht. Wenn das Know-How erst erarbeitet werden muss dann kann man das eigentlich so gut wie nicht voraussagen.

65 -00:15:20

66 -I: Aber die Team Velocity hängt ja auch von anderen Sachen ab. Man schaue sich die letzten Sprints an und dann rechnet man anhand dem wie viel das Team geschafft hat und dann nimmt man sozusagen die Story Points für die nächste Iteration. Aber es ist dementsprechend auch schwierig vorzusagen, wenn du im Projekt immer wieder Krankenstände haben, Urlaube, die genommen werden, Leute, die ausfallen, Leute, die dazukommen, Team Umstrukturierung und so weiter und so fort. Macht es dann in solchen Situationen Sinn, wenn man die Team-Velocity nicht voraussagen kann, dass, man Timeboxed nimmt oder nicht.

67 -00:16:05

68 -B4: Je größer das Team desto mehr kommen solche kommen solche Probleme, die man nicht voraussehen kann. Ich glaube aber, dass man abgesehen, man sollte nicht die Sachen, die passieren die man nicht voraussehen kann mit dem Verknüpfen ob man Timeboxed oder nicht. Es werden immer Leute krank es wird immer irgendwelche Probleme geben es wird eine Umstrukturierung geben und deswegen sollte man eigentlich darauf gewappnet sein, dass das genau passiert und dass jemand einspringen kann. Ich finde nicht, dass man das anhand des entscheiden sollte ob man oder nicht Timeboxed sollte man meiner Meinung nach wenn es das Projekt verlangt und dann muss man das Projekt das Team nach dem Projekt ausrichten.

69 -00:17:04

70 -I: Zur nächsten Frage gleich hängt die Entscheidung auch davon ab ob man jetzt kurz vor einer Go-Live Phase ist? Das heißt du hast einen Monat bist du jetzt mit deinem Produkt oder mit deinem Projekt Go-Live gehst. In dieser Phase in einem Monat ist es natürlich sehr stressig. Du musst vielleicht noch Features fertig machen du musst noch

Bugs fixen, du musst noch einige Sachen erledigen damit du Go-Live gehen kannst. Macht es da Sinn vielleicht Non- Timeboxed zu verwenden oder würdest du das sagen will ich trotzdem Timeboxed.

71 -00:17:45

72 -B4: Ich würde sagen, dass es hier auf jeden Fall Sinn macht, wenn man nun mal in ein Monat muss man muss man live gehen. Dann sollte das auf jeden Fall Timegeboxed werden, weil es einfach eine ganz andere Emotion dem Team überbracht wird. Wir haben jetzt einen Monat den drucken wir durch und dann ist es vorbei. Und da ist auf jeden Fall Timeboxing eine Hilfe.

73 -00:18:17

74 -I: Schon aber da haben wir auch den Nachteil, dass du sagst Du kannst nicht mehr geben, bei vielen bei vielen Projekten ist es so. Wenn das kurz vor Go-Live ist, haben sie einfach so viel zum Abarbeiten das sich das in einem Monat nicht ausgeht. Und sehr viele Teams gehen dann zu einem Einsatz Kanban, dass sie sagen wir machen Non-Timeboxing damit wir die ganzen Pakete so gut wie schnell schaffen und weniger Meetings haben und so weiter. Würdest du diesen Ansatz auch verfolgen oder wieso sagen Nein ich würde weiterhin Timeboxing.

75 -00:18:48

76 -B4: Wir haben ja ganz am Anfang diesen Begriff Timeboxed verwendet. Es geht jetzt darum um eine Meinung das ist die Meinung, die ich habe, aber wie man das Firmenintern quasi regelt oder wie man das quasi auf die Firma adaptiert. Dann bin ich eigentlich der Meinung dass wir einen Go-Live haben wo wir sehr viele Faktoren haben, wo sich gewisse Dinge ändern, dass man einen Plan haben sollte, eine Deadline haben sollte dass das quasi unterzubringen und so flexibel damit umgehen kann das man okey und das kann man jetzt bei Go-Live machen. Und das kann man nach Go-Live machen. Schauen wir einfach das Produkt rausbringen und diese Features brauchen wir jetzt nicht. Was heißt so flexibel und das auch mit dem Kunden kommuniziert, dass man das

Bestmögliche aus dieser Zeit machen kann, weil wenn es nicht geht dann kann man es dann auch nicht dafür sorgen dass wir es fertig machen.

77 -00:19:49

78 -I: Nächste Frage dann gleich welches Projekt Setting ist entscheidend für diese Frage ob ich Timeboxing oder Non- Timeboxing verwende? Ob ich jetzt ein Pilotprojekt habe ob ich jetzt ein Produkt habe ein Firmen internes Projekt ein Wartungs Projekt und so weiter und so fort. Bei welchen Arten von Projekten würdest du was nehmen.

79 -00:20:06

80 -B4: Timeboxing würde ich eigentlich immer nur anwenden, wenn wir eine Zeit haben, die wir auf jeden Fall nicht überschreiten dürfen. Und das gilt bei allen möglichen Faktoren. Das heißt wenn ein Datum da ist und dieses Datum ist absolut nicht flexibel dürfte es nicht dann würde ich da immer Timeboxen und sonst nicht.

81 -00:20:30

82 -I: Ja aber zum Beispiel, du hast am Anfang auch erwähnt Forschungsprojekte. Da hast du nicht wirklich diese Deadlines und da kann man sich auch da braucht noch mehr Zeit, dass man, dass man die Sachen anschaut und dass man schaut wie die Sachen funktionieren deshalb hast du gesagt, dass du da Non- Timeboxed verwenden würdest.

83 -00:20:48

84 -B4: Da will ich auch dabeibleiben, aber das fehlt nicht unter dem Punkt, dass man Deadlines hat.

85 -00:21:01

86 -I: gut dann gleich zur letzten Frage. Die Qualität wenn du dir diesen Aspekt Qualität anschaust in den Projekten würdest du dann bevorzugen dass du jetzt Timeboxed verwendest oder eher Non- Timeboxed, wo du dann mehr Zeit hast sozusagen nicht diesen Stress hast vor dem springende das du die ganze Sache noch fertig testen musst

und das so zum Beispiel wie bei Kanban mehr Zeit hast dann dich für das Testing oder für die Qualität dann mehr einzusetzen?

87 -00:21:40

88 -B4: Qualität ist immer so eine Sache. Man hat immer diesen Punkt die Qualität und die Quantität. Da muss man einfach ein Zwischending haben. Es kommt auch immer auf das Budget des Kunden an. Das heißt die Geschichte mit Testing ist immer so eine Budget Sache, wenn man ein gut kalkuliertes Budget hat und der Kunde auch bereit ist diesen Faktor zu bedienen, weil für Kunden ist das einfach mittlerweile selbstverständlich ist es aber nicht. Ein ausgiebiges Testing ist Zeit und kostenintensiv und ich bin kein Fan von Stress. Ich bin auch kein Fan von okey Wir müssen das jetzt so schnell wie möglich raushauen. Sondern ich sehe das eher wie ein Marathon. Es ist nicht gut der Erste zu sein wie bei einem Sprint, sondern der der schlussendlich den Maraton fertig ist dann der der schlussendlich der Sieger ist.

89 -00:22:43

90 -B4: Und diesen Ansatz würde ich dann also auf jeden Fall in der Softwareentwicklung anwenden. weil die Qualität immer darunter leidet, wenn ein gewisser Stressfaktor da ist und vor allem das mit dem Timeboxing sehr gegeben ist. Also ich würde sogar behaupten, dass im Timeboxing die Qualität darunter leidet als wenn flexible Deadlines gesetzt sind.

91 -00:23:15

92 -I: Vielen Dank für deine Antworten und für das Interview. Alles Gute.

93 -00:23:19

94 -B4: Dankeschön.

BEFRAGTER 5

1 -00:00:00

2 -I: Servus Christian, schön dass du da bist, magst du dich bitte kurz vorstellen.

4 -00:00:16

5 -B5: Ja guten Morgen mache ich gerne mein Name ist Christian ich bin seit knapp 10 Jahre in der Softwareentwicklung als Softwareentwickler tätig. Ich habe an vielen verschiedenen Großprojekten gearbeitet auch in verschiedenen großen Firmen und sehr viel Erfahrung in der Software Entwicklung gesammelt. Ich bin 35 Jahre alt und habe mein Master of Science Abschluss in Softwareentwicklung und Wirtschaft an der TU in Graz absolviert.

7 -00:01:21

8 -I: Super. Erstmals möchte ich mich bei dir herzlich für die Teilnahme am Interview bedanken und möchte dir kurz die Forschungsfrage vorstellen. Die Forschungsfrage lautet: welche Auswahlkriterien sind entscheidend für timeboxing und non timeboxing Ansätze?

10 -00:01:59

11 -B5: ja ich bin damit einverstanden.

13 -00:02:03

14 -I: Super dann fangen wir gleich mit den Fragen an oder hast du noch im Vorfeld fragen.

16 -00:02:20

17 -B5: Nein ich habe keine offenen Fragen, fangen wir gleich doch mit den Fragen an.

19 -00:02:35

20 -I: gut dann starten wir gleich mit der ersten Frage, Bei die erste Frage haben sie haben sie schon vorweggenommen welche Berufserfahrung sie haben in der Softwareentwicklung und welche höchste abgeschlossene Ausbildung sie haben. Ich würde dann gleich mit der nächsten Frage weitermachen. Die Frage lautet benutzt du in deinen Softwareprojekten agile Entwicklungsmethoden und wenn ja welche sind?

22 -00:03:16

23 -B5: ja ich benutze oder wie benutze in der Firma verschiedene Softwareentwicklungsmethoden agile Softwareentwicklungsmethoden wie Scrum Kanban Punkt in meinem derzeitigen Projekt arbeiten wir in einer Teamgröße von 15 Personen und drei kleinen Sub-Teams mit Scrum. Vielleicht kurz auch auf die Fragestellung ja wir benutzen Scrum in unserem Projekt der teilweise nicht Timeboxed ist Punkt dieses aus dem Grund, weil sehr oft einige defekt die unbedingt in diesem Sprint gelöst werden müssen in den Sprint genommen werden, ohne das timeboxing zu respektieren.

25 -00:03:49

26 -I: vielen Dank lieber Christian. Gleich zur nächsten Frage, bezogen auf die Zeit was ist deiner Meinung nach die Auswahlkriterien warum ein Projektmanager timeboxing bzw. non timeboxing Ansätze anwenden soll?

28 -00:04:12

29 -B5: ja das ist eine sehr gute Frage da fließen natürlich viel mehr Aspekte hinein als ich die, dass ich jetzt alle auflisten könnte aber so einige wichtige Entscheidungskriterien werden zum Beispiel Projektart um was für ein Projekt es sich handelt. Genauso aber Team Konstellationen wie das Team aufgebaut ist und ob die Mitarbeiter nur aus Vollzeitkräften oder auch Aushilfskräften bestehen. Und die wesentliche Frage ist auch wie stabil ist das Team, in dem ich arbeite, kann ich mit diesem Team überhaupt planen oder nicht.

31 -00:04:52

32 -I: vielen Dank für deine Antwort du hast schon einige Punkte erwähnt, die ziemlich wichtig für den Verlauf dieses Interviews sein werden, wir kommen später wieder drauf zu. Gleich zur nächsten Frage, inwiefern kann die Rolle der timeboxing bzw. der non timeboxing einen Einfluss auf die Psyche des Mitarbeiters haben also psychologische Effekte wie Motivation Druck Stress und so weiter?

34 -00:05:24

35 -B5: ja ich glaube, dass der psychologische Effekt sehr wohl bei diesen Ansätzen eine große Rolle spielt. Bei timeboxing hast du z.B. diesen Druck im Team wenn das Team nicht gescheit seine Arbeit macht, nicht gescheit die Tasks schätzt, die Schätzung auch nicht ernst nimmt dann hast du auf der einen Seite diesen Stressfaktor, dass am Ende des Sprints deine Tasks nicht fertig werden. Auf der anderen Seite hast du den Stress und den Druck sowieso, weil man in einem Zeit engeren Rahmen sich befindet und die Tasks fertig stellen soll, egal welche Umstände im Sprint sich befinden.

36 -Bei Non-Timeboxed hast du diesen Stressfaktor weniger, natürlich kann man irgendwann mal Releases und Deadlines, aber diese sind nicht alle 2 Wochen, was dem Mitarbeiter eine gewisse Freiheit gibt und ihn nicht unter Druck setzt.

37 -Auf der anderen Seite hast du natürlich bei Timeboxing den Vorteil, dass eine gewisse Motivation gegeben ist, wenn du siehst wie du eines Sprints immer wieder zeitgerecht und erfolgreich abschließt. Dieser Motivationsfaktor ist bei Non-Timeboxed Ansätze nicht gegeben, da man keine kleinen Abschnitte oder Iterationen hat, in denen man arbeitet.

38 -00:07:39

39 -I: Vielen Dank für deine Aussagen über diesen Punkt. Gleich zu nächsten Frage: Inwiefern ist die Struktur des Teams ein Einflussfaktor um sich für Timeboxed bzw. Non-timeboxed Ansätze zu entscheiden? Also wie das Team aufgebaut ist, ob's nur fixe Mitarbeiter oder Aushilfskräfte im Team sind, ob alle Vollzeit angestellt sind oder nicht,

vielleicht spielt auch die Berufserfahrung eine Rolle? Aber genauso auch weitere Firmeninterne Aufgaben wie Teamlead, Chapterlead usw.

41 -00:08:18

42 -B5: Diesen Aspekt habe ich am Anfang eh kurz angerissen. Ja ich finde und das ist mal meine Meinung, ich finde das spielt eine große Rolle wie das Team aufgebaut ist, wie das Teamsetting ist. Wenn man eine optimales Teamsetting hat, wo alle Mitarbeiter, fix angestellt waren, fast alle Vollzeit angestellt ist, wo alle Mitglieder des Teams sich auf die Team Aufgaben fokussieren und nicht weitere Firmenaufgaben erledigen müssen, dann ist das ein optimales Team mit dem man planen kann, und Teams mit denen man gut planen kann, sind super geeignet dafür Timeboxing Ansätze zu verwenden.

