



universität
wien

DIPLOMARBEIT

TITEL DER DIPLOMARBEIT

ANALYSE VON AGILEN MANAGEMENT ANSÄTZEN AM BEISPIEL DES INTERNATIONALEN HANDSET MANAGEMENTS

Verfasser:

Oliver Maurovich

zur Erlangung des akademischen Grades

Magister rerum socialium oeconomicarumque (Mag.rer.soc.oec.)

Wien, 2008

Studienkennzahl lt. Studienblatt: A175

Studienrichtung lt. Studienblatt: Wirtschaftsinformatik

Betreuerin: ao. Univ.-Prof. Dr.techn. Dipl.-Ing. Renate Motschnig

Danksagungen

An dieser Stelle möchte ich mich bei all jenen bedanken, die mich während meinem Studium unterstützt haben.

Besonderer Dank gebührt meinen Eltern Sylvia und Simon Maurovich, die mich in meiner gesamten Schul- und Studienzeit unterstützten.

Weiters Bedanke ich mich bei Frau ao. Univ.-Prof. Dr.techn. Dipl.-Ing. Renate Motschnig und Dr. Georg Wolf für die Betreuung meiner Diplomarbeit und den zahlreichen wissenschaftlichen Ratschlägen, welche stets zur Verbesserung der Arbeit beigetragen haben, und auch meinen Arbeitskollegen, die in Diskussionen wertvolle Beiträge einbrachten.

Inhaltsverzeichnis

Inhaltsverzeichnis	3
1 Einführung	5
2 Beschreibung/Überblick Agile vs. Klassische Methoden	6
2.1 Methoden der Softwareentwicklung.....	6
2.2 Klassische Softwareentwicklung	8
2.2.1 Wasserfallmodell.....	9
2.2.2 Spiralmodell	11
2.2.3 V - Modell	13
2.2.4 Eigenschaften	14
2.2.5 Änderungskostenkurve Klassische Softwareentwicklung.....	16
2.3 Agile Softwareentwicklung	17
2.3.1 Das Agile Manifest.....	17
2.3.2 Eigenschaften	19
2.3.3 Kostenkurve Agile Softwareentwicklung.....	22
2.3.4 Beispiel Scrum	23
2.4 Agile gegenüber klassische Softwareentwicklung	29
2.4.1 Zeiträume.....	29
2.4.2 Dokumente	30
2.4.3 Modellierung	30
2.4.4 Personelle Projektausstattung.....	30
2.4.5 Qualifikationen.....	31
2.4.6 Ergebnis.....	31
2.4.7 Zielgrößen	31

3	Prozesse in der Praxis	37
3.1	Der CVT Prozess	37
3.1.1	Prozessabbildung	39
3.1.2	Prozessbeschreibung	41
3.1.3	Mögliche Probleme	47
4	Persönliche Einschätzung agiles Management im Bereich des internationalen Handset Managements 77	
4.1	Aussage dieser Arbeit	77
4.2	Verbesserungen und Wissensgewinne	77
4.3	Soziale Verbesserungen	79
4.3.1	Kritische Diskussion	79
4.4	Diskursive Reflexion	80
4.4.1	Fragestellungen	80
4.4.2	Ergebnisse	81
5	Zusammenfassung	85
5.1	Persönliche Einsichten	85
6	Glossar	88
7	Abbildungsindex	89
8	Literatur	91
	Anhang A Darstellung in UML	94
	Anhang B Abstract & Lebenslauf	100

1 Einführung

Ich habe mich für dieses Thema entschieden, da es für mich wichtig ist die derzeitige Arbeitsweise in unserem Unternehmen anhand der Theorie und Praxis erklären zu können, das Verständnis für Agilität betreffend der Arbeits- und Vorgehensweise zu durchleuchten, verstehen warum viele Prozesse bei uns effektiver sind als bei anderen Firmen. Besonders in meinem Arbeitsbereich, dem Handset Management möchte ich zeigen, dass durch Einfluss von Agilität eine Verringerung der Durchlaufzeit von Prozessen möglich ist und dies auch Einfluss auf die Kommunikation und die Zusammenarbeit zwischen den Mitarbeitern und diversen Teams innerhalb eines Unternehmens haben kann.

Die Diplomarbeit gliedert sich prinzipiell in zwei verschiedene Teile auf. Erstens wird die Theorie der klassischen und agilen Methoden bei der Software Entwicklung erklärt und die Unterschiede versucht dem Leser durch Aufbereitung verschiedenster Literatur näher zu bringen. Ich möchte hier noch darauf hinweisen, dass zum Beispiel Autoren wie Winston Royce und Barry W. Boehm die klassische Softwareentwicklung behandeln, Wikipedia jedoch besonders verständliche und aktuelle Erklärungen gibt, auf die in dieser Arbeit mit Schwerpunkt Praxis Bezug genommen wird. Der zweite Teil befasst sich mit der praktischen Arbeit in unserem Unternehmen, wie Agilität Prozesse verbessern kann. Hier werden Prozesstrukturen und Beschreibungen festgehalten, die bisher in unserem Unternehmen zwar gelebt, jedoch nie visualisiert und genau ausdefiniert wurden. Ich versuche mit diesen Darstellungen den praktischen Unterschied zwischen klassischer und agiler Methodik zu beschreiben um dem Leser ein umfangreiches Beispiel hierfür bieten zu können. Danach gehe ich auf die daraus gewonnenen Erkenntnisse ein, auf die Auswirkungen und Möglichkeiten die Agilität meiner Ansicht nach in unserem Bereich bietet. Können aufgrund der agilen Beispiele und Erfahrungen auch Rückschlüsse auf Kommunikation und die Teambeschaffenheit gezogen werden? Danach führe ich mit einem Querschnitt meiner Kollegen Interviews, um ihnen Fragen zu ihrer Erfahrung mit Agilität stellen, und wie sie bestimmte Themen für sich persönlich bewerten.

2 Beschreibung/Überblick Agile vs. Klassische Methoden

2.1 Methoden der Softwareentwicklung

Martin Fowler (2005) meint, die Entwicklung von Software ist von Natur aus eine chaotische Angelegenheit, meistens nach dem Motto „code and fix“: es existiert kein Plan, wie die Software im Endzustand aussehen soll, das Design ist ein Flickwerk von vielen kurzsichtigen Entschlüssen. Dies mag für kleine Systeme funktionieren (bis ca. 300 Codezeilen und ein Entwickler), allerdings ist bei größeren Projekten (mehr als 300 Codezeilen und mehrere Entwickler) dieses Vorgehen zum Scheitern verurteilt. Es ist schwierig bis unmöglich, das System mit neuer Funktionalität auszustatten, Fehler und Defekte werden zunehmend schwerer zu beheben sein. Typisch für nach diesem Prinzip gefertigte Software ist eine lange Testphase, nachdem das System „fertig“ ist. Da die Dauer für das Testen und die darauf folgende Fehlerbehebung nicht vorhersehbar sind, ist eine solche Testphase nicht genau zu planen.

Laut Fowler (2005) lebte man lange mit dieser Art Software zu entwickeln, aber es existiert ein Gegenmittel gegen diese chaotischen Zustände: Methoden. Methoden zwingen der Softwareentwicklung einen disziplinierten Prozess auf. Dies hat zum Ziel, die Entwicklung vorhersehbarer und effizienter zu machen. Ermöglicht wird das mit einem detaillierten Prozess, der seinen Schwerpunkt auf die Planung legt, inspiriert von anderen Ingenieursdisziplinen (Bauingenieure etc). Man spricht hier deshalb auch von Ingenieurmethoden (engl. engineering methodology).

Der organisatorische Rahmen für die Softwareentwicklung wird laut Cockburn (2002) durch ein Prozessmodell beschrieben:

- Phasen und Reihenfolge des Arbeitsablaufes
- jeweils durchzuführende Aktivitäten

- Verantwortlichkeiten und Konsequenzen
- Definition ihrer Anforderungen und der Teilprodukte
- erforderliche Inputs und ihrer Anforderungen
- notwendige Mitarbeiterqualifikationen
- anzuwendende Standards, Richtlinien, Methoden und Werkzeuge

Ingenieurmethoden existieren schon lange Zeit, jedoch blieb ihnen laut Fowler (2005) der voll durchschlagende Erfolg verwehrt. Des Weiteren sind sie unpopulär, da sie viel Aufwand verursachen und sehr bürokratisch sind, wodurch die Geschwindigkeit der Entwicklung verlangsamt wird.

In den letzten Jahren ist als Reaktion auf diese Methoden eine Reihe von neuen Ansätzen aufgetaucht. Diese waren eine Zeit lang unter dem Begriff „lightweight methodologies“ (leichtgewichtige Methoden) bekannt (vgl. Cockburn 2002), jedoch der geläufige Ausdruck ist mittlerweile „agile Methoden“.

Für viele Personen ist das Aufkommen dieser neuen Ansätze als Reaktion auf die Bürokratie der Ingenieurmethoden zu sehen. Die neuen Methoden versuchen laut Fowler (2005) einen brauchbaren Kompromiss zwischen zu viel Prozess (Ingenieurmethoden) und keinem Prozess (Chaos) zu erzielen, indem sie genau den richtigen Anteil an Prozess annähern, um ein sinnvolles Resultat zu erhalten.

„Die Agile Softwareentwicklung ist eine Gegenbewegung zu den oft als schwergewichtig und bürokratisch angesehenen traditionellen Softwareentwicklungsprozessen [...]“, heißt es bei Wikipedia. Die Regeln und Vorgehensweisen des klassischen Projektmanagements werden auf einige wenige Regeln und Praktiken reduziert.

Die methodische Softwareentwicklung wird heute primär in zwei Arten unterteilt:

- Entwicklung nach klassischen Methoden
- Entwicklung nach agilen Methoden

Da in der Softwareentwicklung der Begriff der Anforderung vorkommt, möchte ich hier eine kurze Erklärung des Begriffs festhalten.

Definition einer Anforderung:

1. Eine Bedingung oder Fähigkeit einer Person, die zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
2. Eine Bedingung oder Fähigkeit, die eine Software erfüllen oder besitzen muss, um einen Vertrag, eine Norm oder ein anderes, formell bestimmtes Dokument zu erfüllen.

(IEEE 310.12-1990)

2.2 Klassische Softwareentwicklung

Die Methoden der klassischen Softwareentwicklung basieren auf der gleichen Philosophie und haben ähnliche Schwerpunkte, angefangen mit den Ahnen der Prozessmodelle wie zum Beispiel das Wasserfall- oder Spiralmodell, bis hin zu den Weiterentwicklungen wie das V-Modell¹ oder der Rational Unified Process (RUP)².

¹ <http://www.v-modell.iabg.de/>

² <http://www.rational.com/products/rup/index.jsp>

2.2.1 Wasserfallmodell

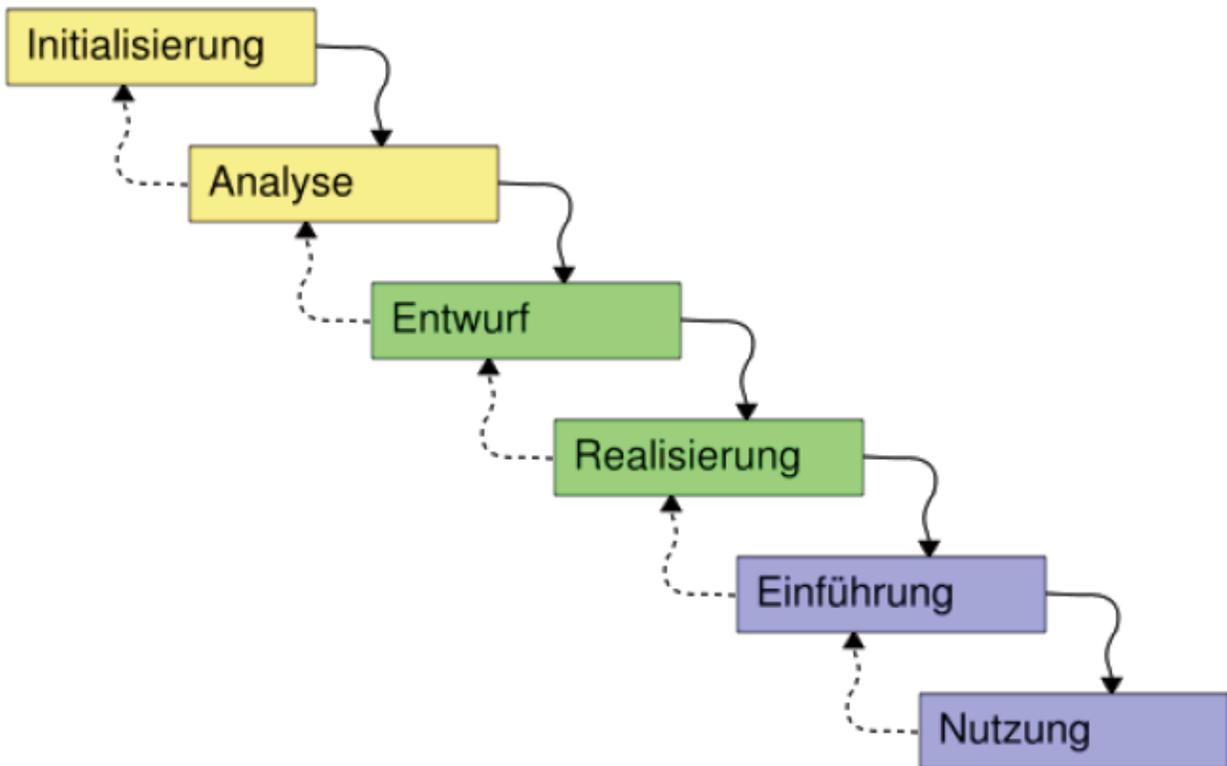


Abbildung 1 Wasserfall Modell

(Quelle: <http://de.wikipedia.org/wiki/Wasserfallmodell>)

In Wikipedia ist folgendes über das Wasserfallmodell (vgl. Abbildung 1) zu lesen: „Das Wasserfallmodell ist ein lineares (nicht-iteratives) Vorgehensmodell in der Softwareentwicklung, bei dem der Softwareentwicklungsprozess in Phasen organisiert wird. Dabei gehen die Phasenergebnisse wie bei einem Wasserfall immer als bindende Vorgaben für die nächst tiefere Phase ein.“

Im Wasserfallmodell hat jede Phase vordefinierte Start- und Endpunkte mit eindeutig definierten Ergebnissen. In Meilensteinsitzungen am jeweiligen Phasenende werden die Ergebnisdokumente

verabschiedet. Zu den wichtigsten Dokumenten zählen dabei das Lastenheft sowie das Pflichtenheft. In der betrieblichen Praxis gibt es viele Varianten des reinen Modells. Es ist aber das traditionell am weitesten verbreitete Vorgehensmodell.

Der Name „Wasserfall“ kommt von der häufig gewählten grafischen Darstellung der fünf bis sechs als Kaskade angeordneten Phasen.

Erweiterungen des einfachen Modells (Wasserfallmodell mit Rücksprung) führen iterative Aspekte ein und erlauben ein schrittweises „Aufwärtslaufen“ der Kaskade, sofern in der aktuellen Phase etwas schief laufen sollte, um den Fehler auf der nächst höheren Stufe beheben zu können.

Das Wasserfallmodell wird allgemein dort vorteilhaft angewendet, wo sich Anforderungen, Leistungen und Abläufe in der Planungsphase relativ präzise beschreiben lassen.“

Winston Royce (1970) beschreibt das Wasserfallmodell als fehlerträchtiges und mit einem Kostenrisiko behaftetes Modell der Softwareentwicklung - dabei bezieht er sich laut Wikipedia sowohl auf die einfache Variante als auch auf die erweiterte, mit schrittweisen Rücksprungmöglichkeiten. Stattdessen schlägt Royce gemäss Wikipedia in dieser Publikation ein wesentlich um iterative Elemente erweitertes Modell vor.