43 -Auf der anderen Seite weiß man das dieses optimales Teamsetting in weniger Projekten gegeben ist. Sehr oft hat man, was ich auch in meinen Projekten habe, Mitarbeiter, die Firmenweite Aufgaben haben, wo man sich schwer tun mit ihnen zu planen, weil sie zu diesem Meeting müssen, und da hinreisen müssen, und bei einem anderen Projekt aushelfen müssen und weil sie Leader Aufgaben haben. Natürlich kann man sagen, durch den Informationsfluss im Team muss das Team wissen was es für Ressourcen genau hat und was es im Stande ist zu tun, aber dann kommen Sachen, wie der Kunde des Projekts möchte nicht abrüsten, sondern eher mehr an Story Points bekommen, und wo das Team dann nicht mehr weiß was er tun soll.

44 -Aushilfskräfte sind genauso ein Punkt wo ich eher dazu tendiere, wenn ich viele Aushilfskräfte im Projekt habe, dass ich Non-Timeboxed Ansätze verwende, weil ich schwer mit diesen Mitarbeitern rechnen kann. Aber auch Teilzeit Kräfte, wenn es viele im Team sind, dann eher Non-Timeboxing. Ich glaube aber nicht, dass die Berufserfahrung einen Faktor für diese Entscheidung sein kann, da sowas immer in beiden Ansätzen gegeben ist und im Endeffekt das Teamintern gemanagt werden muss.

46 -00:13:09

47 -I: Vielen Dank für die ausführliche Antwort, gleich zu nächsten Frage: Inwiefern kann die Team-Velocity und die Sprintdauer (Springumfang) ein Einflussfaktor sein um sich für Timeboxing oder Non-timeboxing zu entscheiden?

49 -00:13:50

50 -B5: Ja das mit der Velocity ist immer ein Thema. Grundsätzlich sollte die Velocity getrackt werden, um in Zukunft anhand der zu erfahren wie viel das Team schafft und damit bei Timeboxed Ansätze plant. Die Frage ist, aber wie ist es, wenn ich kontinuierlich ein unstabiles Team habe, weil jemand kündigt, dann kommt jemand dazu, dann wechselt einer in ein anderes Team, dann kommen andere externe Mitarbeiter ins Team, dann will der Kunde das Team reduzieren usw. Diese Fälle sind real und derzeit in meinem Team sehr gegeben. Deshalb ist bei uns die Velocity kaum und sehr schwer vorauszusagen. Was natürlich das ganze schwieriger macht einen Timeboxing Ansatz zu verwenden, weil man nicht weiß wie viele Story Points das Team schaffen kann und somit schwer damit planen kann.

51 -Wenn's um die Sprintdauer geht, dann wird's spannend. Eine Iteration von 2 Wochen ist gerade noch planbar aber erst wenn ich die Velocity kenne, Alles über 2 Wochen ist sehr unüberschaubar, und natürlich auch schwer planbar. Bei Solchen Fällen dann eher Non-Timeboxing verwenden.

53 -00:16:41

54 -I: Vielen Dank für deine Antworten. Die nächste Frage lautet: Inwiefern kann die Entscheidung davon abhängen, ob man in der Go Live Phase ist und man die Themen fertigstellen muss?

56 -00:17:00

57 -B5: In unseren Projekten gehen die meisten Teams, wenn es anfängt stressiger zu wird vor einem Go-Live Termin, wo die Features noch fertig gestellt werden und wo sehr viele Bugs seitens der Kunden auch eingelastet werden, gehen die meistens Teams und sagen wir machen jetzt Non-Timeboxing. Der Grund dahinter ist, dass man erstens sehr

viele Bugs reinbekommt und es schwer wird jetzt zu sagen ja dieser wichtige Bug wartet jetzt mal bis zum nächsten Sprint, denn ich kann nichts mehr im Sprint aufnehmen. Und zweitens, weil man effektiver und schneller entwickeln kann, wenn man am Tag nicht dauern von gewissen Scrum Meetings unterbrochen wird und somit auch weiterkommt bei der Implementierung.

59 -00:19:11

60 -I: alles klar, wird der Prozess auch seitens der Kunde so gewünscht, dass man Richtung Non-Timeboxed geht?

62 -00:19:26

63 -B5: in den meisten Fällen interessiert es dem Kunden in solche Situationen sind nicht in welchem Prozess wir arbeiten. Der Kunde will bei einer Go-Live Phase seine Features und sein fertiges Produkt haben. Nicht mehr und nicht weniger.

66 -00:20:12

67 -I: vielen Dank für deine Antwort, die nächste Frage gleich. Welches Projektsetting ist entscheidend für Timeboxing bzw. Non-timeboxing? Du hast am Anfang auch gesagt, je nach Projektart und Projektsetting. Obs jetzt ein Produkt, ein Projekt, ein Pilotprojekt, ein Forschungsprojekt, ein Wartungs- & Supportprojekt ist.

69 -00:20:49

70 -B5: Ja ich glaube die Projektart ist ein ganz wichtiger Entscheidungskriterium. Zum Pilotprojekt zum Beispiel, hatten wir den Fall in der Firma, dass ein Kunde eine Technologie ausprobieren wollte in Kombination mit dem Produkt den wir entwickeln. Die Technologie war recht neu von SAP released und der Kunde wollte es eben das wir es ausprobieren und austesten. Wir haben im Endeffekt versucht es in unserem Sprint mit aufzunehmen, aber mit der Zeit haben wir gemerkt es geht irgendwie nicht da wir es weder schätzen könne noch in einem Sprint oder mehreren damit fertig werden. In unserer

Retrospektive haben wir dann entschieden, es in einem Non-Timeboxed Ansatz anzugehen, da es einfach mehr Sinn macht bei solchen Projekten.

71 -Bei Projekt und Produktentwicklung wäre meiner Meinung nach, wenn du ein Team hast, mit dem du planen kannst, Timeboxing zu verwenden, weil da hast du die kurzen Abschnitte, da hast du die Releases und kannst so Schritt für Schritt es weiterentwickeln.

72 -Bei Wartungs- und Supportprojekten macht es durchaus Sinn Non-Timeboxing zu verwenden. In solchen Projekten geht's halt drum Probleme auf die Schnelle zu lösen, dem Kunden zu supporten usw. Und bei solchen Projekten kann man schwer Timeboxing, weil die Sachen auf einem plötzlich zukommen und man es lösen muss.

74 -00:23:14

75 -I: Und wie schauts bei Forschungsprojekten aus, welchen Ansatz würdest du da verfolgen?

77 -00:23:25

78 -B5: Ich würde Forschungsprojekte wie Pilotprojekte behandeln, da man einfach die gleichen Rahmenbedingungen hat, man probiert etwas aus, schaut sich's an und testet es, um es dann zu verwenden. Deshalb würde ich da genauso Non-Timeboxing verwenden.

80 -00:23:55

81 -I: Vielen Dank für die Ausführung, gleich zur letzten Frage: Haben die 2 Ansätze „Timeboxing“ und „Non-timeboxing“ einen Einfluss auf die Qualität der Software?

84 -00:24:15

85 -B5: Oh, dass ist aber eine schwierige Frage! Ich würde sagen, ja es hat einen Einfluss, denn mehr Zeit heißt im Grunde mehr Qualität, somit würde ich schon sagen das beim Timeboxing die Qualität drunter leiden könnte, weil man will ja seinen Sprint schnell schnell fertig bekommen, alles am letzten Tag noch irgendwie testen, damit man den Sprint schafft. Und dieses schnell schnell testen, führt teilweise dazu, dass die Qualität

drunter leidet und sinkt. Anders als bei einem Non-Timeboxing Ansatz, wo dieser Zeitdruck in kleineren Iterationen nicht gegeben ist. Natürlich gibt es eine Deadline, die eingehalten werden muss, aber diese Deadline ist im Normalfall nicht alle 2 Wochen.

87 -00:25:32

88 -I: vielen Dank für das Interview und deine Antworten Christan.

90 -00:25:40

91 -B5: Danke auch!

BEFRAGTER 6

1 -00:00:00

2 -I: Hallo Anna, magst du dich kurz vorstellen?

3 -00:00:09

4 -B6: Mein Name ist Anna. Ich bin 29 Jahre alt. Ich habe das Studium IT-Recht und Management an der FH-Joanneum abgeschlossen und mein Master Titel dadurch erlangt. Und arbeite schon seit 3 Jahre als Scrum Master in Software Unternehmen in Softwareentwicklungsteams. Davor habe ich ca. 5 Jahre als Entwicklerin gearbeitet.

5 -00:00:39

6 -I: Alles klar. Dank deiner Teilnahme am Interview, ich möchte dir die Forschungsfrage kurz vorstellen. Die Frage lautet Welche Auswahlkriterien sind entscheidend für Timeboxed und Non- Timeboxed Ansätze? Gibt es jetzt schon eine Frage dazu? Oder ist es verständlich?

7 -00:01:12

8 -B6: nein alles soweit klar.

9 -00:01:16

10 -Als nächstes möchte ich dich über die anonyme Behandlung der Daten informieren und möchte damit auch dein Einverständnis einholen.

11 -00:01:30

12 -B6: Bin einverstanden.

13 -00:01:35

14 -I: gut, fangen wir gleich mit der zweiten Frage an, weil die erste Frage hast du eigentlich schon beantwortet, und das wäre deine kurze Vorstellung. Die zweite Frage lautet: du hast ja in verschiedenen Firmen in verschiedenen Softwareentwicklungsteams gearbeitet. Welche agile Software Entwicklungsprozesse habt ihr da angewandt, und waren diese Methoden timeboxed oder eher nicht?

15 -00:02:18

16 -B6: Ja wir haben agile Software Entwicklungsprozesse in allen Teams wo ich gearbeitet angewandt. Alle Teams wo ich gearbeitet habe, sehr wohl als Entwicklerin als auch als Scrum Master haben die Methode „Scrum“ angewandt.

17 -Scrum wurde bei all den Projekten, wo ich involviert war, timeboxed. Es gab aber immer wieder so kleine Ausnahmefälle, wo man nicht timeboxed gehabt hat, aber das waren wirklich Seltenheiten. Ein Beispiel wäre z.B. kurz vor einer GoLive Phase.

18 -00:03:22

19 -I: Finde ich sehr interessant, zu dem Punkt Go-Live kommen wir später noch einmal. Vielen Dank erstmal für die Beantwortung dieser Frage. Vielleicht gleich zur nächsten Frage, Was sind deiner Meinung nach die Auswahlkriterien warum ein Projektmanager Timeboxing bei einem Projekt nimmt oder nicht Timeboxed als Entwicklungsansatz nimmt?

20 -00:04:08

21 -B6: Ja ich glaube das ich ist die Kernfrage des heutigen Abends, es ist schwer pauschal zu sagen es hängt von einem einzigen Kriterium ab. Viel mehr sind meistens mehrere Unterschiedliche Kriterien drin involviert. Solche Kriterien können zum Beispiel Sachen sein wie Teamgröße, Teamkonstellationen auch ob mehrere Sub-Teams an einem Produkt oder Projekt zusammenarbeiten, aber genauso was der Kunde vorgibt oder gernhätte.

22 -Möchte der Kunde für sich kontinuierlich einen Überblick schaffen und regelmäßig Abgeschlossenen Iterationen haben?, dann kommt eher Timeboxing in Frage. Ist es dem Kunden wurscht Hauptsache er hat sein Endprodukt zu der Deadline könnte auch Non-Timeboxing in Frage kommen. Genauso hat die Frage damit zu tun, ob man mit dem Team planen kann, mit dem ich arbeite. Gibt mir die Firma überhaupt diese Rahmenbedingungen, dass ich ein Team habe mit dem ich geplant arbeiten kann. Wie gesagt es spielen da mehrere Faktoren mit. Auch Sachen wie was für ein Projektsetting das ist. Ist es eher ein Forschungsprojekt oder ein gängiges Projekt.

23 -00:05:37

24 -I: Ganz genau du hast es e schon richtig erkannt, dass ich eigentlich einer der Hauptfragen dieses Interviews. Du hast schon sehr viele wichtige Punkte erwähnt, auf die wir dann gleich einzeln zurückkommen werden. Aber Vielleicht gleich die nächste Frage als Zwischen Frage: Inwiefern kann die Rolle des Timeboxing oder des Non- Timeboxing Einfluss auf die Psyche der Mitarbeiter haben, einfach psychischen Effekte wie Motivation, Stress und Druck.

25 -00:06:11

26 -B6: Natürlich kann man sagen Timeboxing ist eher stressiger und druckvoller als Non-Timeboxing, weil man in einer gewissen Zeit mit gewissen Aufgaben fertig sein muss. Es ist auch um einiges stressiger, aber die Frage ist wie geht das gesamte Team damit um, setzen wir uns ein Polster bzw. einen Buffer, um nicht in diesem Druck gedrängt zu werden oder nicht.

27 -Und natürlich ist Timeboxing motivierender als Non-Timeboxing, weil du die gewissen Slots oder Iterationen hast, wo man Sachen erledigt und wo man motivierend in die nächste Iteration geht, weil man die letzte Iteration erfolgreich abgeschlossen hat. Es kann aber demotiviert sein, wenn man immer wieder die Iterationen nicht schafft, und keine Maßnahmen ergreift, um dies zu verbessern, wie z.B. Velocity erneut validieren oder Buffer einbauen usw.

28 -00:07:44

29 -I: Gut danke, du hast bei der ersten Frage kurz angesprochen Sachen wie Teamgröße, Teamkonstellationen. Die nächste Frage lautet eben: Inwiefern ist die Struktur des Teams oder Inwiefern hat die Struktur des Teams einen Einfluss auf diese Entscheidung? Also wie das Team aufgebaut ist.

30 -00:08:15

31 -B6: Ja eben ein Kriterium wäre eben die Frage des Teams. In welche Teamgröße arbeite ich. Habe ich ein Team, das ziemlich groß ist, sprich 5 plus, dann braucht das Team mehr Struktur, und mehr Struktur bedeutet Timeboxing. Teams, wo der Reifegrad sehr hoch ist und wo die Kommunikation ohne Meetings und ohne sämtliche Treffs funktioniert, könnte auch mit Non-Timeboxing arbeiten.

32 -Genauso ist die Frage arbeite ich mit mehreren Sub-Teams ob nur ein Team. Wir haben z.B. 2 Sub-Teams, eines ist ein reines Frontend Team und eines ist ein Backend Team. Bei Sub-Teams oder bei Abteilungen, wo die Aufgaben von einander abhängen ist Timeboxing perfekt geeignet. Denn dadurch kann man planen, was man macht und was nicht, und muss nicht ewig auf eine Sache von dem anderen Team warten, weil sie eben Non-Time boxen.