2.2.2 Spiralmodell

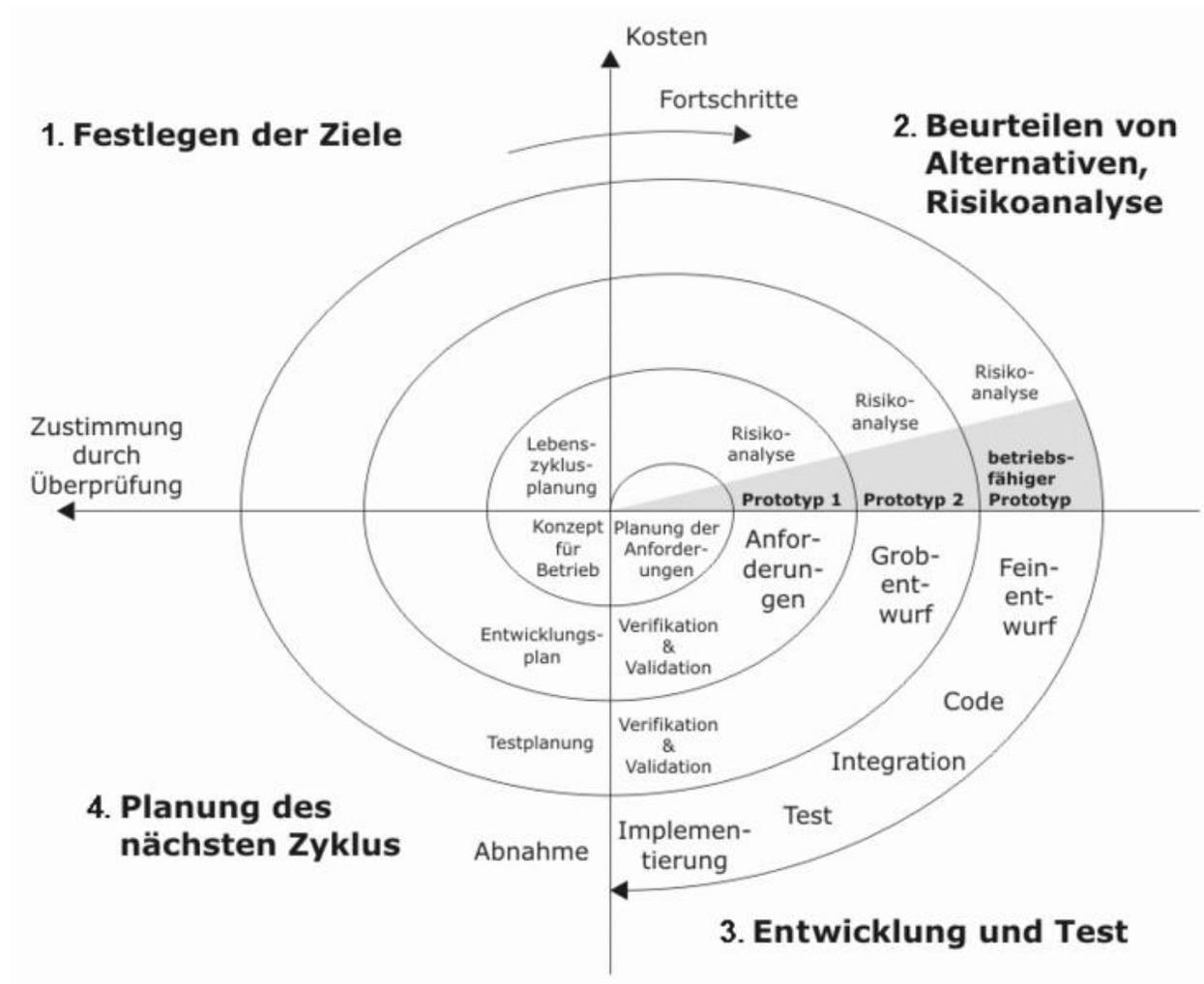


Abbildung 2 Spiralmodell

(Quelle: <http://de.wikipedia.org/wiki/Spiralmodell>)

Wikipedia schreibt über das Spiralmodell (vgl. Abbildung 2):“ Das Spiralmodell ist ein Vorgehensmodell in der Softwareentwicklung, das im Jahr 1988 von Barry W. Boehm beschrieben wurde. Es ist ein generisches Vorgehensmodell und daher offen für bereits existierende Vorgehensmodelle. Das Management kann immer wieder eingreifen, da man sich spiralförmig voran entwickelt.

Das Spiralmodell fasst den Entwicklungsprozess im Software-Engineering als iterativen Prozess auf, wobei jeder Zyklus in den einzelnen Quadranten folgende Aktivitäten enthält:

1. Festlegung von Zielen, Identifikation von Alternativen und Beschreibung von Rahmenbedingungen
2. Evaluierung der Alternativen und Erkennen, Abschätzen und Reduzieren von Risiken
3. Realisierung und Überprüfung des Zwischenprodukts
4. Planung des nächsten Zyklus der Projektfortsetzung.

Am Ende jeder Windung der Spirale steht ein Betrachten des Projektfortschritts (engl. Review). Dabei wird auch der Projektfortgang geplant und verabschiedet.

Das Spiralmodell gehört zu den inkrementellen oder iterativen Vorgehensmodellen. Es ist eine Weiterentwicklung des Wasserfallmodells, in der die Phasen mehrfach spiralförmig durchlaufen werden.

Das inkrementelle und iterative Vorgehensmodell sieht daher eine zyklische Wiederholung der einzelnen Phasen vor. Dabei nähert sich das Projekt langsam den Zielen an, auch wenn sich die Ziele während des Projektfortschritts verändern. Durch das Spiralmodell wird nach Boehm das Risiko eines Scheiterns bei großen Softwareprojekten entscheidend verringert.“

2.2.3 V - Modell

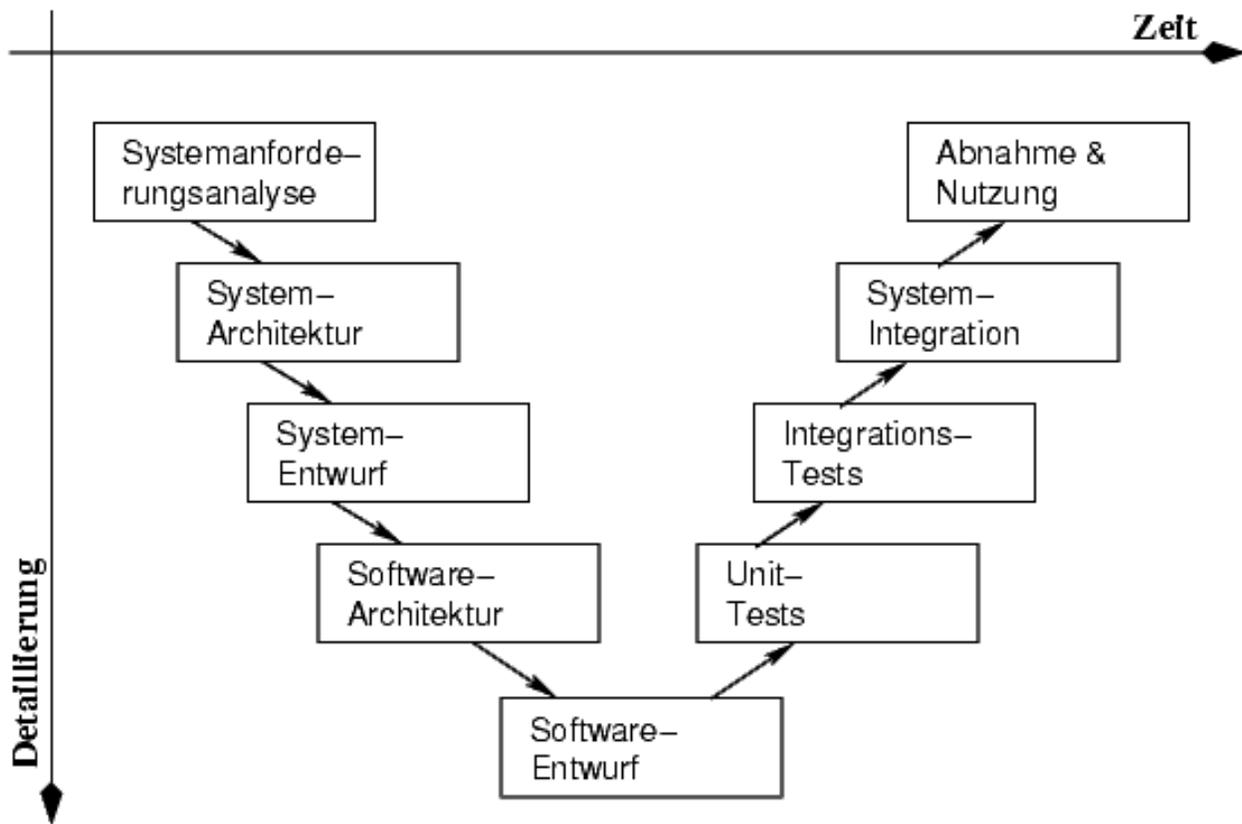


Abbildung 3 V – Modell

(Quelle: <http://de.wikipedia.org/wiki/V-Modell>)

Das V-Modell (vgl. Abbildung 3) ist laut Wikipedia „eine abstrakte, umfassende Projektmanagement-Struktur für die IT-Systementwicklung. Sein Name bezieht sich auf die V-förmige Darstellung der Projektelemente wie IT-Systemdefinitionen und Tests, gegliedert nach ihrer groben zeitlichen Position und ihrer Detailtiefe (siehe Abbildung).

Die Idee zum V-förmigen Vorgehen kam von Barry Boehm 1979. Das erste V-Modell wurde 1986 in Deutschland entwickelt. Zunächst war es für IT-Projekte der öffentlichen Hand vorgesehen, inzwischen wird es aber auch in der Privatwirtschaft eingesetzt.

In der Regel wird eine neue Variante des V-Modells aus der jeweils vorhergehenden Variante entwickelt, sobald ein Verbesserungsbedarf erkannt wird. Allgemeine Eigenschaften dieser Varianten und die Vor- und Nachteile, die mit ihrer Anwendung einhergehen, werden in separaten Artikeln beschrieben, siehe: Vorgehensmodell (Software).

Im Gegensatz zu einem klassischen Phasenmodell werden im V-Modell lediglich Aktivitäten und Ergebnisse definiert und keine strikte zeitliche Abfolge gefordert. Insbesondere fehlen die typischen Abnahmen, die ein Phasenende definieren. Dennoch ist es möglich, die Aktivitäten des V-Modells z.B. auf ein Wasserfallmodell oder ein Spiralmodell abzubilden.“

2.2.4 Eigenschaften

Bei der klassischen Softwareentwicklung ist es wichtig eine umfangreiche und genaue Planung für die jeweiligen Projekte zu erstellen. Die Entwicklung muss nach einem genau vorgeschriebenen Ablaufplan erfolgen. Die Projekte werden ganz genau durchgeplant, mit einer systematischen Zerlegung des Problems, um später keinerlei Änderungen mehr zu benötigen - die Umwelt wird als stabil betrachtet. Der Implementierung geht das Design in jedem Fall voraus. Tests werden erst am Ende des Gesamtprozesses ausgeführt, deshalb können Entwurfsfehler erst spät erkannt werden. Die Herausforderung für die klassische Softwareentwicklung ist laut Bögli (2003) die große zeitliche Differenz zwischen der Analyse der Anforderungen und der effektiven Einführung des Systems, sowie das fehlende kundenseitige Feedback während der Entwicklung. Einige ihrer Methoden führen ihre Tätigkeit nur einmal aus, andere wiederholen sich iterativ, aber alle kennen im Gesamtablauf eine Abfolge von (wenigen) Phasen, wie zum Beispiel Systemdefinition, Anforderung, Konzipierung und Realisierung beim Spiralmodell oder Inception, Elaboration, Construction und Transition bei RUP.

Falls doch Wünsche des Kunden für eine Anpassung oder Veränderung auftreten, können diese als Change-Requests mit monetärem und zeitlichem Aufwand durchgeführt werden. Design ist

hierbei ein repetitiver Prozess, bei dem auf Kundenwünsche erst nach der Fertigstellung des Projektes eingegangen werden kann. Manager dienen dabei als Kontrollinstanz, besonderes Augenmerk wird auf das Erreichen der festgelegten Meilensteine gelegt. Prozesse und Tools stellen einen wichtigen Bestandteil dieses Ansatzes dar, ebenso wie komplexe Dokumentation (Prozesse sind häufig Dokument zentriert). Vertragsverhandlungen haben wesentlich auch mit der Anforderungsanalyse zu tun. Die Anforderungen müssen laut Bögli (2003) so genau erfasst werden, dass sie über die Projektdauer hinweg möglichst stabil bleiben, in den meisten Fällen werden sie auch rechtlich bindend sein. Es ist also im Interesse beider Seiten, den Anforderungen genug Beachtung zu schenken, denn der Kunde erhält ja das erste nutzbare System erst am Ende des Projektes. „Das Ziel der klassischen Methoden ist es, einen Prozess zu definieren, der funktioniert, egal wer ihn ausführt“ (Kauflin 2003, Seite 8).

2.2.5 Änderungskostenkurve Klassische Softwareentwicklung

Die Kostenkurve (vgl. Abbildung 4) von Scott Ambler (2002) visualisiert die Veränderung der Kosten bei einem Release unter Verwendung des Wasserfallmodells. Es werden die relativen monetären Aufwände gezeigt, welche nötig sind um eine Anforderung während des Lebenszyklus zu verändern. Die Kosten steigen exponentiell, je später sie entdeckt werden, da die Artefakte in einem seriellen Prozess, wie bei der klassischen Softwareentwicklung üblich, aufeinander aufbauen.

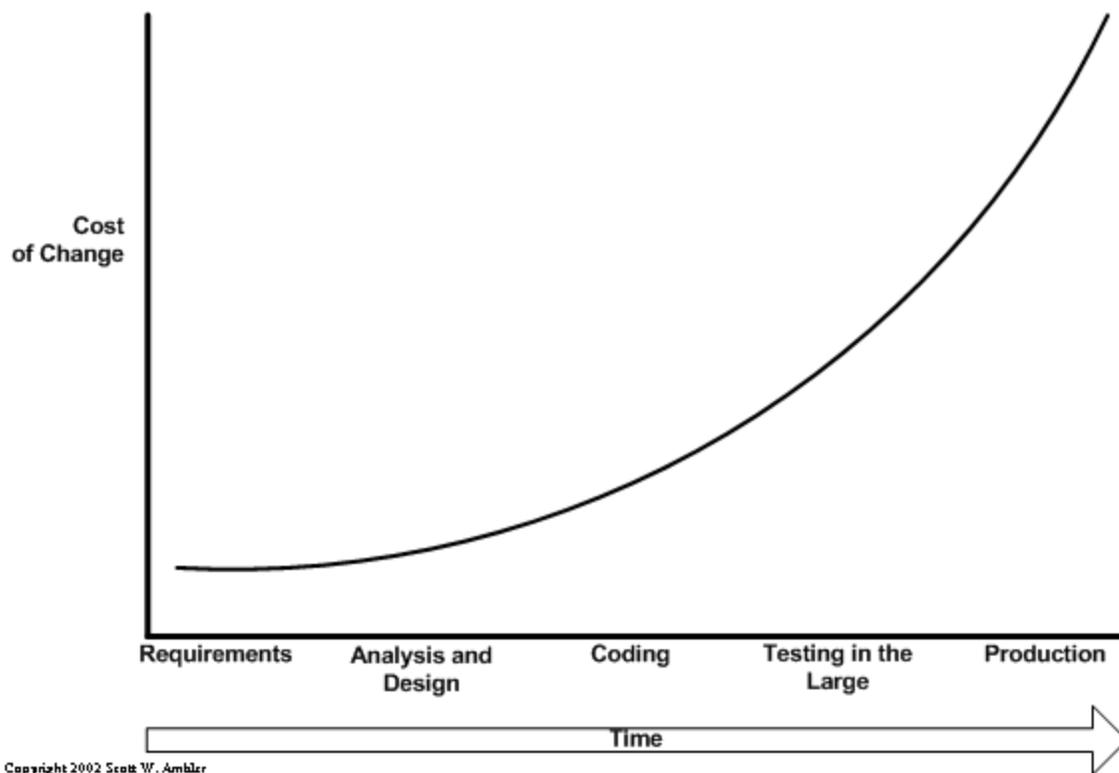


Abbildung 4 Änderungskostenkurve: Klassische Softwareentwicklung

(Quelle: <http://www.agilemodeling.com/essays/costOfChange.htm>)

2.3 Agile Softwareentwicklung

Wolfhart Grote (2008) meint, das oftmalige Scheitern von Softwareentwicklungen führte um die Jahrtausendwende zu Überlegungen, ob nicht der grundsätzliche Ansatz falsch sei. Man suchte die Ursache für das Scheitern nicht bei den handelnden Personen oder bei der unzureichenden Anwendung der Methoden, sondern bei den Methoden selbst. Statt mehr und bessere Regelwerke aufzubauen, versuchte man, mit viel weniger auszukommen. Ziel war es, alle Prozesse flexibel und schnell zu machen und so agil zu werden. Dies zog sich durch alle Bereiche, von der Anforderungsdefinition über die Programmierung bis zum Projektmanagement und der Organisation selbst. Es entstanden sehr unterschiedliche Ansätze mit dem Ziel, diese Agilität zu erreichen.

2.3.1 Das Agile Manifest

„Der Begriff „agil“ wurde maßgeblich von den Autoren geprägt, die zusammen das so genannte „Manifesto for Agile Software Development“³ unterzeichneten“ (vgl. Bögli 2003, Seite 6). Das Manifest der Agile Alliance lautet wie folgt:

We are uncovering better ways of developing software by doing it and helping others to do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

³

<http://www.agilemanifesto.org>

That is, while there is value in the items on the right, we value the items on the left more.