33 -00:09:49

34 -I: Spielt es auch eine Rolle ob ich jetzt fixe Mitarbeiter habe oder Nebenjob Vollzeit Angestellte oder Aushilfskräfte. Spielt das eine Rolle auf die Entscheidung ob sie die

Berufserfahrung vielleicht auch ob Senioren ob Junior Entwickler sind hat das alles damit zu tun ob ich jetzt Timeboxing nehme oder nicht?

35 -00:10:20

36 -B6: Als erstes möchte ich mal die Erfahrung ausschließen. Ich finde nicht das die Frage etwas mit Erfahrung zu tun hat, weil im Endeffekt hat man immer Teams mit unterschiedlichen Niveaus, mit verschiedenen Levels an Seniorität und Erfahrung. Was aber die Anstellungsart angeht, so spielt es auf jeden Fall eine Rolle. Wenn man viel zu viele Aushilfskräfte im Team hat, die nicht immer da sind oder externe Mitarbeiter hat, mit ihnen kann man ganz einfach nicht planen, und da wird es dann schwer Timeboxing anzuwenden.

37 -Optimal ist natürlich Vollzeit Mitarbeiterinnen, aber oft hat man auch den Fall das einige Teilzeit Angestellte im Team sind. Teilzeit Kräfte sind halb so schlimm wie externe Ressourcen, mit ihnen kann man zumindest planen wann sie arbeiten an welchen Tagen und wann nicht.

38 -00:11:43

39 -I: Vielen Dank für die Antwort. Zur nächsten Frage gleich, hat indirekt auch etwas mit dem Team zu tun, inwiefern kann die Team Velocity und die Sprint Dauer also der Umfang ein Einflussfaktor sein, um sich für Timeboxed oder Non- Timeboxed zu entscheiden.

40 -00:12:03

41 -B6: Ja die Velocity ist ein breites Thema, es kann mit der Entscheidung zu tun haben, wenn man ein inkonstantes Team hat. Ein Team, das eben nicht aus fixen Mitarbeitern besteht, sondern nur immer externe Mitarbeiter dazu holt, wenn sie gebraucht werden.

42 -In so einem Team kann man die Velocity sehr schwer messen bzw. tracken, weil man jede Iteration andere Personen am Board hat. In solchen Situationen macht natürlich Non-

Timeboxing mehr Sinn. Bei Teams wo die Velocity vorauszusagen ist, weil das Team eben immer dasselbe ist, ist es sehr sinnvoll Timeboxed Ansätze zu verwenden.

43 -Ich bin sowieso ein Verfechter Fan von Timeboxing, und bin der Meinung, Timeboxing immer verwenden, wenn man mit dem Team planen kann. Kann man mit dem Team aus verschiedenen Gründen nicht planen so geht man eben zu Non-Timeboxing.

44 - 00:13:30

45 -I: Und inwiefern hat die Dauer der Iteration damit zu tun?

46 -00:13:41

47 -B6: Bei Timeboxing Ansätze wie Scrum, ist die übliche Iterationsdauer 2 Wochen. Bei 2 Wochen macht es gerade noch Sinn Timeboxing zu verwenden, wird diese Zeit mehr, 3 oder 4 oder mehr Wochen so sollte man eher dazu gehen und Non-Timeboxing verwenden. Bei uns im Team in der Weihnachtszeit also e bald in ein paar Sprints, machen wir einen 4 bzw. 5 Wochen Sprint, weil man eben die ganzen Feiertage hat.

48 -Da wir die Timeboxing Methode Scrum verwenden, gehen wir natürlich nicht nur in diesem Monat zu einem Non-Timeboxing Ansatz. Obwohl es eigentlich im Grunde Sinn macht. Wir merken jedes Jahr, wenn wir dabei sind, diese längere Iteration zu planen, dass die Planung einfach nicht funktioniert. Es macht kein Sinn für 5 Wochen voranzuplanen, denn in diese 4 bis 5 Wochen kann vieles passieren, mit dem man nicht rechnen kann.

49 -Somit würde ich sagen, wenn Iterationen von länger als 2 Wochen gesetzt sind dann würde ich eher Non-Timeboxing verwenden, weil die Planung damit einfach viel zu schwierig wird.

50 -00:15:42

51 -I: Danke für die Ausführung. Zur nächsten Frage gleich hängt die Entscheidung auch davon ab ob man jetzt kurz vor einer Go-Live Phase ist? Das heißt du hast einen Monat

bist du jetzt mit deinem Produkt oder mit deinem Projekt Go-Live gehst. In dieser Phase in einem Monat ist es natürlich sehr stressig. Du musst vielleicht noch Features fertig machen du musst noch Bugs fixen, du musst noch einige Sachen erledigen damit du Go-Live gehen kannst. Macht es da Sinn vielleicht Non- Timeboxed zu verwenden.

52 -00:16:45

53 -B6: Ich habe am Anfang e schon erwähnt gehabt, dass ich es öfters erlebt habe, sowohl als Entwicklerin als auch als Scrum Master, dass Teams in so eine stressige Phase wie, die Go-Live Phase, dazu tendieren eher Non-Timeboxing Ansätze zu verwenden.

54 -Ich glaube es ist schon ein recht berechtigter Gedanke, weil man will zum Go-Live alles fertig stellen, die ganzen aufgetretenen Bugs oder Defekts fixen, man hat vielleicht noch einige Features, die man fertig implementieren muss und noch gut testen muss, und die Zeit ist eben knapp.

55 -Der Kunde möchte nicht mehr verschieben, weil es mit anderen Marketing Maßnahmen usw. verbunden ist vielleicht. Und in diesem ganzen Stress und Druck haben die Teams verschiedene Meetings ausfallen lassen, haben alles in einem Board gepackt und als Ziel gesetzt, dies alles muss bis zu der gesetzten Deadline fertig werden.

56 -Man muss halt nur drauf aufpassen, dass du den Ausfall von sämtlichen Meetings, der Informationsfluss nicht verloren geht im Team, und dass das Team updated ist. Also zusammengefasst, macht es durchaussinn einen Non-Timeboxing Ansatz bei so einer Phase anzuwenden, aber mit dem Gedanken das der Informationsfluss trotzdem gesichert ist.

57 -00:18:39

58 -I: Vielen Dank. Nächste Frage dann gleich: welches Projekt Setting ist entscheidend für diese Frage ob ich Timeboxing oder Non- Timeboxing verwende? Da fallen eben Projekte wie Forschungsprojekte, Pilotprojekte, ob ein Produkt oder ein externes oder Internes Projekt, ob ein Wartungsprojekt usw.

59 -00:19:10

60 -B6: Ja ich glaube ich habe es am Anfang es auch kurz angeschnitten. Die Projektart und das Projektsetting spielen bei dieser Frage eine große Rolle. Timeboxing würde ich primär bei gängigen Projekten verwenden. Auch bei Produktentwicklung und Projekt egal ob intern oder extern würde ich Timeboxing verwenden, wenn ich damit planen kann und die Rahmenbedingungen so gesetzt sind für die Vorausplanung.

61 -Bei Wartungsprojekte und Supportprojekte, wo es drum geht, dem Kunden zu supporten und einige Wartungsaufgaben zu übernehmen, würde ich eher Non-Timeboxing Ansätze verwenden, weil man bei solchen Projekten ganz einfach wenig bis kaum vorausplanen kann. Es kommen einfach Aufgaben, die kurzfristig gemacht werden müssen, und da kann man eben schwer vorausplanen, was in den nächsten 2 Iterationen gemacht wird.

62 -Genauso würde ich bei Forschungs- und Pilotprojekte Non-Timeboxing verwenden, weil man bei solchen Projekten einfach mal neue Technologien ausprobiert, Sachen austestet, um diese vielleicht dann im Produkt oder Projekt zu adaptieren. Bei Forschung kann man eben schwer vorausplanen, ich würde sagen man setzt bei solchen Projekten ganz einfach Meilensteine, damit auch die Forschung nicht allzu lang dauert und man in absehbarer Zeit zu einem Ergebnis kommt.

63 -00:21:18

64 -I: gut dann gleich zur letzten Frage: Haben die 2 Ansätze „Timeboxing“ und „Non-timeboxing“ einen Einfluss auf die Qualität der Software?

65 -00:21:40

66 -B6: Natürlich spielt die Zeit da auch mit, weil durch mehr Zeit hat man im Grunde mehr Qualität. Weil man sich eben dann Zeit nehmen kann, um die Features ausgiebig zu testen.

67 -Was hier auch noch für die Qualität dazu kommt ist das Refactoring. Mit der Zeit braucht man immer wieder Zeit dafür um den Code zu refactoren, weil einfach so viel dazu gekommen ist, dass der Code nicht mehr lesbar ist. Natürlich kann man sagen ja wir planen das ein in einer Timeboxed Iteration, aber die Erfahrung zeigt, dass der Kunde dann immer weitere Features will und keinen Wert und keine Priorität für das Refactoring sieht.

68 -Im Gegensatz kann man leichter bei Non-Timeboxed Ansätze das Refactoring machen und auf diesen Aspekt der Qualität achten.

69 -00:22:50

70 -I: Vielen Dank für deine Antworten und für das Interview. Ich wünsche dir noch alles gute!

71 -00:23:01

72 -B6: Dankeschön.

KODIERUNG

KATEGORIE-BILDUNG INTERVIEW 1

Textaussage	Paraphrase	Generalisierung	Reduktion
für Timeboxing spricht, dass wir dem Kunden gegenüber einem zeitlichen Überblick schaffen und dass auch garantieren so gut wie möglich, dass somit auch dem Kunden die Sicherheit gibt auch die Kosten-technischen Fragen für Kunde schon am Anfang des Projektes eine Klarheit schafft	Timeboxing: Zeitlichen Überblick dem Kunden schaffen. Klarheit über kosten-technische Fragen und das garantierten der Fertigstellung so gut es geht	Zeitlicher Überblick durch Timeboxing kosten-technische Fragen beantworten durch Timeboxing Fertigstellung garantieren durch Timeboxing	Kategorie 1: Überblick über das Projekt
Seite der Auftragnehmer in dem Fall, das Unternehmen die Software entwickelt auch mit ihren eigenen Ressourcen rechnen. In dieser Phase habe ich so viele Leute eingestellt und somit ist die Ressource Planung des Unternehmens auch leicht handzuhaben.	Timeboxing: Auftragnehmer kann eine bessere Planung der Ressourcen durchführen.	Bessere Ressourcenplanung durch Timeboxing	Kategorie 1
Was das Non-Timeboxing angeht gibt es wiederum auf der anderen Seite den Nachteil für Kunden, dass er vielleicht hier keinen Überblick hat.	Non-Timeboxing: der Nachteil ist kein Überblick.	Kein Überblick durch Non-Timeboxing	Kategorie 1
Da müsste man dann noch einmal anschauen welche Methode Sinn macht. Es kommt auf das Projekt Art an. Es kommt auf die Kundenwünsche an, auf	Entscheidung hängt ab von mehreren Faktoren: Kundenwünsche,		

die zeitliche Ressourcen, auf die personelle Ressourcen, die hier angewendet werden sollen.	Projektart, zeitliche und personelle Ressourcen.		
---	--	--	--

Tabelle 1: Kategorienbildung, Interview B1 Frage 3

Textaussage	Paraphrase	Generalisierung	Reduktion
wenn man von Timeboxing spricht meistens ist es so dass man immer einen engen Zeitraum definiert. Das ist für die Mitarbeiter oder für das gesamte Team das Projektteam, Ist es ein zeitlicher Druck das die Kreativität der Mitarbeiter einschränken könnte.	Timeboxing: engerer Zeitraum, dass einen zeitlichen Druck für das Team darstellt und somit die Kreativität einschränkt.	Zeitlicher Druck durch Timeboxing Kreativität einschränken durch Timeboxing	Kategorie 2: Psychische Effekte Unterkategorie 2.1: zeitlicher Druck Kategorie 3: Innovation Unterkategorie 3.1 Kreativität
Es ist ein zeitlicher Druck, dass die Motivation der Mitarbeiter sinken könnte	Timeboxing: ein zeitlicher Druck, dass die Motivation senken könnte.	Motivation sink durch zeitlichen Druck des Timeboxing	Unterkategorie 2.2: Motivation
Es ist ein zeitlicher Druck, dass auch die Freiheit des Mitarbeiters auf die Umsetzung des Features einschränken wird. Der Mitarbeiter wird vielleicht nie auf die Idee kommen neues bzw. neue	Timeboxing: ein zeitlicher Druck, dass die Freiheit des Mitarbeiters auf die Umsetzung des Features einschränkt. Freiheit bei	Einschränkung der Freiheit des Mitarbeiters auf die Umsetzung des Features durch Timeboxing	Unterkategorie 3.1

<p>Technologien anzuwenden, weil er diese Zeit vielleicht nicht hat. Dementsprechend hat man diese Freiheit nicht und das ist schon ein großer Faktor.</p>	<p>der Umsetzung neue Ideen/Technologien, ist durch Zeitdruck nicht gegeben.</p>		
<p>Und vor allem in den kritischen Projekten, die dann auch eskaliert werden an das Managementteam ist generell auch der Wunsch in die Arbeit zu gehen manchmal sogar nicht da. Das Mitarbeiter sagen das ist es für mich schon zu viel und ich würde ehrlich gesagt am besten jetzt aufhören.</p>	<p>Timeboxing: bei kritischen Projekten, wo es an das Managementteam eskaliert wird, haben die Mitarbeiter keine Lust mehr und überlegen aufzuhören.</p>	<p>Bei kritischen Projekten, keine Lust auf Arbeit durch Timeboxing</p>	<p>Unterkategorie 2.2</p>
<p>Man hat einen Go-Live Termin definiert und es müsste bis zu den Go-Live wirklich alles fix und fertig sein. Aber man müsste ja dann schauen, Ich kann ich. Ich habe da Beispiele wo auch die Mitarbeiterinnen und Mitarbeiter bis in der Nacht gearbeitet haben. Aber dann in dem Team schon zu Kündigungen kam, freiwillige Kündigungen, weil sie einfach nicht mehr mit diesem Druck leben konnten.</p>	<p>Timeboxing: Kurz vor einer Go-Live Phase, wo es sehr stressig wird, wo alles fertig gestellt werden muss, war der Druck so hoch, dass das Team über die Nacht gearbeitet haben. Dadurch haben einige Mitarbeiter freiwillig gekündigt.</p>	<p>Go-Live Phase kommt es durch den hohen Stress zu Kündigungen durch Timeboxing</p>	<p>Unterkategorie 2.1</p>

<p>Auf der Non-Timeboxing hat man natürlich diese Freiheit und kann man sich ein wenig selbst Spielraum schaffen das schon für die Selbstentwicklung des Mitarbeiters eine wichtige Rolle spielt.</p>	<p>Non-Timeboxing: hat man den Freiraum und nicht diesen Druck, und das ist für die Selbstentwicklung des Mitarbeiters wichtig</p>	<p>Mehr Freiraum durch Non-Timeboxing ist gut für die Selbstentwicklung des Mitarbeiters</p>	<p>Unterkategorie 3.2 Weiterentwicklung</p>
<p>Selbstverständlich wenn diese extremer Druck nicht da ist, ist es natürlich ein Motivationsgrund, hat man gewisse stolz als Team aber auch als Einzelperson. Dass wir in dem vereinbarten Zeitraum das geschafft haben und dazu gibt es immer wieder Zuckerln auch von der Kundenseite dafür, dass man sich dafür einhält.</p>	<p>Timeboxing: wenn dieser Druck nicht da ist, ist Timeboxing sehr motivierend, dass man die Arbeitspakete im vereinbarten Zeitraum geschafft hat.</p>	<p>Motivierend wenn kein es kein Zeitdruck gibt im Timeboxing Motivierend das abschließen von Iterationen im Timeboxing</p>	<p>Unterkategorie 2.2</p>

Tabelle 2: Kategorienbildung, Interview B1 Frage 4

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>hat man Leute die bereits Senioren sind also Senior EntwicklerInnen Senior sind. Da hat man einen gewissen Flow schon entdeckt auch in der Firma schon. Und man geht diese Phase nach und deswegen sind auch in dem Fall auch Timeboxing Methoden für sie auch einfacher handzuhaben.</p>	<p>Timeboxing: bei Teams mit Senioren Entwickler hat man einen gewissen Flow entdeckt, was dann auch einfacher ist Timeboxing anzuwenden.</p>	<p>Senior Mitarbeiter besser für Timeboxing</p>	<p>Kategorie 4: Teamsetting Unterkategorie 4.1 Erfahrung</p>
<p>Des Weiteren ist es wichtig wie viel kann der Mitarbeiter geben. Ist er wirklich Vollzeit für das Projekt da. Manchmal hat man drei Projekte zu betreuen gleichzeitig und überall zu 50 Prozent zu 65 Prozent. Das spielt auch für den Mitarbeiter einen Faktor, dass er sich immer umsteigen muss von einem Projekt zum anderen er die anderen Features da hat er die anderen Regelungen da hat er die anderen Probleme. Das spielt auch einen Faktor, wenn der Mitarbeiter Vollzeit also wirklich ganz voll dabei ist bei dem Projekt ist es wichtig auch ein verlässlicher Mitarbeiter, wenn er Teilzeit da ist gewisse Tage ist er da gewisse Tage ist er nicht da ist er wiederum problematisch mit ihm in Timeboxing zu arbeiten.</p>	<p>Timeboxing: bei Mitarbeiter, die Vollzeit angestellt sind, verlässlich sind, und nur an einem Projekt involviert sind, und nicht gleichzeitig mehrere Projekte betreuen.</p> <p>Non-Timeboxing: bei Mitarbeiter, die an verschiedenen Projekten gleichzeitig arbeiten, die dauern von einem Projekt auf den anderen umsteigen müssen. Auch bei Mitarbeiter die Teilzeit beschäftigt ist.</p>	<p>Verlässliche Vollzeit Mitarbeiter besser für Timeboxing</p> <p>Nicht gleichzeitig mehrere Projekte betreuen, ist besser für Timeboxing Ansatz</p> <p>Bei Teilzeit Mitarbeitern Non-Timeboxing verwenden.</p> <p>Bei Beteiligung an mehreren Projekten</p>	<p>Unterkategorie 4.2 Anstellungsart</p> <p>Unterkategorie 4.3 100% bei einem Projekt</p> <p>Unterkategorie 4.2</p>

		Non-Timeboxing verwenden.	Unterkategorie 4.3
Wenn wie gesagt die Berufserfahrung spielt hier eine Rolle wie viele Jahre hat man schon Projekt entwickelt. Seit wie viel Jahren sind man überhaupt in der Firma. Es kann sein, dass er schon seit zehn Jahren Software Projekte entwickelt. Aber es ist halt neu in der Firma und ist neu konfrontiert mit dieser Art und Weise wie die Firma arbeitet.	Timeboxing: Firmenerfahrung spielt ebenso eine Rolle, weil man mit der Art und Weise wie die Firma arbeitet, konfrontiert ist.	Timeboxing verwenden bei Mitarbeitern mit Firmenerfahrung	Unterkategorie 4.1
Man hat viele schon Released aber dann braucht man externe Ressourcen und nimmt sie mit auf bis diese Leute dann on-boarded werden. Ist diese On-boarding Phase ist für das Projekt nimmt es auch sehr viel Zeit Anspruch und das könnte auch für Timeboxed Ansätze problematisch werden.	Non-Timeboxing: externe Mitarbeiter. die in einem Projekt einsteigen, der schon länger gibt, brauchen Zeit bis sie sich ins Projekt reinfinden und in Timeboxing arbeiten können.	Non-Timeboxing verwenden bei Quereinsteiger und externe Kräfte	Unterkategorie 4.2
Und auch wenn man die Mitarbeiter anschaut, man müsste auch die Ressourcen anschauen was die Mitarbeiter hergeben kann, wenn der Mitarbeiter in der Firma bereits auch andere Rollen hat, wie zum Beispiel Team Lead, Entwicklungslinie, Development LEAD, und so weiter und so fort. Dann kann er auch nicht alles	Non-Timeboxing: bei Mitarbeiter, die mehrere Rollen (z.B. Development Lead) in der Firma haben und gleichzeitig auch in einem Projekt involviert sind, sodass man schwer mit ihnen	Non-Timeboxing bei mehreren Rollen in der Firma gleichzeitig	Unterkategorie 4.3

hergeben oder wenn der Projektmanager fünf Projekte leitet und sich nicht auf einem Projekt konzentrieren kann könnte es hier in Timeboxed zu Problemen kommen.	rechnen oder planen kann.		
dass bei manchen Strukturen des Teams also wenn nicht optimal strukturiert ist, das dann Non-Timeboxed besser funktionieren würde als Timeboxed weil hier eben diese zeitliche Freiheit schon noch gerade gibt. Es ist nichts sehr locker, aber es ist auch nicht so streng wie es in Timeboxed ist	Non-Timeboxing: bei nicht optimale Teamstruktur, weil man einen zeitlichen Spielraum hat.	Nicht optimale Teamstruktur	Kategorie 4

Tabelle 3: Kategorienbildung, Interview B1 Frage 5

Textaussage	Paraphrase	Generalisierung	Reduktion
wenn man wirklich geschaffen hat was auch schwierig ist, die Velocity definiert hat und dann auch eine gewisse Sprint-Dauer hat dann kann man sich schon die Timeboxed Methode entscheiden, weil man weiß Man kann den Task mit so und so viele Story Points in so vielen Tagen erledigen. Dann hat man die Timeboxed Methode, die einfach anzuwenden ist.	Timeboxing: bei bekannte Velocity und kurze Iterationen, wo man weiß das man gewisse Story Points in dieser Dauer schafft.	Timeboxing bei bekannte Velocity und kurze Iterationen verwenden	Kategorie 5: Velocity und Iterationsdauer
Jetzt ist es aber so dass es selbstverständlich im Projekt zu unerwarteten Vorfällen kommen könnte. Was die Ressourcen angeht. Das würde	Non-Timeboxing: bei zu viele unerwartete Ressourcenvorfälle im Projekt. Z.B.	Bei unerwarteten Ressourcenvorfällen (Teamreduzierung z.B.) Non-	Unterkategorie 5.1 Velocity

<p>heißen Die Kunden haben nicht mehr das Geld und senken die Anzahl der Mitarbeiter für das Team. Die Mitarbeiterinnen und Mitarbeiter im Team sind ausgefallen aus verschiedenen Gründen Krankheit, Pflegeurlaub, etc. die man nicht damit rechnet und auch im nächsten Fall, ich mein, man kann schon in einem gewissen Raum rechnen wann der Mitarbeiter Urlaub geht. Man kann die Planung des Urlaubs vielleicht drei Monate vorher auch schon einkalkulieren, aber manchmal kommt es auch dass der Mitarbeiter spontan Urlaub nehmen musste selbst drei Wochen vorher. Und das hat auch einen Einfluss darauf auf den Team Velocity deswegen ist es sehr schwer vorherzusagen was das Team Velocity wirklich sein kann. Man kann ungefähr sagen was aber wiederum nicht genug ist, um Timeboxed richtig einzusetzen.</p>	<p>Wunsch nach Teamreduzierung seitens Kunden, Ausfall von Mitarbeitern durch Kündigungen, unerwartete Pflegeurlaube, Krankheit etc.</p>	<p>Timeboxing verwenden</p>	
<p>die Länge des Sprints sehr wichtig. Derzeit ist es in vielen Projekten so dass man mit zwei Wochen rechnet und zwei Wochen sind durchaus unschaubare Tage, also sehr viele Tage, die schwer zu handhaben sind. Wenn man heute von einem Sprint redet in drei Tage Zyklus dann kann man das leichter handhaben</p>	<p>Timeboxing: bei 2 oder mehr als 2 Wochen Iterationen wird es schwieriger damit zu rechnen, da es durchaus undurchschaubar Tage sind, als wenn</p>	<p>Bei Iteration aus 2 Wochen oder weniger, Timeboxing verwenden.</p>	<p>Unterkategorie 5.2 Iterationsdauer</p>

<p>und leichter definieren okay in drei Tagen kann ich das und das, umsetzen aber wenn du zwei Wochen hast und in zwei Wochen kann vieles passieren kann vieles auf dich zukommen. Wenn wir wirklich einen überschaubaren Zeitrahmen haben kann man dann hier auch die Timeboxed Methode anwenden was dann auch vorteilhaft ist.</p>	<p>man z.B. 3 Tage Iteration hat, die überschaubar sind und mit denen man besser planen kann.</p>		
--	---	--	--

Tabelle 4: Kategorienbildung, Interview B1 Frage 6

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Optimal ist es das in der Go-Live Phase, das heißt kurz vor der Stabilitätsphase, es hat einen internen Go-Live gegeben und jetzt bin ich am Fixieren von Defekt etc. etc. Macht es Sinn, dass man hier einen Non-Timeboxed Ansatz verwendet, weil man auch nicht unbedingt damit rechnen kann wie viel Defekts zum Beispiel auf mich zukommen oder was muss noch schnell angepasst werden, was sind noch schnell die Wünsche von Kunden.</p>	<p>Non-Timeboxing: kurz vor Go-Live gehen, weil man nicht damit rechnen kann wie viel Defekts auf das Team zukommen oder was man noch schnell anpassen muss.</p>	<p>Non-Timeboxing bei Kurz vor Go-Live viele Defekts und kurzfristige Anpassungen</p>	<p>Kategorie 6: Go-Live</p>

<p>Es ist sehr schwer in einer Timeboxed Variante zu arbeiten, wenn es kritisch wird. Kurz vor der Umsetzung der Features. Das bedeutet Es hat Probleme gegeben und es wird kritisch mit der Zeit. Dann müsste man mehr tun als Timeboxed. Man müsste viel mehr geben und hier müsste man schauen, Timeboxed schafft ja auch eine Möglichkeit, dass der Mitarbeiter selbst weiß wie viel Zeit geben muss und wie viel er wie viele Stunden er jetzt noch geben muss. Aber kurz vor der Umsetzung eines Features das kritisch ist schafft man das nicht was auch vertraglich definiert ist. Manchmal muss man über den Rahmen gehen, um den Kunden das zu geben was auch versprochen wurde.</p>	<p>Timeboxing: wir schwer, wenn es kritisch wird, weil man mit möglichen Problemen nicht unbedingt rechnet, die auf einem zustoßen, die gelöst werden müssen. Und somit man mit Timeboxing nicht wirklich planen kann.</p>	<p>Timeboxing schwer in Go-Live Phase, da man nicht viel planen kann.</p>	<p>Kategorie 6</p>
--	--	---	--------------------

Tabelle 5: Kategorienbildung, Interview B1 Frage 7

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Wenn es zum Beispiel ein Projekt ist was das Entwicklungsteam bisher immer wieder gemacht hat und sich auskennt, wenn gewisse Frameworks angewendet wurden etc., etc. dann hat man eine gewisse Flow und Erfahrung und dann ist man schon mittendrin. Und das läuft. Das heißt Projekte die bereits öfters in der Art und Weise entwickelt wurden. Wie</p>	<p>Timeboxing: bei Projekten, die bereits öfters in der Art und Weise von einem Team entwickelt wurden, wo die ähnlichen Frameworks,</p>	<p>Timeboxing bei ähnlichen Projekten, die das Team schon kennt</p>	<p>Kategorie 7: Projektart Unterkategorie 7.1 Bekannte Projekte</p>