Was bedeutet dieses Manifest (vgl. Kaulin 2003)?

- Individuals and interactions over processes and tools
Die agile Bewegung unterstreicht die Gemeinschaft und Beziehung von den Menschen im Zentrum der Entwicklung und den Softwareentwicklern. Das wirkt sich in engen Teambeziehungen, teamgerechten und angenehmen Arbeitsbedingungen und allem, was den Teamgeist fördert, aus.
- Working software over comprehensive documentation
Das Ziel muss sein, funktionsfähige und getestete Software auszuliefern. Neue Releases werden in relativ kurzen Abständen gefertigt (von stündlich bis monatlich). Entwickler werden angehalten, den Code zielgerichtet, einfach, nach den neuesten technologischen Erkenntnissen zu erstellen und damit die Dokumentation auf ein passables Minimum zu reduzieren.
- Customer collaboration over contract negotiation
Der Zusammenarbeit und Beziehung zwischen Kunden und Entwicklern wird Vorrang gegenüber strikten Verträgen gegeben. Dennoch steigt die Wichtigkeit von gut ausformulierten Verträgen mit der Projektgröße. Der Verhandlungsprozess sollte als Mittel, eine lebensfähige Beziehung zu erreichen und zu erhalten, gesehen werden.
- Responding to change over following a plan
Die Entwicklungsgruppe (Kunden und Entwickler) sollten über den Projektstand gut informiert, kompetent und autorisiert sein, mögliche Änderungen während des Entwicklungsprozesses zu erwägen. Das bedeutet, dass die Teilnehmer auf Änderungen vorbereitet sind und die Verträge diese

Änderungen erlauben.

2.3.2 Eigenschaften

Bei der agilen Softwareentwicklung ist meiner Erfahrung nach die enge Zusammenarbeit mit dem Kunden und das stärkere Einbeziehen der Stakeholder sehr wichtig und ausschlaggebend für den Erfolg, Unsicherheit und Veränderung werden als nicht kontrollierbare Faktoren akzeptiert. Aufgrund des ständigen Kontakts ist man auch flexibel bei Änderungen. Die Phasen (Definition der Anforderungen, Analyse, Implementierung und Test) werden hier mehrmals durchlaufen. Das Design und die Implementation entwickeln sich deshalb gemeinsam, man präsentiert schnelle sichtbare Fortschritte. Agile Ansätze beruhen auf beweglichen Umgebungen, in denen Planstreue (Pläne werden Monate im voraus erstellt) ein schlechtes Maß für Erfolg ist, meint Highsmith (2002).

Laut Cockburn (2002) muss das Team selbstorganisierend sein, Selbstverantwortung und Zurechenbarkeit besitzen, die Verantwortung wird dabei geteilt. Der Kunde bekommt schon nach dem ersten Durchlauf einen lauffähigen Produktteil zur Ansicht. Am Ende jedes dieser Durchläufe (Iterationen) sind Veränderungen und Anpassungen möglich, die Kundenzufriedenheit steht bei der agilen Software Entwicklung im Mittelpunkt. Diese Änderungen werden im so genannten „Backlog“ mitdokumentiert. Diese Liste von Anforderungen ist immer aktuell und priorisiert, sie ersetzt damit die umfassende Dokumentation der bisherigen, klassischen Softwareentwicklung. Aufwände sind bei agilen Methoden also nicht am Projektanfang konzentriert, sondern über die komplette Projektlaufzeit verteilt.

Gegenüber dem klassischen Ansatz gibt es laut Fowler (2005) weniger Planung und Dokumentation. Kurze Releasezyklen sind ebenfalls wie iteratives Vorgehen und ständiges Testen Eckpfeiler der agilen Softwareentwicklung, wodurch sich auch Flexibilität und schnelle Antwortzeiten als Ziel ergeben. Es wird ebenfalls darauf geachtet, dass als Endprodukt ein

funktionierender Programmcode im Vordergrund steht, dadurch ergibt sich auch eine geringere Time-to-Market. Im Praxis Teil dieser Arbeit spielt dies eine wichtige Rolle.

Alle zwei bis drei Monate ist ein Release geplant, und jedes einzelne für sich ist sinnvoll nutzbar. Ein wichtiger Aspekt ist laut Cockburn (2002) die Kommunikation, welche um vieles stärker ausgeprägt ist als bei der klassischen Entwicklung, Individuen und Interaktionen befinden sich im Mittelpunkt, die teaminterne Kommunikation ist wichtiger als Dokumente zu schreiben, deshalb herrscht sehr oft auch räumliche Zusammenarbeit. Es wird versucht eine schnelle Entscheidungsfindung zu praktizieren, unter Einbeziehung von Experten erfolgt inkrementelles Arbeiten und partizipative Entscheidungsfindung. Einzigartige Stärken des Teams werden versucht hervorzuheben, es wird in „Features“ anstatt in Aufgaben gedacht. Der Fokus wird auf Talent, Fähigkeiten und Kommunikation gelegt, was sich auch positiv auf die Motivation der Mitarbeiter auswirkt. Was uns die Praxis hier besonders aufzeigt ist, dass kein Prozess mangelnde Qualifikation ersetzt - seien es nun Soft- oder Hardskills.

Agile Projekte sollten laut Abrahamson, Salo, Ronkainen, Warsta (2002) mit einer sogenannten Explorationsphase begonnen werden, in welcher der Projektraum erkundet wird. Hier kann man die ersten Erfahrungen im Anwendungsbereich sammeln, essentielle Technologien evaluieren (meistens mit Prototypen), erste Ideen für die Handhabung des Systems werden gewonnen und eine grobe Architekturskizze wird erstellt. In der Explorationsphase wird außerdem ermittelt, welche Anwender und Interessensgruppen wie eingebunden werden müssen, und wie die Kunden-/Produktmanagerrolle besetzt werden muss und soll. Des weiteren kann man hier ebenfalls abschätzen, welche Entwicklungsmethode verwendet werden soll. Dies kann zwischen 1 und 3 Monaten dauern. Wenn erforderlich, kann dieser Vorgang auch wiederholt werden.

Cockburn (2002) sieht den Mittelpunkt der agilen Software-Entwicklungsmethoden als eine Nutzung von leichten, aber genügenden Projektverhaltensregeln und Anwendung von menschen- und kommunikationsorientierten Regeln. Der agile Prozess ist genügend wie auch leicht.. Leichtigkeit dieht als Mittel, um beweglich zu bleiben. Genügend ist eine Sache des Im-Spiel-bleibens.

Er listet folgende Punkte für ein erfolgreiches Projekt auf:

- Anwender vor Ort
- zwei bis acht Leute in einem Raum
- erfahrene Entwickler
- kurze inkrementelle Phasen
- voll automatisierte Regressionstests⁴

⁴ Unter einem Regressionstest versteht man in der Softwaretechnik die Wiederholung aller oder einer Teilmenge aller Testfälle, um Nebenwirkungen von Modifikationen in bereits getesteten Teilen der Software aufzuspüren.
(Quelle: <http://de.wikipedia.org/wiki/Regressionstest>)

2.3.3 Kostenkurve Agile Softwareentwicklung

Die Kostenkurve (vgl. Abbildung 5) von Scott Ambler (2002) für die agile Softwareentwicklung steigt langsam mit fortlaufender Dauer eines Projekts.