<p>gesagt in üblichen Projekten, die man bis jetzt hatte, dann ist es möglich die Timeboxing Methode anzuwenden.</p>	<p>Prozesse etc. angewandt wurden.</p>		
<p>Pilotprojekte sind kritische Fälle das bedeutet Es gibt jetzt weltweit neue Features neue Art und Weise wie es zum Beispiel Micro Services und die nicht mehr den alten Entwicklungsansatz vor allem sondern neu definiert wurden. Und man arbeitet schon mit Micro Services zum ersten Mal etc. Dann hat man hier auch eine Schwierigkeit zu sagen was nehme ich. Deswegen musste man sich auch nochmal anschauen da in dem Fall ist Timeboxing halt sehr schwer.</p>	<p>Non-Timeboxing: bei Pilotprojekten, wo neue Features, neue Ideen, die auf dem Markt noch nicht existieren überlegt und auch geforscht werden.</p>	<p>Non-Timeboxing bei Pilotprojekten</p>	<p>Unterkategorie 7.2 Pilot-, Forschungsprojekte und Prototypen Entwicklung</p>
<p>Wenn du aber zum Beispiel Projekte hast wie es in meinem alltäglichen Job der Fall ist, Support Projekte, Betriebs Projekte wo man auch sehr viel davon abhängt was der Kunde oder was den Dritt-Partner jetzt einlastet, was für Blockers, Inzidenz etc. einlastet werden. Damit kannst du ja nicht rechnen. Das sind Vorfälle, die kommen auf dich zu. Du kannst maximal ein paar Schätzungen machen wie viel auf mich zukommen wird. Aber in solchen Support und Betriebs und Wartungs-Projekten ist eine Timeboxing Ansatz unvorstellbar deswegen wird auch</p>	<p>Non-Timeboxing: bei Support-, Betriebs- und Wartungsprojekten, wo Tickets von Kunden und Dritt-Partner eingelastet werden, ist es schwer voraus zu planen.</p>	<p>Non-Timeboxing bei Support-, Betriebs- und Wartungsprojekten</p>	<p>Unterkategorie 7.3 Support-, Betriebs- und Wartungsprojekte</p>

<p>in vielen Support Teams und betrieb Teams Operations-Teams auch Kanban angewendet.</p>			
<p>Des Weiteren wenn du ein Produkt Entwicklungsunternehmen bist und ein Produkt für dich selbst entwickelt, hast du auf der einen Seite keine Kunde vielleicht der den zeitlichen Druck macht, aber du hast wahrscheinlich das Management-Team, das sagt Ich will zu dem Zeitraum fertig sein und ich will dieses Tool jetzt haben. Das kann schon problematisch werden. Andererseits muss zu schauen, dass du so wenig wie möglich an Ressourcen reinsteckst, weil das wird ja nicht von Externen bezahlt. Das wird von internen Finanzen Topf vom Budget vom internen Budget finanziert. Gerade bei solchen Projekten schaut man so wenig wie möglich Geld auszugeben. Hängt sehr stark davon ab was die Vorgaben sind für dieses Produkt aber eher auch Timeboxing würde ich sagen</p>	<p>Timeboxing: bei Entwicklung von eigenen Software-Produkten, hat man vielleicht nicht einen zeitlichen Druck seitens Kunden, aber wahrscheinlich vom Management-Team. Gerade bei solchen Projekten, wo die eigene Firma, es finanziert, wird so gut es geht an Ressourcen gespart da es vom internen Budget finanziert wird.</p>	<p>Timeboxing bei Entwicklung von eigenen Software-Produkten</p>	<p>Unterkategorie 7.4 Produkte und Projekte</p>

Tabelle 6: Kategorienbildung, Interview B1 Frage 8

Textaussage	Paraphrase	Generalisierung	Reduktion
-------------	------------	-----------------	-----------

<p>Der zeitliche Druck kann schon sehr viel mitspielen wird schon einen ziemlichen Einfluss haben auf die Qualität, selbstverständlich man kalkuliert Quality Assurance Zeit. Und es wird einen Test-Agent geben, der das testet, eine gewisse Qualität Checklisten durchführt. Aber wenn hier der Druck da ist, konzentriert man darauf, dass es mal funktioniert egal wie es auch funktioniert.</p>	<p>Timeboxing: die Qualität sinkt bei einem zeitlichen Druck automatisch, da man schneller fertig werden muss und nicht wirklich die Features gut testet.</p>	<p>Qualität sinkt durch den zeitlichen Druck bei Timeboxing</p>	<p>Kategorie 8: Qualität</p>
<p>Hat man in der Timeboxing Ansatz überhaupt Zeit für zum Beispiel Refactoring genommen. Dass man noch einmal durchgeht und durchschaut was man besser implementieren. Das man dann nur mehr kleine Workaround implementiert und nicht wirklich gescheit implementiert und dann kriegt irgendeinen Defekt rein den löst man auch nicht wirklich, sondern man schaut es irgendwie noch funktioniert aber das dahinterliegende Problem des Defekts analysiert man auch nicht tiefgründig. Weil es gibt keine Ressourcen, es gibt keine Zeit. Deswegen hat das Timeboxing aus meiner Sicht schon einen gewissen Einfluss darauf, dass dann die Qualität doch schlechter ist.</p>	<p>Timeboxing: man hat kaum Zeit für Refactoring. Man entwickelt immer wieder kleinere Workarounds um schneller fertig zu werden, anstatt die Probleme tiefgründigere zu analysieren.</p>	<p>kaum Zeit für Refactoring bei Timeboxing nur kleinere Workarounds, keine tiefgründigere Analyse bei Timeboxing</p>	<p>Kategorie 8: Qualität</p>

Tabelle 7: Kategorienbildung, Interview B1 Frage 9

KATEGORIE-BILDUNG INTERVIEW 2

Textaussage	Paraphrase	Generalisierung	Reduktion
Wenn man das Ganze wie einen großen Teig quasi ansieht, hat man nie einen Erfolg dazwischen, sondern man arbeitet einfach kontinuierlich weiter. Ich würde sagen, dass es was Psychologisches hat, wenn man es Timeboxed macht.	Non-Timeboxing: man hat nie einen Zwischenerfolg. Timeboxing ist psychologisch vorteilhafter.	Zwischenerfolg psychologisch vorteilhaft bei Timeboxing	Unterkategorie 2.2
Das heißt man hat immer einen Abschnitt betrachtet dem geht es gut oder schlecht beziehungsweise müssen wir zum Beispiel Gas geben oder hätten wir gut gemacht zum Beispiel, um die Qualität zu erhöhen oder ist es vielleicht wichtiger, dass man Arbeitspakete schneller umsetzt. Das sind eher die Sachen, die mich Richtung Timeboxing bringen würde.	Timeboxing: Raum zur Reflexion über die vergangene Iteration, um davon Lehren für die Zukunft zu ziehen.	Raum zur Reflexion durch Timeboxing	Kategorie 3.2

Tabelle 8: Kategorienbildung, Interview B2 Frage 3

Textaussage	Paraphrase	Generalisierung	Reduktion
Ich glaube es ist sehr motivierend, wenn man sieht was man geschafft hat. Wenn man das immer wieder kontinuierlich anfangt es zu betrachten. Das funktioniert meiner Meinung nach nur wenn du dir quasi Scheiben von den	Timeboxing: es ist motivierend, wenn man sieht das man einen Abschnitt /	Motivierend mit Iterationen zu arbeiten bei Timeboxing	Unterkategorie 2.2

Software Projekt runter schneidest und diesen Abschnitt anfängst zu betrachten.	Iteration geschafft hat.		
Es kann jetzt natürlich auch einen gewissen Druck ausüben, weil wenn du sagst okay ich muss schon so viele schaffen in diesem Abschnitt erzeugt das automatisch bzw. natürlich auch ein Stress im Team. Weil vielleicht schaffst du es nicht, weil es zu viel ist usw. Ich glaube, dass der Druck bzw. der Stress ist, der liegt. Da liegt eher bei Timeboxed.	Timeboxing kann einen Stress bzw. Druck im Team schaffen, wenn dem Team mehr Aufgaben gegeben werden, als sie schaffen könnten.	Stress bzw. Druck durch Überplanung bei Timeboxing	Unterkategorie 2.1

Tabelle 9: Kategorienbildung, Interview B2 Frage 4

Textaussage	Paraphrase	Generalisierung	Reduktion
Ich würde sagen es ist für Timeboxing. Ist es immer besser Du hast natürlich fixen Mitarbeitern mit Vollzeit. Kannst du viel besser planen. Vor einigen Jahren war es noch so dass viele Studenten einfach dabei waren gerade beim Testen von Arbeitspaketen. Und das hat immer dazu geführt, dass man vieles nicht abschließen können, weil du nicht richtig damit planen hast können. Es hängt davon ab Du hast ja Beispiel Teilzeit arbeitet 20 Stunden dann wirst du es immer haben, wenn er mal krank	Timeboxing: besser, wenn man fixe Mitarbeiter hat, da man z.B. mit studentischen Mitarbeitern oder externen Mitarbeitern schwer planen kannst. Auch bei Teilzeit Beschäftigte ist es schwer zu planen.	Fixe Mitarbeiter bei Timeboxing besser Bei studierendem Mitarbeiter, externe Mitarbeiter und Teilzeit Beschäftigte Non-Timeboxing besser	Unterkategorie 4.2 Unterkategorie 4.2

<p>ist und das macht sehr schwierigen einen Arbeitspaket abzuschließen.</p>			
<p>Ich glaube, dass Erfahrung gar nicht so eine große Rolle spielt, sondern dass vom Team davon abhängt wie gut es mit diesen Situationen umgehen kann. Erfahrung ist natürlich super. Wenn du wirklich ein Gebiet ein Genie bist. Aber wenn man jetzt so Standard Menschen nimmt und Senior und Junior dann hängt es davon ab meiner Meinung nach wie gut das Team miteinander interagiert. Umso besser die quasi ihre Aufgaben weitergeben können und so besser miteinander kommunizieren können. Umso leichter können Sie das noch abschließen. Ich glaube nicht, dass es so viel davon abhängt wie die Berufserfahrungen ist.</p>	<p>Erfahrung spielt keine große Rolle bei der Entscheidung. Die Frage ist eher wie geht das Team mit solchen Situationen um.</p>	<p>Erfahrung spielt keine Rolle</p>	<p>Unterkategorie 4.1</p>
<p>Wenn ich jetzt sage du hast einen Lead und der muss sich um seine Lead-Aufgaben kümmern hat aber zusätzlich auch noch eine Rolle in diesem Team. Dann ist immer die Frage wie er dies setzt. Wahrscheinlich kannst du auf das Projekt irgendwie sagen Okay 20 Prozent keine Ahnung und 80 Prozent Projekt. Aber das ist immer schwierig. Ich würde sagen es wäre sicher ein</p>	<p>Timeboxing: Vorteilhafter, wenn Mitarbeiter 100% in einem Projekt sind, und nicht zusätzliche Aufgaben in der Firma haben, sonst würde Non-Timeboxing mehr Sinn machen, da man</p>	<p>Bei Timeboxing besser, wenn Mitarbeiter 100% in einem Projekt sind</p>	<p>Unterkategorie 4.3</p>

Vorteil, wenn man im Non-Timeboxed gehen wird, aber dort hat man nicht so den Stress. Ich glaube bei Timeboxed das ist sehr schwierig, weil du musst Du musst einfach konstant Leistung zeigen.	diesen Zeitdruck nicht hat.		
---	-----------------------------	--	--

Tabelle 10: Kategorienbildung, Interview B2 Frage 5

Textaussage	Paraphrase	Generalisierung	Reduktion
wenn man jetzt zwei Wochen oder vier Wochen wäre das okay. Aber es gibt ja es gibt ja manches. Bei manchen Firmen gibts Sprints, die ein halbes Jahr dauern. Da wird es dann schwer Timeboxed zu verwenden wahrscheinlich, da wärest du Automatisch in Non- Timeboxed. Ich glaube, dass bei vier Wochen du schon eher dazu tendiert in Richtung Non-Timeboxed zu gehen, weil das schon so lange ist.	Timeboxing: bei 2 Wochen Iterationsdauer. Bei mehr als 4 Wochen wäre Non-Timeboxing besser, da man 4 Wochen schwer vorausplanen kann.	2 Wochen Iterationsdauer bei Timeboxing Bei mehr als 4 Wochen Non-Timeboxing	Unterkategorie 5.2 Unterkategorie 5.2

Tabelle 11: Kategorienbildung, Interview B2 Frage 6

Textaussage	Paraphrase	Generalisierung	Reduktion
Ich bin eigentlich ein Verfechter, dass wir trotzdem bei Timeboxed bleibt. Auch wenn mir persönlich vorkommt, dass es halt sehr schnell immer in dieses Setting hineingeht, dass man quasi man war lange im Timeboxed drinnen. Also	Non-Timeboxing: bei Go-Live Phase wo es sehr stressig wird, und man die letzten Arbeitspakete fertigstellen muss.	Schneller Arbeitspakete erledigen in der Go-Live Phase durch	Kategorie 6

speziell jetzt auch Scrum und Kanban betrachtet. Aber dann Switch du dann meistens hinüber, weil du denkst dir, es ist jetzt sowieso eine Kransh Phase. Wir müssen alle Überstunden machen und schauen, dass das weitergeht. Man versucht Zeit zu sparen. standardmäßig bei solchen Fällen Non-Timeboxed und schaut, dass man dieses Projekt fertig macht und diese Phase ist oder so anstrengend.	Man versucht durch Non-Timeboxing Zeit zu sparen in dem man auch auf diverse Timeboxed Meeting verzichtet.	Verzicht auf Meetings im Non-Timeboxing	
--	--	---	--

Tabelle 12: Kategorienbildung, Interview B2 Frage 7

Textaussage	Paraphrase	Generalisierung	Reduktion
bei Produkt wäre ich immer bei Timeboxed und Projekt ist es unterschiedlich hängt immer ein bisschen davon ab Tendenz zu Timeboxed, aber es gibt sicher Phasen wo es Non- Timeboxed ist bzw. Projekte wo es einfach mehr Sinn macht. Aber ich denke das muss man sich mal im Detail anschauen was man damit erreichen möchte.	Timeboxed: bei Produkt-Entwicklung und Projekte, wobei es Phasen (z.B. Go-Live Phase) gibt, wo Non-Timeboxing mehr Sinn macht.	Timeboxed bei Produkt-Entwicklung und Projekte	Unterkategorie 7.4
Ich denke bei Pilot-Projekten ist es würde ich sagen Es ist sehr wichtig, dass du schnell quasi etwas hin stellst damit	Timeboxing: bei Pilotprojekten, ist es wichtig, dass man am	Timeboxing bei Pilotprojekten	Unterkategorie 7.2

<p>muss man auch kann. Ich denke es ist einfach wichtig, dass man das sieht auch wenn es vielleicht am Anfang irgendwie so stellst du mal einfach nur ein Dreirad hin. Und nicht wirklich den Jet. Nur mal schauen ob das Sinn macht. Insofern würde ich quasi sagen. Auch da. Könnte man Richtung Timeboxed gehen. (..) Ich hätte ein bisschen die Angst, wenn es Non-Timeboxed, dass du zu lange quasi dran sitzt. Das man sich mehr Zeit lässt.</p>	<p>Anfang schon ein Prototyp auf die Beine stellt und nun mal schaut ob es Sinn macht weiter zu machen oder nicht. Sonst wäre die Angst bei Non-Timeboxed, dass es sich in die Länge zieht.</p>		
--	---	--	--

Tabelle 13: Kategorienbildung, Interview B2 Frage 8

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Ich würde ein bisschen dazu tendieren. Non-Timeboxed zu nehmen. Für die Qualität war es einfach mehr Zeit hast du und mehr Zeit heißt eigentlich mehr Qualität.</p>	<p>Non-Timeboxing: um die Qualität zu steigern. Mehr Zeit bedeutet mehr Qualität.</p>	<p>Qualität zu steigern durch mehr Zeit im Non-Timeboxing</p>	<p>Kategorie 8</p>
<p>Wobei das natürlich nicht sein muss. Was jetzt einfach Qualität betrifft ist ich fahre das Hardline. Es muss die Qualität muss auf jeden Fall einen sehr hohen Standard haben. Dann geht das sicher auch mit Scrum. Es hängt natürlich auch davon ab was für ein Produkt du entwickelst. Sagen wir jetzt Software für Flugzeuge wirst du ganz anders</p>	<p>Timeboxing: wenn man hohe Qualitätsstandards im Projekt hat, dann würde Timeboxing auch gut funktionieren.</p>	<p>Mithilfe von hohen Qualitätsstandards Timeboxing auch möglich</p>	<p>Kategorie 8</p>

rangehen als Software für irgendein Zip Programm.			
---	--	--	--

Tabelle 14: Kategorienbildung, Interview B2 Frage 9

KATEGORIE-BILDUNG INTERVIEW 3

Textaussage	Paraphrase	Generalisierung	Reduktion
würde ich mit einem geplanten Scope über zwei drei Iterationen Timeboxed vorziehen. Hätte ich diesen Planungshorizont nicht dann würde ich wahrscheinlich Non-Timeboxing vorziehen.	Timeboxing: vorteilhaft bei einem geplanten Scope. Wären die Iterationen nicht planbar, dann eher Non-Timeboxing.	Fixe planbare Planungsdauer im Timeboxing	Unterkategorie 5.2
Das ändert sich in der Scope in der Umsetzung bzw. Planungshorizont häufiger. Oder ich habe von der Projektsituation etwas wo man kurzzeitiger agieren muss, wie zum Beispiel Prototypen Entwicklung oder ähnliches dann würde man eher einen Non-Timeboxed Ansatz bevor ziehen. Das heißt wenn man auf die Gegebenheiten schneller reagieren muss.	Non-Timeboxing: bei Projektsituation, wo man kurzzeitig reagieren muss und wo sich der Planungsraum häufig ändert. Auch Prototypen Entwicklung.	Non-Timeboxing, wenn man kurzzeitig reagieren muss Non-Timeboxing bei Planungsraum der sich häufig ändert Non-Timeboxing bei Prototypen Entwicklung	Kategorie 7 Unterkategorie 7.2

Tabelle 15: Kategorienbildung, Interview B3 Frage 3

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Wenn man es bezieht aus dem Bereich ich hab einen Projekt das umgesetzt wird ist wahrscheinlich mit hoher Wahrscheinlichkeit sogar für alle Personen die in diesen Teams sitzen, Komfortabler wenn es über Iterationen geht, weil Scrum auch vorgibt dass die Planung steht dass was in der nächsten Iteration ist steht, dass sich die Leute darauf einstellen können was sie erwartet. Diese Transparenz ist für das Team natürlich von Vorteil. Und wirkt sich natürlich auf den Stresslevel positiv aus.</p>	<p>Timeboxing: bei einem Projektumsetzungsteam, ist es komfortabler, wenn es über Iterationen geht, weil Timeboxed auch vorgibt, dass die Planung der nächsten Iteration steht, damit sich die Mitarbeiter darauf einstellen können. Diese Transparenz ist wirkt sich positiv auf den Stresslevel aus.</p>	<p>Transparenz wirkt sich positiv auf den Stresslevel aus durch Timeboxing</p>	<p>Unterkategorie 2.1</p>
<p>I: Das heißt nicht, dass wenn ich in einem Timeboxed Ansatz bin ich eher gestresst bin, weil ich die Aufgaben fertig machen muss? Was nicht bei einem Non-Timeboxed Ansatz der Fall ist. B: Das Team bringt sich in die Planung des Sprints ganz klar ein. Wenn ich weiß wie meine Velocity wie meine Kapazitäten ausschauen. Wie ich mit der technischen Umsetzung umgehe, dann weiß ich worauf kann ich mich einlassen bzw. kann das auch entsprechend</p>	<p>Timeboxing: Das Team bringt sich in die Planung der Iteration so ein, dass es auch ein Puffer in der Iteration gibt, damit kein Zeitdruck zustande kommt.</p>	<p>Timeboxing, wenn man einen Puffer in der Iteration miteinplant</p>	<p>Unterkategorie 2.1</p>

<p>planen. Das Team ist da gefragt aktiv mitzugestalten und kann sozusagen diesen nächsten zwei Wochen richtig planen. Sie wissen was sie erwartet</p>			
--	--	--	--

Tabelle 16: Kategorienbildung, Interview B3 Frage 4

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Je mehr Einflüsse von außen wie Aushilftsmitarbeiter oder Nebenjobs oder Externe die in anderen Zeitzonen sitzen oder ähnliches. Man sollte die Transparenz im Team hochhalten. Man soll die Möglichkeiten geben allen Mitgliedern im Team ausreichende Informationen zukommen zu lassen. Aus dieser Perspektive her wirklich daran arbeiten zu können als gemeinsames Team ist das sicherlich von Vorteil, wenn man das getimeboxed macht. (..) Die Transparenz im Team ist aber eine viel Höhere also theoretisch gesehen sollte das Team an höheren Informationsfluss haben durch die Meetings durch das vorgegebene Framework. Sie stimmen sich besser ab sie haben einen besseren Informationsfluss, der vorgegeben ist.</p>	<p>Timeboxing: Durch die Transparenz im Team (durch diverse Meetings) können alle möglichen Teamsettings-Umstände einfacher gehandhabt werden.</p>	<p>Im Timeboxing lösen Meetings Kommunikations- & Planungsprobleme</p>	<p>Kategorie 4</p>

In Kanban zum Beispiel muss man das aus den Personen herausholen. Kanban oder Non-Timeboxed setzt einen höheren Reifegrad des Teams voraus als es Scrum tut.	Non-Timeboxing: setzt einen höheren Reifegrad des Teams voraus.	Non-Timeboxing bei höherem Reifegrad des Teams	Unterkategorie 4.1
--	---	--	--------------------

Tabelle 17: Kategorienbildung, Interview B3 Frage 5

Textaussage	Paraphrase	Generalisierung	Reduktion
Wir haben in unseren Teams normalerweise eine Dauer von zwei Wochen und hatten auch über Weihnachten und ähnliches eine Dauer von vier Wochen und die Praxis hat gezeigt, dass es fast nicht planbar war. Je kürzer desto besser und übersichtlich.	Timeboxing: zwei Wochen Iteration sind noch planbar. Je kürzer die Iteration desto planbarer. Alles über 2 Wochen ist nicht planbar.	2 Wochen Iteration bei Timeboxing Mehr als 2 Wochen dann Non-Timeboxing	Unterkategorie 5.2 Unterkategorie 5.2
Je stabiler das Team desto eher die Velocity desto eher kann ich sie feststellen. Wenn man jetzt wirklich Aushilfskräfte, Teilzeitkräfte denn sie sollen ja fixer Bestandteil des Teams sein. Das heißt Leute die ständig wechseln die machen das ganze schwieriger.	Timeboxing: bei einem stabilen Team da auch die Velocity stabiler ist. Je mehr Aushilfskräfte, Teilzeitkräfte desto schwieriger wird es sein die Velocity zu bestimmen.	Non-Timeboxing bei Aushilfskräften, Teilzeitkräfte die Vorhersage der Velocity erschweren.	Unterkategorie 5.1 Unterkategorie 4.2

Tabelle 18: Kategorienbildung, Interview B3 Frage 6

Textaussage	Paraphrase	Generalisierung	Reduktion
Es gibt einige Teams die in solchen Situationen vor dem Go-Live wirklich dann auf Kanban umsteigen. Weil sie sagen, sie sind damit effizient sie können schneller den Task oder Bugfixes oder ähnliches abarbeiten. Da gibt es einige Punkte zu berücksichtigen. Man erspart sich zwar die Meetings auf der einen Seite, aber man muss auf der anderen Seite gewährleisten, dass der Informationsfluss bestehen bleibt.	Non-Timeboxing: vor einer Go-Live Phase wäre es effizienter auf Non-Timeboxing umzusteigen, da man schneller die Tasks abarbeiten kann.	Non-Timeboxing damit man beim Go-Live schneller die Tasks abarbeitet	Kategorie 6

Tabelle 19: Kategorienbildung, Interview B3 Frage 7

Textaussage	Paraphrase	Generalisierung	Reduktion
Pilotprojekt würde ich wahrscheinlich eher kurz ist mit Non-Timeboxing entwickeln. Support und Wartungs-Projekte sind eher auch eher nach Non-Timeboxing	Non-Timeboxing: bei Pilotprojekten & Support- und Wartungsprojekte	Non-Timeboxing bei Pilotprojekten & Support- und Wartungsprojekte	Unterkategorie 7.2 Unterkategorie 7.3
Projekt bzw. Produkt eher Timeboxing	Timeboxing: bei Projekt und Produkt	Timeboxing bei Projekt und Produkt	Unterkategorie 7.4

Tabelle 20: Kategorienbildung, Interview B3 Frage 8

Textaussage	Paraphrase	Generalisierung	Reduktion
Man kann fast nichts sagen das man sagt wir können uns nicht Zeit lassen bis wir etwas abgeben, weil es ist Non-Timeboxed und Dadurch ist die Qualität höher. Diese Schlussfolgerung würde ich jetzt nicht ziehen.	Durch Non-Timeboxing würde sich die Qualität nicht verbessern. Da man trotzdem eine Deadline zum Einhalten hat.	Kein Entscheidungsmerkmal	Kategorie 8

Tabelle 21: Kategorienbildung, Interview B3 Frage 9

KATEGORIE-BILDUNG INTERVIEW 4

Textaussage	Paraphrase	Generalisierung	Reduktion
Es kommt immer auf das Projekt drauf an. Es gibt Projekte, die erfordern ein gewisses Launch oder Go-Live und das ist dann mit anderen Maßnahmen verbunden mit Marketing mit Geschäftseröffnung et cetera mit gewissen Förderungen die gewisse Phasen erreichen müssen. Und da ist es auf jeden Fall sinnvoll, dass man sich strikt an die Zeiten hält und da ist ein Timeboxing auf jeden Fall auch sinnvoll.	Timeboxing: bei Projekten mit einem Go-Live Termin, Förderprojekte, wo zu gewissen Phasen, gewisse Förderungen erreicht werden müssen.	Timeboxing, bei Projekten mit Deadlines	Unterkategorie 7.4

<p>I: Wann wäre Non-Timeboxing sinnvoll anzuwenden? B: In der klassischen Prototyping bzw. Produktentwicklung Modus. wenn es darum geht eine gewisse Idee zu bauen und da ist es meistens so dass man gewisse Sachen testet dann wieder entwickelt dann verfeinert und da ist sehr viel Forschung da. Meiner Meinung nach ist das nicht sinnvoll da Time zu boxen.</p>	<p>Non-Timeboxing: In der klassischen Prototyping bzw. Produktentwicklung Modus, wo noch Ideen generiert werden und verfeinert werden. Auch bei Projekten, wo sehr viel geforscht wird.</p>	<p>Non-Timeboxing beim klassischen Prototyping Non-Timeboxing bei Ideengenerierung & -verfeindung Non-Timeboxing bei Forschungsprojekte</p>	<p>Unterkategorie 7.2</p>
--	---	---	---------------------------

Tabelle 22: Kategorienbildung, Interview B4 Frage 3

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Es kommt immer darauf an wie realistisch dieses Timebox gesetzt werden. Wenn man einen, wenn man einen Spielraum quasi definiert ein gewisses Polster integriert in das Timeboxing dann sollte es keine psychischen Belastungen geben. Wenn man aber ein Timeboxing anwendet wo man gewisse Requirements gibt und diese abgeschätzt werden und man keinen Polster hat dann kann es schon zu einen gewissen Stress Faktor kommen und dadurch auch einen größeren Fehler verursachen.</p>	<p>Timeboxing: wenn man einen gewissen Puffer mit einkalkuliert, sollte es keine psychischen Belastungen geben.</p>	<p>Puffer mitkalkuliert beim Timeboxing</p>	<p>Kategorie 2.1</p>

Tabelle 23: Kategorienbildung, Interview B4 Frage 4

Textaussage	Paraphrase	Generalisierung	Reduktion
ich glaube es hat etwas mit einer gewissen Größe zu tun. Wenn es wirklich nur um die Entwicklung geht wie viele Entwickler Es gibt je größer das Team desto mehr Struktur braucht es desto mehr macht es Sinn Timeboxing anzuwenden weil gewisse Pakete von einem Teammitglied umgesetzt werden dann werden sie von einem anderen übernommen und dort weiter gemacht. Das heißt wenn ein Mitglied auf dem anderen warten muss, weil er einen gewissen Freiraum hat dann ist das auf jeden Fall Timeboxing das Passende.	Timeboxing: macht Sinn ab einer gewissen Teamgröße, da es mehr Struktur braucht. Vor allem wenn voneinander abhängige Arbeitspaketen von mehreren Teams entwickelt werden.	Timeboxing, ab einer gewissen Teamgröße Timeboxing bei voneinander abhängigen Teams	Unterkategorie 4.4 Teamgröße Unterkategorie 4.5 Sub-Teams

Tabelle 24: Kategorienbildung, Interview B4 Frage 5

Textaussage	Paraphrase	Generalisierung	Reduktion
Da würde ich glaube ich gleich anschließen und sagen das beste Modell eines Sprints, sind zwei Wochen und in diesen zwei Wochen sollten sollte man eigentlich die Timeboxing soweit anwenden, dass man zwei Wochen Zeit haben diese Pakete abzuarbeiten.	Timeboxing: wenn man 2 Wochen Iterationen hat.	2 Wochen Iterationen bei Timeboxing	Unterkategorie 5.2