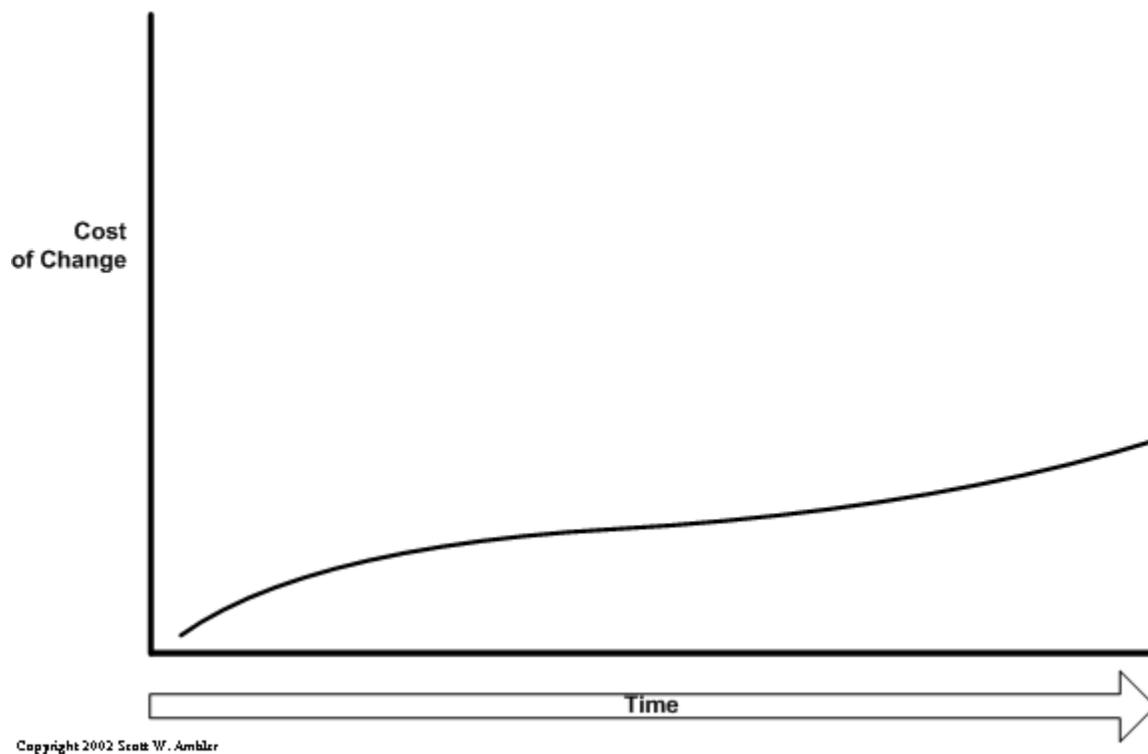


Abbildung 5 Kostenkurve: Agile Softwareentwicklung

(Quelle: <http://www.agilemodeling.com/essays/costOfChange.htm>)

2.3.4 Beispiel Scrum

2.3.4.1 Eigenschaften

Bei der agilen Softwareentwicklung gibt es mehrere unterschiedliche Prozesse welche angewendet werden können, wie zum Beispiel Scrum.

Scrum ist ein leichtgewichtiger Management Prozess, dessen Ideen in den neunziger Jahren entstanden und der 2001 von Ken Schwaber und Mike Beedle in einem Buch veröffentlicht wurde. Die Bedeutung des Wortes „Scrum“ findet laut Kauflin (2003) seinen Ursprung im Rugby; es sei die Strategie, einen Ball wieder ins Spiel zu bringen. Scrum tauchte 1986 zum ersten Mal auf, als ein schneller, adaptiver und selbst organisierender Produktprozess aus Japan von Takeuchi und Nonaka präsentiert wurde. In Abbildung 6 wird die Funktionsweise von Scrum dargestellt, wie der Ablauf bei dieser Methode aussieht.

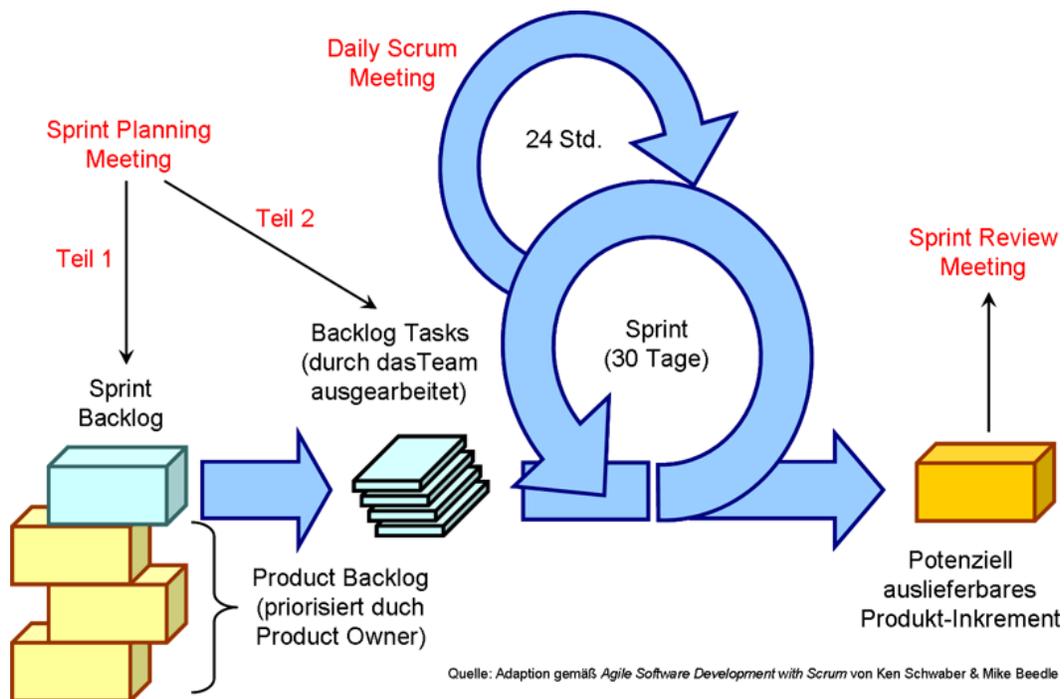


Abbildung 6 Funktionsweise von Scrum

(Quelle: <http://scrum-master.de/content/view/61/31/>)

2.3.4.2 Rollenverteilung in Scrum

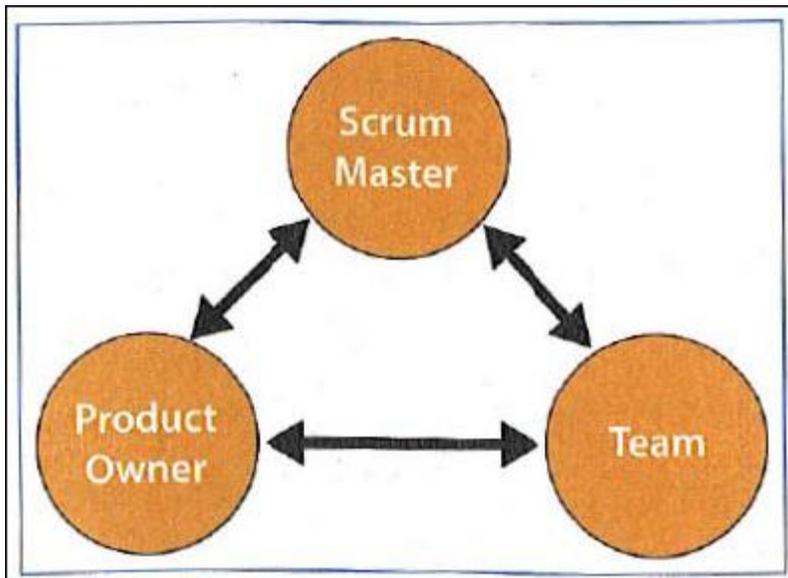


Abbildung 7 Drei unterschiedliche Rollen in Scrum

(Quelle: http://www.infolab.de/presse/pressespiegel/t3n_scrum_2008_06_01.pdf)

Wie in Abbildung 7 gezeigt wird, unterscheidet man hier drei unterschiedliche Rollen.

Folgende Definitionen laut Wikipedia:

2.3.4.2.1 Product Owner

Wikipedia:“Der Product Owner legt das gemeinsame Ziel fest, welches das Team erreichen muss. Er stellt das Budget zur Verfügung. Er setzt regelmäßig die Prioritäten der einzelnen Product Backlog Elemente. Dadurch legt er fest, welche die wichtigsten Features sind, aus denen das Entwicklungsteam eine Auswahl für den nächsten Sprint trifft. Kundenvertreter, Vertrieb und Marketing dürfen ihre Wünsche nur indirekt über den Product Owner einbringen.“

2.3.4.2.2 Team

Wikipedia:“Das Team schätzt die Aufwände der einzelnen Backlog Elemente ab, und beginnt mit der Implementierung der für den nächsten Sprint machbaren Elemente. Dazu wird vor dem Beginn des Sprints ein weiteres Planungstreffen durchgeführt, bei dem die höchst priorisierten Elemente des Backlogs und konkrete Aufgaben aufgeteilt werden. Das Team arbeitet selbstorganisiert im Rahmen einer Time Box (dem Sprint), und hat das Recht (und die Pflicht), selbst zu entscheiden, wie viele Elemente des Backlogs nach dem nächsten Sprint erreicht werden müssen, man spricht dabei von Commitments.“

2.3.4.2.3 Scrum Master

Wikipedia:“Der Scrum Master hat die Aufgabe, die Prozesse der Entwicklung und Planung durchzuführen und die Aufteilung der Rollen und Rechte zu überwachen. Er hält die Transparenz während der gesamten Entwicklung aufrecht, und fördert das Zu-Tage-Treten der bestehenden Verbesserungspotentiale. Er ist keinesfalls für die Kommunikation zwischen Team und Product Owner verantwortlich, da diese direkt miteinander kommunizieren. Er steht dem Team zur Seite, ist aber weder Product Owner noch Teil des Teams. Der Scrum Master sorgt mit allen Mitteln dafür, dass das Team produktiv ist, also die Arbeitsbedingungen stimmen und die Teammitglieder zufrieden sind.“ Der „Scrum Master“ soll wie ein Schäferhund arbeiten. Er hält die Herde (das Team) zusammen und hält die Wölfe (Außenstehende) fern, meint Grote (2008).