Das heißt innerhalb von sechs Monate macht es keinen Sinn meiner Meinung nach Timeboxing zu machen.	Non-Timeboxing: bei längeren Zeiträumen.	Non-Timeboxing bei längeren Zeiträumen	Unterkategorie 5.2
---	--	--	--------------------

Tabelle 25: Kategorienbildung, Interview B4 Frage 6

Textaussage	Paraphrase	Generalisierung	Reduktion
Dann sollte das auf jeden Fall Timegeboxed werden, weil es einfach eine ganz andere Emotion dem Team übergebracht wird. Wir haben jetzt einen Monat den drucken wir durch und dann ist es vorbei. Und da ist auf jeden Fall Timeboxing eine Hilfe.	Timeboxing: Bei einer Go-Live Phase, damit man, diese letzten Iterationen besser planen und effizienter arbeiten kann.	Timeboxing um Go-Live Phase besser und effizienter planen	Kategorie 6

Tabelle 26: Kategorienbildung, Interview B4 Frage 7

Textaussage	Paraphrase	Generalisierung	Reduktion
Timeboxing würde ich eigentlich immer nur anwenden, wenn wir eine Zeit haben, die wir auf jeden Fall nicht überschreiten dürfen. Und das gilt bei allen möglichen Faktoren. Das heißt wenn ein Datum da ist und dieses Datum ist absolut nicht flexibel dürfte es nicht dann würde ich da immer Timeboxen und sonst nicht.	Timeboxing: Wenn man eine Deadline hat, die nicht überschreiten werden darf. Non-Timeboxing: wenn man keine Deadline hat.	Timeboxing, bei Projekten mit Deadlines Non-Timeboxing bei Projekten ohne Deadlines	Unterkategorie 7.4 Unterkategorie 7.2

<p>I: Ja aber zum Beispiel, du hast am Anfang auch erwähnt Forschungsprojekte. Da hast du nicht wirklich diese Deadlines und da kann man sich auch da braucht noch mehr Zeit, dass man, dass man die Sachen anschaut und dass man schaut wie die Sachen funktionieren deshalb hast du gesagt, dass du da Non- Timeboxed verwenden würdest. B: Da will ich auch dabeibleiben, aber das fehlt nicht unter dem Punkt, dass man Deadlines hat.</p>	<p>Non-Timeboxing: bei Forschungsprojekte</p>	<p>Non-Timeboxing bei Forschungsprojekten</p>	<p>Unterkategorie 7.2</p>
--	---	---	---------------------------

Tabelle 27: Kategorienbildung, Interview B4 Frage 8

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>die Qualität immer darunter leidet, wenn ein gewisser Stressfaktor da ist und vor allem das mit dem Timeboxing sehr gegeben ist. Also ich würde sogar behaupten, dass im Timeboxing die Qualität darunter leidet als wenn flexible Deadlines gesetzt sind.</p>	<p>Non-Timeboxing ist besser, um die Qualität zu erhöhen, da man diesen Stressfaktor nicht hat.</p>	<p>Qualität steigt sich durch Non-Timeboxing</p>	<p>Kategorie 8</p>

Tabelle 28: Kategorienbildung, Interview B4 Frage 9

KATEGORIE-BILDUNG INTERVIEW 5

Textaussage	Paraphrase	Generalisierung	Reduktion
da fließen natürlich viel mehr Aspekte hinein als ich die, dass ich jetzt alle auflisten könnte aber so einige wichtige Entscheidungskriterien werden zum Beispiel Projektart um was für ein Projekt es sich handelt. Genauso aber Team Konstellationen wie das Team aufgebaut ist und ob die Mitarbeiter nur aus Vollzeitkräften oder auch Aushilfskräften bestehen. Und die wesentliche Frage ist auch wie stabil ist das Team, in dem ich arbeite, kann es überhaupt mit einem mit diesem Team überhaupt planen	Mehrere Aspekte, Projektart um was für ein Projekt es sich handelt, genauso aber Team Konstellationen wie das Team aufgebaut ist und ob die Mitarbeiter nur aus Vollzeitkräften oder auch Aushilfskräften bestehen. Auch wie stabil das Team ist und ob ich damit planen kann.	Projektart Teamkonstellationen Aushilfskräfte Vollzeitkräfte Teamsetup, stabiles Team	Kategorie 7 Kategorie 4 Unterkategorie 4.2 Kategorie 4

Tabelle 29: Kategorienbildung, Interview B5 Frage 3

Textaussage	Paraphrase	Generalisierung	Reduktion
der psychologische Effekt sehr wohl bei diesen Ansätzen eine große Rolle spielt. . Bei timeboxing hast du z.B. diesen Druck im Team wenn das Team nicht gescheit seine Arbeit macht, nicht gescheit die Tasks schätzt, die Schätzung auch nicht ernst nimmt dann hast du auf	Bei timeboxing hast du z.B. diesen Druck im Team, weil man in einer Zeit engeren Rahmen sich befindet und die	Bei Timeboxing ist ein gewisser Druck und Stressfaktor gegeben. Bei Non-Timeboxing nicht.	Unterkategorie 2.1

<p>der einen Seite diesen Stressfaktor, dass am Ende des Sprints deine Tasks nicht fertig werden.</p> <p>Bei Non-Timeboxed hast du diesen Stressfaktor weniger, natürlich kann man irgendwann mal Releases und Deadlines, aber diese sind nicht alle 2 Wochen, was dem Mitarbeiter eine gewisse Freiheit gibt und ihn nicht unter Druck setzt.</p>	<p>Tasks fertig stellen soll.</p> <p>Bei Non-Timeboxing hat man diesen Stressfaktor weniger.</p>		
<p>bei Timeboxing den Vorteil, dass eine gewisse Motivation gegeben ist, wenn du siehst wie du eines Sprints immer wieder zeitgerecht und erfolgreich abschließt. Dieser Motivationsfaktor ist bei Non-Timeboxed Ansätze nicht gegeben, da man keine kleinen Abschnitte oder Iterationen hat, in denen man arbeitet.</p>	<p>Bei Timeboxing hat man den Vorteil, dass man eine Motivation hat, wenn man, diese kurze Iteration abschließt. Bei Non-Timeboxing hat man diesen Vorteil nicht.</p>	<p>Bei Timeboxing ist die Motivation gegeben, durch das abschließen von Iterationen, bei Non-Timeboxing nicht.</p>	<p>Unterkategorie 2.2</p>

Tabelle 30: Kategorienbildung, Interview B5 Frage 4

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Wenn man eine optimales Teamsetting hat, wo alle Mitarbeiter, fix angestellt waren, fast alle Vollzeit angestellt ist, wo alle Mitglieder des Teams sich auf die Team Aufgaben fokussieren und nicht weitere Firmenaufgaben erledigen müssen, dann ist das ein optimales Team</p>	<p>Timeboxing bei optimales Teamsetting, mit Vollzeit Kräfte, ohne weitere Firmenaufgaben.</p> <p>Wenn man mit dem</p>	<p>Timeboxing bei optimales Teamsetting</p>	<p>Kategorie 4</p>

mit dem man planen kann, und Teams mit denen man gut planen kann, sind super geeignet dafür Timeboxing Ansätze zu verwenden.	Teamplanen kann dann Timeboxing.		
Sehr oft hat man, was ich auch in meinen Projekten habe, Mitarbeiter, die Firmenweite Aufgaben haben, wo man sich schwer tun mit ihnen zu planen, weil sie zu diesem Meeting müssen, und da hinreisen müssen, und bei einem anderen Projekt aushelfen müssen und weil sie Leader Aufgaben haben.	Timeboxing schlecht bei Mitarbeitern mit anderen zusätzlichen Aufgaben.	Non-Timeboxing bei Mitarbeitern mit zusätzlichen Firmenaufgaben.	Unterkategorie 4.3
Aushilfskräfte sind genauso ein Punkt wo ich eher dazu tendiere, wenn ich viele Aushilfskräfte im Projekt habe, dass ich Non-Timeboxed Ansätze verwende, weil ich schwer mit diesen Mitarbeitern rechnen kann. Aber auch Teilzeit Kräfte, wenn es viele im Team sind, dann eher Non-Timeboxing.	Non-Timeboxing bei vielen Aushilfskräften und Teilzeitmitarbeiter im Projekt, weil man schwer mit ihnen rechnen kann.	Non-Timeboxing bei vielen Aushilfskräften und Teilzeitmitarbeiter	Unterkategorie 4.2
Ich glaube aber nicht, dass die Berufserfahrung einen Faktor für diese Entscheidung sein kann, da sowas immer in beiden Ansätzen gegeben ist und im Endeffekt das Teamintern gemanagt werden muss.	Berufserfahrung spielt keine große Rolle.	Berufserfahrung spielt keine große Rolle	Unterkategorie 4.1

Tabelle 31: Kategorienbildung, Interview B5 Frage 5

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Grundsätzlich sollte die Velocity getrackt werden, um in Zukunft anhand der zu erfahren wie viel das Team schafft und damit bei Timeboxed Ansätze plant. Die Frage ist, aber wie ist es, wenn ich kontinuierlich ein unstabiles Team habe, weil jemand kündigt, dann kommt jemand dazu, dann wechselt einer in ein anderes Team, dann kommen andere externe Mitarbeiter ins Team, dann will der Kunde das Team reduzieren usw. Diese Fälle sind real und derzeit in meinem Team sehr gegeben. Deshalb ist bei uns die Velocity kaum und sehr schwer vorzusagen. Was natürlich das ganze schwieriger macht einen Timeboxing Ansatz zu verwenden, weil man nicht weiß wie viele Story Points das Team schaffen kann und somit schwer damit planen kann.</p>	<p>Bei einem Team, wo ein regelmäßiger Wechsel stattfindet, wo die Mitarbeiter nicht fix bleiben ist die Velocity nicht track bar und somit Non-Timeboxing geeigneter.</p>	<p>Bei einem Unstabilen Team Non-Timeboxing, da man die Velocity nicht rechnen kann.</p>	<p>Unterkategorie 5.1</p>
<p>Wenn's um die Sprintdauer geht, dann wird's spannend. Eine Iteration von 2 Wochen ist gerade noch planbar aber erst wenn ich die Velocity kenne, Alles über 2 Wochen ist sehr unüberschaubar, und natürlich auch schwer planbar. Bei</p>	<p>Bei 2 Wochen Sprint wäre Timeboxing geeignet, weil ich damit noch planen kann.</p>	<p>Timeboxing: Bei Iterationen bis zu 2 Wochen</p>	<p>Unterkategorie 5.2</p>

Solchen Fällen dann eher Non-Timeboxing verwenden.	Bei mehr als 2 Wochen eher Non-Timeboxing weil der Zeitfenster unüberschaubar wird.	Non-Timeboxing: bei mehr als 2 Wochen	
--	---	---------------------------------------	--

Tabelle 32: Kategorienbildung, Interview B5 Frage 6

Textaussage	Paraphrase	Generalisierung	Reduktion
In unseren Projekten gehen die meisten Teams, wenn es anfängt stressiger zu werden vor einem Go-Live Termin, wo die Features noch fertig gestellt werden und wo sehr viele Bugs seitens der Kunden auch eingelastet werden, gehen die meistens Teams und sagen wir machen jetzt Non-Timeboxing. Der Grund dahinter ist, dass man erstens sehr viele Bugs reinbekommt und es schwer wird jetzt zu sagen ja dieser wichtige Bug wartet jetzt mal bis zum nächsten Sprint, denn ich kann nichts mehr im Sprint aufnehmen. Und zweitens, weil man effektiver und schneller entwickeln kann, wenn man am Tag nicht dauern von gewissen Scrum Meetings	Timeboxing wird schwer bei einer Go-Live Phase, weil man sehr viele Bugs und Features fixen bzw. implementieren muss und fertig stellen muss. Um schneller zu arbeiten gehen viele Teams zu Non-Timeboxing Ansätze	Non-Timeboxing bei Go-Live Phasen, um das Ziel schneller zu erreichen.	Kategorie 6

unterbrochen wird und somit auch weiterkommt bei der Implementierung.			
---	--	--	--

Tabelle 33: Kategorienbildung, Interview B5 Frage 7

Textaussage	Paraphrase	Generalisierung	Reduktion
ich glaube die Projektart ist ein ganz wichtiger Entscheidungskriterium. Zum Pilotprojekt zum Beispiel, hatten wir den Fall in der Firma, dass ein Kunde eine Technologie ausprobieren wollte in Kombination mit dem Produkt den wir entwickeln. (...) . Wir haben im Endeffekt versucht es in unserem Sprint mit aufzunehmen, aber mit der Zeit haben wir gemerkt es geht irgendwie nicht da wir es weder schätzen könne noch in einem Sprint oder mehreren damit fertig werden. In unserer Retrospektive haben wir dann entschieden, es in einem Non-Timeboxed Ansatz anzugehen, da es einfach mehr Sinn macht bei solchen Projekten.	Bei Pilotprojekten macht es mehr Sinn Non-Timeboxing zu verwenden, da man viel probieren, testen und forschen muss	Bei Pilotprojekten Non-Timeboxing verwenden	Unterkategorie 7.2
Ich würde Forschungsprojekte wie Pilotprojekte behandeln, da man einfach die gleichen Rahmenbedingungen hat	Forschungsprojekte genau so wie Pilotprojekte Non-Timeboxing verwenden.	Non-Timeboxing bei Forschungsprojekte	Unterkategorie 7.2

<p>Bei Projekt und Produktentwicklung wäre meiner Meinung nach, wenn du ein Team hast, mit dem du planen kannst, Timeboxing zu verwenden, weil da hast du die kurzen Abschnitte, da hast du die Releases und kannst so Schritt für Schritt es weiterentwickeln.</p>	<p>Timeboxing bei Projekt- und Produktentwicklung, um auch den Fortschritt zu verfolgen.</p>	<p>Timeboxing bei Projekt- und Produktentwicklung</p>	<p>Unterkategorie 7.4</p>
<p>Bei Wartungs- und Supportprojekten macht es durchaus Sinn Non-Timeboxing zu verwenden. In solchen Projekten geht's halt drum Probleme auf die Schnelle zu lösen, dem Kunden zu supporten usw. Und bei solchen Projekten kann man schwer Timeboxing, weil die Sachen auf einem plötzlich zukommen und man es lösen muss.</p>	<p>Non-Timeboxing bei Wartungs- und Supportprojekten, da man auf Probleme schneller reagieren kann und sie lösen kann.</p>	<p>Non-Timeboxing bei Wartungs- und Supportprojekten</p>	<p>Unterkategorie 7.3</p>