2.3.4.3 Artefakte

2.3.4.3.1 Product Backlog

Wikipedia:“Das Product Backlog enthält die Features des zu entwickelnden Produkts. Es beinhaltet alle Funktionalitäten, die der Kunde wünscht, zuzüglich technischer Abhängigkeiten. Vor jedem Sprint werden die Elemente des Product Backlogs neu bewertet und priorisiert, dabei

können bestehende Elemente entfernt sowie neue hinzugefügt werden. Hoch priorisierte Features werden von den Entwicklern im Aufwand geschätzt und in den Sprint Backlog übernommen. Ein wesentliches Merkmal des Backlogs ist die Tiefe der Beschreibung von einzelnen Features. Hoch priorisierte Features werden im Gegensatz zu niedrig priorisierten sehr detailliert beschrieben. Somit wird viel Zeit für die wesentlichen Elemente und wenig für unwesentliche verwendet.“

2.3.4.3.2 *Sprint Backlog*

Wikipedia:“Das Sprint Backlog enthält alle Aufgaben, die notwendig sind, um das Ziel des Sprints zu erfüllen. Eine Aufgabe sollte dabei nicht länger als 16h dauern. Längere Aufgaben sollten in kurze Teilaufgaben zerlegt werden. Bei der Planung des Sprint werden nur so viele Aufgaben eingeplant, wie das Team an Kapazität aufweisen kann.

Die Kapazität berechnet sich gemäß folgender Formel: $\text{Kapazität (in Stunden)} = \text{Arbeitstage} \times \text{Anzahl Personen} \times 7\text{h}$.“

2.3.4.3.3 *Burndown Chart*

Wikipedia:“Der Burndown Chart ist eine täglich aktualisierte graphische Darstellung des noch zu erbringenden Restaufwands für den aktuellen Sprint. Ist der Restaufwand am Ende des Sprints 0, war der Sprint erfolgreich. Die Burndown Chart lässt leicht erkennen, ob dieses Ziel voraussichtlich erreicht werden kann.“

2.3.4.3.4 *Impediment List*

Wikipedia:“In die Impediment List werden alle Hindernisse des Projekts eingetragen. Der Scrum-Master ist dafür zuständig, diese gemeinsam mit dem Team auszuräumen.“

2.3.4.3.5 Sprint

Wikipedia:“Zentrales Element von Scrum ist der Sprint. Ein Sprint bezeichnet die Umsetzung einer Iteration. Scrum schlägt ca. 30 Tage als Iterationslänge vor, typischerweise ist die Dauer 2 oder 4 Wochen. Vor dem Sprint werden die Produkthanforderungen des Kunden in einem Product Backlog gesammelt. Auch technische und administrative Aufgaben werden dort aufgenommen. Das Product Backlog muss nicht vollständig sein - es wird laufend fortgeführt.

Am Beginn eines jeden Sprints steht ein Sprint Planning Meeting - hier wird die Arbeit für den Sprint geplant, wo die am höchsten priorisierten Anforderungen aus dem Backlog in Angriff genommen werden. Das Team einigt sich auf die zu erreichende Funktionalität bis Ende des Sprints. Während des laufenden Sprints ist das Team gegen Änderungen von außen geschützt.

Die Anforderungen für den ersten Sprint sind meistens rasch aufgestellt. Die Anforderungen werden informell skizziert. Für einen Sprint wird ein Sprint Backlog erstellt, in dem Anforderungen übernommen werden, die während des Sprints umzusetzen sind. Die Entscheidung, welche Anforderungen umgesetzt werden, wird vom Kunden nach von ihm festgelegten Prioritäten getroffen. Zum Sprint organisiert sich das Entwicklungsteam selbst und braucht daher keine detaillierten methodischen Vorschriften.“

2.3.4.3.6 Daily Scrum

Wikipedia:“An jedem Tag findet ein kurzes (maximal 15-minütiges) Daily Scrum statt.

Scrum definiert keine konkrete Uhrzeit für das Meeting, außer dass der tägliche Termin dafür fix ist. Empfohlener Zeitpunkt für das Scrum-Meeting ist die Zeit nach dem Mittagessen, da morgendliche Scrum-Meetings oft mit flexiblen Gleitzeitregelungen kollidieren und der müde Punkt nach dem Mittagessen bei einem Scrum-Meeting, welches durchaus im Stehen abgehalten werden kann, nicht so stark ins Gewicht fällt wie bei anderen Tätigkeiten. Die Meetings sind kürzer als am Morgen, weil allgemeine Dinge und Neuigkeiten schon vorher diskutiert wurden und die Mitarbeiter mit dem Kopf ganz bei der Arbeit sind.

Das Team stellt sich gegenseitig die folgenden Fragen:

- "Bist du gestern mit dem fertig geworden, was du dir vorgenommen hast?"
- "Welche Aufgaben wirst du bis zum nächsten Meeting bearbeiten?"
- "Gibt es ein Problem, das dich blockiert?"

Die Sitzung dient dem Informationsaustausch des Teams untereinander. Hier geht es darum, dass möglichst alle alles wissen, die Teammitglieder sich synchronisieren und erkennen, wo Probleme auftreten könnten. Falls neue Hindernisse erkannt werden, sind diese vom Scrum Master zu bearbeiten. Dazu werden sie in das Impediment Backlog eingetragen.

Größere Projekte werden durch das Einführen von Scrum-of-Scrum Meetings, Product Owner Daily Scrums und Scrum Master Weekly gesteuert.“

2.3.4.3.7 Review

Wikipedia:“Nach einem Sprint wird das Sprint-Ergebnis einem informellen Review durch Team und Kunden unterzogen (Sprint Review Meeting). Dazu wird das Ergebnis des Sprints (die laufende Software) vorgeführt, eventuell werden technische Eigenschaften präsentiert.“ Die Arbeitsergebnisse werden laut Schwaber (2004) vom Product Owner als Vertretung des Kunden überprüft, Änderungen im Product Backlog dokumentiert. Ein weiterer Vorteil die Position des Product Owner einzusetzen wäre, im Falle mehrerer Kunden welche an einem Projekt beteiligt sind, aus unterschiedlichen Kanälen Anforderungen aufzunehmen, zu konsolidieren und dann die Priorisierung vorzunehmen.

2.3.4.3.8 Retrospektive

Wikipedia:“In der Retrospektive wird die zurückliegende Sprint-Phase betrachtet. Es handelt sich dabei nicht um Lessons Learned, sondern um einen zunächst wertfreien Rückblick auf die Ereignisse des Sprints. Alle Teilnehmer notieren dazu die für sie wichtigen Ereignisse auf Zetteln und ordnen sie dem Zeitstrahl des Sprints zu. Anschließend schreiben die Teilnehmer alle Punkte auf, welche ihnen zu den Fragen "Was war gut?" (Best practice) bzw. "Was könnte

verbessert werden?" (Verbesserungspotential) einfallen. Jedes Verbesserungspotential wird priorisiert und einem Verantwortungsbereich (Team oder Organisation) zugeordnet. Alle der Organisation zugeordneten Themen werden vom Scrum Master aufgenommen und in das Impediment Backlog eingetragen. Alle teambezogenen Punkte werden in das Product Backlog aufgenommen. Der Product Owner kann am Beginn des nächsten Sprints mit Hilfe neuer Priorisierung des Backlogs auf mögliche Änderungen reagieren und gegebenenfalls die Ziele für den weiteren Vorgang adaptieren.