Tabelle 34: Kategorienbildung, Interview B5 Frage 8

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Ich würde sagen, ja es hat einen Einfluss, denn mehr Zeit heißt im Grunde mehr Qualität, somit würde ich schon sagen das beim Timeboxing die Qualität drunter leiden könnte, weil man will ja</p>	<p>Die Qualität sinkt und leidet unter Timeboxing sehr, da man immer schnell seine Tasks</p>	<p>Qualität leidet unter Timeboxing und steigt mit Non-Timeboxing</p>	<p>Kategorie 8</p>

<p>seinen Sprint schnell schnell fertig bekommen, alles am letzten Tag noch irgendwie testen, damit man den Sprint schafft. Und dieses schnell schnell testen, führt teilweise dazu, dass die Qualität drunter leidet und sinkt. Anders als bei einem Non-Timeboxing Ansatz, wo dieser Zeitdruck in kleineren Iterationen nicht gegeben ist.</p>	<p>abschließen muss, um den Sprint zu schaffen.</p> <p>Non-Timeboxing besser für die Qualität da man diesen Zeitdruck nicht hat.</p>		
--	--	--	--

Tabelle 35: Kategorienbildung, Interview B5 Frage 9

KATEGORIE-BILDUNG INTERVIEW 6

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Viel mehr sind meistens mehrere Unterschiedliche Kriterien drin involviert. Solche Kriterien können zum Beispiel Sachen sein wie Teamgröße, Teamkonstellationen auch ob mehrere Sub-Teams an einem Produkt oder Projekt zusammenarbeiten, aber genauso was der Kunde vorgibt oder gern hätte.</p>	<p>Mehrere Kriterien wie Teamgröße, Teamkonstellationen auch ob mehrere Sub-Teams an einem Projekt zusammenarbeiten.</p>	<p>Teamgröße</p> <p>Teamkonstellationen</p> <p>mehrere Sub-Teams an einem Projekt arbeiten</p>	<p>Unterkategorie 4.4</p> <p>Kategorie 4</p> <p>Unterkategorie 4.5</p>
<p>Möchte der Kunde für sich kontinuierlich einen Überblick schaffen und regelmäßig Abgeschlossenen Iterationen haben?, dann kommt eher Timeboxing in Frage. Ist es dem Kunden wurscht Hauptsache er hat sein</p>	<p>Timeboxing: kontinuierlich einen Überblick für den Kunden schaffen</p>	<p>Kontinuierlichen Überblick für den Kunden</p>	<p>Kategorie 1</p>

Endprodukt zu der Deadline könnte auch Non-Timeboxing in Frage kommen.			
Auch Sachen wie was für ein Projektsetting das ist. Ist es eher ein Forschungsprojekt oder ein gängiges Projekt.	Die Projektart spielt auch eine Rolle ob man Timeboxed oder nicht.	Projektart wie Forschungsprojekt oder ein gängiges Projekt.	Kategorie 7

Tabelle 36: Kategorienbildung, Interview B6 Frage 3

Textaussage	Paraphrase	Generalisierung	Reduktion
Natürlich kann man sagen Timeboxing ist eher stressiger und druckvoller als Non-Timeboxing, weil man in einer gewissen Zeit mit gewissen Aufgaben fertig sein muss. Es ist auch um einiges stressiger, aber die Frage ist wie geht das gesamte Team damit um, setzen wir uns ein Polster bzw. einen Buffer, um nicht in diesem Druck gedrängt zu werden oder nicht.	Timeboxing: ist eher stressiger und druckvoller als Non-Timeboxing. Das Team muss damit umgehen, z.B. Buffer setzen.	Stress bzw. Druck durch Überplanung bei Timeboxing	Unterkategorie 2.1
Und natürlich ist Timeboxing motivierender als Non-Timeboxing, weil du die gewissen Slots oder Iterationen hast, wo man Sachen erledigt und wo man motivierend in die nächste Iteration geht, weil man die letzte Iteration erfolgreich abgeschlossen hat.	Timeboxing kann viel motivierender als Non-Timeboxing sein, wenn man die geplanten Iterationen abschließt.	Motivation des Teams durch Timeboxing	Unterkategorie 2.2

Tabelle 37: Kategorienbildung, Interview B6 Frage 4

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Ja eben ein Kriterium wäre eben die Frage des Teams. In welche Teamgröße arbeite ich. Habe ich ein Team, das ziemlich groß ist, sprich 5 plus, dann braucht das Team mehr Struktur, und mehr Struktur bedeutet Timeboxing. Teams, wo der Reifegrad sehr hoch ist und wo die Kommunikation ohne Meetings und ohne sämtliche Treffs funktioniert, könnte auch mit Non-Timeboxing arbeiten.</p>	<p>Größere Teams brauchen mehr Struktur. Mehr Struktur bedeutet ein Timeboxing Ansatz.</p> <p>Bei einem höheren Reifegrad des Teams könnte auch Non-Timeboxing funktionieren</p>	<p>Timeboxing besser bei größeren Teams</p> <p>Non-Timeboxing würde bei einem höheren Reifegrad des Teams funktionieren</p>	<p>Unterkategorie 4.4</p> <p>Unterkategorie 4.1</p>
<p>Genauso ist die Frage arbeite ich mit mehreren Sub-Teams ob nur ein Team. Wir haben z.B. 2 Sub-Teams, eines ist ein reines Frontend Team und eines ist ein Backend Team. Bei Sub-Teams oder bei Abteilungen, wo die Aufgaben von einander abhängen ist Timeboxing perfekt geeignet. Denn dadurch kann man planen, was man macht und was nicht, und muss nicht ewig auf eine Sache von dem anderen Team warten, weil sie eben Non-Time boxen.</p>	<p>Bei Zusammenarbeit mit mehreren Sub-Teams oder Abteilungen, wo die Aufgaben von einander abhängen ist Timeboxing perfekt geeignet.</p>	<p>Timeboxing bei Zusammenarbeit mit mehreren Sub-Teams oder Abteilungen.</p>	<p>Unterkategorie 4.5</p>
<p>Als erstes möchte ich mal die Erfahrung ausschließen. Ich finde nicht das die Frage etwas mit Erfahrung zu tun hat,</p>	<p>Erfahrung spielt keine Rolle bei dieser Fragestellung.</p>	<p>Erfahrung spielt keine Rolle</p>	<p>Unterkategorie 4.1</p>

weil im Endeffekt hat man immer Teams mit unterschiedlichen Niveaus, mit verschiedenen Levels an Seniorität und Erfahrung.			
Was aber die Anstellungsart angeht, so spielt es auf jeden Fall eine Rolle. Wenn man viel zu viele Aushilfskräfte im Team hat, die nicht immer da sind oder externe Mitarbeiter hat, mit ihnen kann man ganz einfach nicht planen, und da wird es dann schwer Timeboxing anzuwenden.	Bei viel zu viele Aushilfskräfte im Team macht es keinen Sinn Timeboxing zu verwenden, weil man schwer mit ihnen planen kann.	Non-Timeboxing bei viel zu viele Aushilfskräfte im Team	Unterkategorie 4.2

Tabelle 38: Kategorienbildung, Interview B6 Frage 5

Textaussage	Paraphrase	Generalisierung	Reduktion
Ja die Velocity ist ein breites Thema, es kann mit der Entscheidung zu tun haben, wenn man ein inkonstantes Team hat. Ein Team, das eben nicht aus fixen Mitarbeitern besteht, sondern nur immer externe Mitarbeiter dazu holt, wenn sie gebraucht werden. In so einem Team kann man die Velocity sehr schwer messen bzw. tracken, weil man jede Iteration andere Personen am Board hat. In solchen Situationen macht natürlich Non-Timeboxing mehr Sinn.	Non-Timeboxing: Wenn man ein inkonstantes Team hat, wo man mit externen Mitarbeitern arbeitet, ist schwer die Velocity zu messen.	Non-Timeboxing: wenn man ein inkonstantes Team hat, wo man die Velocity nicht messen kann.	Unterkategorie 5.1

<p>Bei Timeboxing Ansätze wie Scrum, ist die übliche Iterationsdauer 2 Wochen. Bei 2 Wochen macht es gerade noch Sinn Timeboxing zu verwenden, wird diese Zeit mehr, 3 oder 4 oder mehr Wochen so sollte man eher dazu gehen und Non-Timeboxing verwenden. (..)</p> <p>Wir merken jedes Jahr, wenn wir dabei sind, diese längere Iteration zu planen, dass die Planung einfach nicht funktioniert. Es macht kein Sinn für 5 Wochen vor auszuplanen, denn in diese 4 bis 5 Wochen kann vieles passieren, mit dem man nicht rechnen kann.</p>	<p>Timeboxing: bei Iterationsdauern von bis 2 Wochen. Denn damit kann man noch rechnen und planen.</p> <p>Non-Timeboxing: wenn die Dauer der Iterationen länger als 2 Wochen wird, weil man so ein längerer Zeitraum schwer genau vorausplanen kann.</p>	<p>Timeboxing: bei Iterationsdauern von bis 2 Wochen</p> <p>Non-Timeboxing: bei Iterationsdauern von mehr als 2 Wochen</p>	<p>Unterkategorie 5.2</p>
---	--	--	---------------------------

Tabelle 39: Kategorienbildung, Interview B6 Frage 6

Textaussage	Paraphrase	Generalisierung	Reduktion
<p>Teams in so eine stressige Phase wie, die Go-Live Phase, dazu tendieren eher Non-Timeboxing Ansätze zu verwenden. Und in diesem ganzen Stress und Druck haben die Teams verschiedene Meetings ausfallen lassen, haben alles in einem Board gepackt und</p>	<p>In einer Go-Live Phase, wo es ziemlich stressig wird, macht es mehr Sinn Non-Timeboxing zu verwenden um das</p>	<p>Non-Timeboxing bei Go-Live Phasen, um das Ziel zu erreichen.</p>	<p>Kategorie 6</p>

als Ziel gesetzt, dies alles muss bis zu der gesetzten Deadline fertig werden.	Hauptziel zu erreichen.		
--	-------------------------	--	--

Tabelle 40: Kategorienbildung, Interview B6 Frage 7

Textaussage	Paraphrase	Generalisierung	Reduktion
Die Projektart und das Projektsetting spielen bei dieser Frage eine große Rolle. Timeboxing würde ich primär bei gängigen Projekten verwenden. Auch bei Produktentwicklung und Projekt egal ob intern oder extern würde ich Timeboxing verwenden, wenn ich damit planen kann und die Rahmenbedingungen so gesetzt sind für die Vorausplanung.	Timeboxing: bei gängigen Projekten, aber auch bei der Produktentwicklung.	Timeboxing: bei gängigen Projekten, aber auch bei der Produktentwicklung	Unterkategorie 7.1 Unterkategorie 7.4
Bei Wartungsprojekte und Supportprojekte, wo es drum geht, dem Kunden zu supporten und einige Wartungsaufgaben zu übernehmen, würde ich eher Non-Timeboxing Ansätze verwenden, weil man bei solchen Projekten ganz einfach wenig bis kaum vorausplanen kann. Es kommen einfach Aufgaben, die kurzfristig gemacht werden müssen, und da kann man eben schwer vorausplanen,	Non-Timeboxing: Support- und Wartungsprojekte	Non-Timeboxing: bei Support- und Wartungsprojekte	Unterkategorie 7.3

was in den nächsten 2 Iterationen gemacht wird.			
Genauso würde ich bei Forschungs- und Pilotprojekte Non-Timeboxing verwenden, weil man bei solchen Projekten einfach mal neue Technologien ausprobiert, Sachen austestet, um diese vielleicht dann im Produkt oder Projekt zu adaptieren. Bei Forschung kann man eben schwer vorausplanen, ich würde sagen man setzt bei solchen Projekten ganz einfach Meilensteine, damit auch die Forschung nicht allzu lang dauert und man in absehbarer Zeit zu einem Ergebnis kommt.	Non-Timeboxing bei Pilotprojekten und Forschungsprojekten	Non-Timeboxing bei Pilotprojekten und Forschungsprojekten	Unterkategorie 7.2

Tabelle 41: Kategorienbildung, Interview B6 Frage 8

Textaussage	Paraphrase	Generalisierung	Reduktion
Natürlich spielt die Zeit da auch mit, weil durch mehr Zeit hat man im Grunde mehr Qualität. Weil man sich eben dann Zeit nehmen kann, um die Features ausgiebig zu testen.	Durch Non-Timeboxing hat man natürlich mehr Zeit für die Qualität und dadurch steigt auch die Qualität. Nicht nur durch das Testing	Qualität steigt sich durch Non-Timeboxing (Refactoring, Testing)	Kategorie 8

Was hier auch noch für die Qualität dazu kommt ist das Refactoring. Mit der Zeit braucht man immer wieder Zeit dafür um den Code zu refactoren, weil einfach so viel dazu gekommen ist, dass der Code nicht mehr lesbar ist.	sondern auch durch das Refactoring.		
--	-------------------------------------	--	--

Tabelle 42: Kategorienbildung, Interview B6 Frage 9

ENDGÜLTIGES KATEGORIENSYSTEM

Nr.	Kategorie	Unterkategorie
1	Überblick über das Projekt	
2	Psychische Effekte	2.1 zeitlicher Druck 2.2 Motivation
3	Innovation	3.1 Kreativität 3.2 Weiterentwicklung
4	Teamsetting	4.1 Erfahrung 4.2 Anstellungsart 4.3 100% in einem Projekt 4.4 Teamgröße 4.5 Sub-Teams

5	Velocity und Iterationsdauer	5.1 Velocity 5.2 Iterationsdauer
6	Go Live	
7	Projektart	7.1 bekannte Projekte 7.2 Pilot-, Forschungsprojekte und Prototypen Entwicklung 7.3 Support-, Wartungs- und Betriebsprojekte 7.4 Produkte und Projekte
8	Qualität	

Tabelle 29: Endgültiges Kategoriensystem