Wird für die Retrospektiven und deren Vorbereitung nicht genug Zeit eingeräumt, bleiben die Erkenntnisse und Ergebnisse oberflächlich und die Resultate nach jedem Sprint ähneln sich. Dann läuft man Gefahr, dass die Retrospektiven an Stellenwert verlieren oder ganz gestrichen werden, weil die Ergebnisse vorhersehbar sind.“

2.3.4.4 Kritische Betrachtung

Wikipedia:“Auch das Vorgehensmodell Scrum kann scheitern, wenn es unausgewogen eingesetzt wird. Ein besonderes Risiko sind dominante Teamplayer, die den Prozess der Selbstorganisation stören, ohne einen gleichwertigen eigenen Beitrag in diesen Organisationsprozess und die Problemlösung einzubringen.“

2.4 Agile gegenüber klassische Softwareentwicklung

Bleek und Floyd (2004) meinten unter anderem zu diesem Thema:

2.4.1 Zeiträume

Klassische Methoden verarbeiten große Mengen an Aufgaben, was sich über mehrere Tage und Wochen erstrecken kann.

Agile Methoden versuchen über einen genau gegliederten, täglich wiederkehrenden Prozess dem Entwicklungsteam die Arbeit zu erleichtern. Die Aufgaben werden nie alleine gelöst und keines der anfallenden Probleme soll größer sein, als dass man es an einem Tag erledigen könnte.

2.4.2 Dokumente

Klassische Methoden benutzen Dokumente, Anforderungsdefinitionen, Spezifikationen, Szenarien, Glossar, Entwurfs- und Interaktionsdiagramme, Code usw.) zur Vergegenständlichung ihres gemeinsam erarbeiteten Wissens.

Agile Methoden verzichten weitgehend auf Dokumente, solange sich die wesentlichen Aussagen auch im Code festhalten lassen.

2.4.3 Modellierung

Klassische Methoden haben als Grundannahme, dass große Teile des zukünftigen Systems zuerst erschlossen und dann ein Kernsystem detailliert modelliert wird.

Agile Methoden implementieren zuerst das kleinste zu identifizierende Problem, ohne jedoch das Problemfeld vollständig erschlossen zu haben. Je nach Entwicklungspfad wird dann weiter exploriert bzw. Bestehendes umgebaut.

2.4.4 Personelle Projektausstattung

Klassische Methoden bilden für große Probleme große Teams (>10) und beschäftigen Spezialisten. Koordiniert wird die Zusammenarbeit über Meetings auf verschiedenen Ebenen einer eingestufteten Hierarchie.

Agile Methoden bevorzugen kleine Projektteams und vermeiden Spezialisten durch Collective Code Ownership. Es gibt eine flache Hierarchie. Koordiniert wird über die tägliche Planung.

2.4.5 Qualifikationen

Klassische Methoden besitzen die Annahme, dass jede Person ihrer Qualifikation nach eingesetzt werden muss. Je nach Expertise werden die Aufgaben ausgewählt.

Agile Methoden gehen davon aus, dass es nur hochqualifizierte Mitarbeiter gibt. Fast alle können alle Aufgabenstellungen erfüllen und erledigen dies auch im Laufe eines Projektes.

2.4.6 Ergebnis

Klassische Methoden bieten erst nach dem Ende des Softwareentwicklungsprozesses ein benutzbares Ergebnis.

Agile Methoden stellen dem Kunden schon nach dem ersten Durchlauf ein funktionstüchtiges Programm zur Lösung eines Teilproblems zur Verfügung.

2.4.7 Zielgrößen

2.4.7.1 Klassische Methoden

Klassische Methoden stellen laut Grote (2008) im Allgemeinen den Funktionsumfang als Zielgröße nicht in Frage, sondern machen eher Abstriche bei Qualität und Gewinn, um Vertragsstrafen bei Nicht-Einhaltung von Terminen zu umgehen. Damit bei allen Zielgrößen ein großer Wert auch immer ein gutes Ergebnis bedeutet, wird die Zielgröße „Kosten“ durch deren Inversion, nämlich „Gewinn“ ersetzt, im Idealfall soll die Zielgröße 100% erfüllen. In der

Realität kommt es jedoch oft zu Engpässen, es werden Kompromisse eingegangen und somit auf die Vollerfüllung einzelner Zielgrößen verzichtet.

Eine Möglichkeit bietet das Einsparen an der Qualität des Produktes (vgl. Abbildung 8). Die unvollständige Erfüllung ist zunächst nicht offensichtlich. Aufgrund von möglichen Vertragsstrafen werden Funktionsumfang und Termintreue eingehalten, ebenso wie die Forderung seitens des Managements die geplanten Kosten nicht zu übersteigen, um den geplanten Gewinn zu erreichen. Natürlich können in Folge auch Nachbesserungen vonnöten sein, was erneut Kosten verursacht und den Gewinn schmälert (vgl. Abbildung 9). Eine weitere Möglichkeit sind Prestigeprojekte, mit denen man sich präsentieren möchte, jedoch der Gewinn gegen Null tendiert (vgl. Abbildung 10).

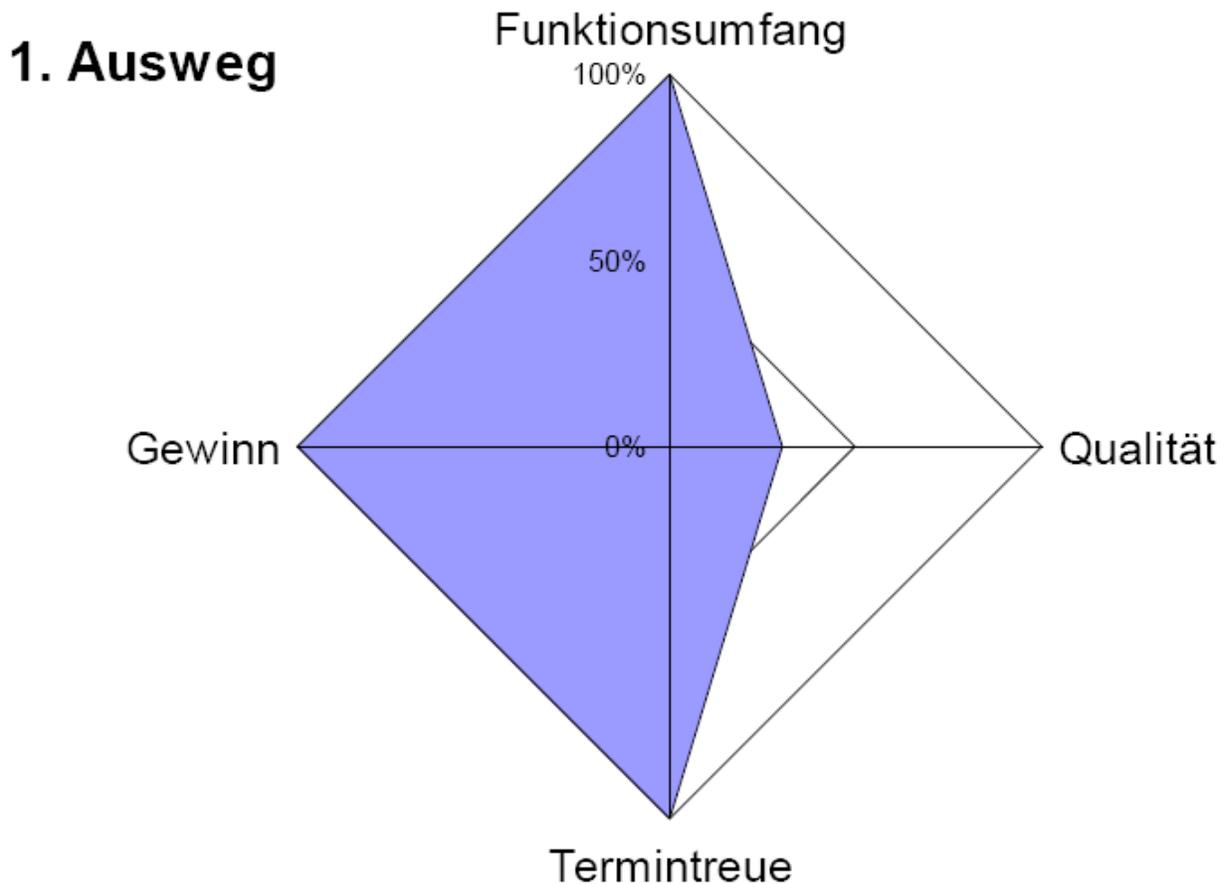


Abbildung 8 Sparen an der Qualität

(Quelle: http://www.infolab.de/presse/pressespiegel/t3n_scrum_2008_06_01.pdf)

2. Ausweg

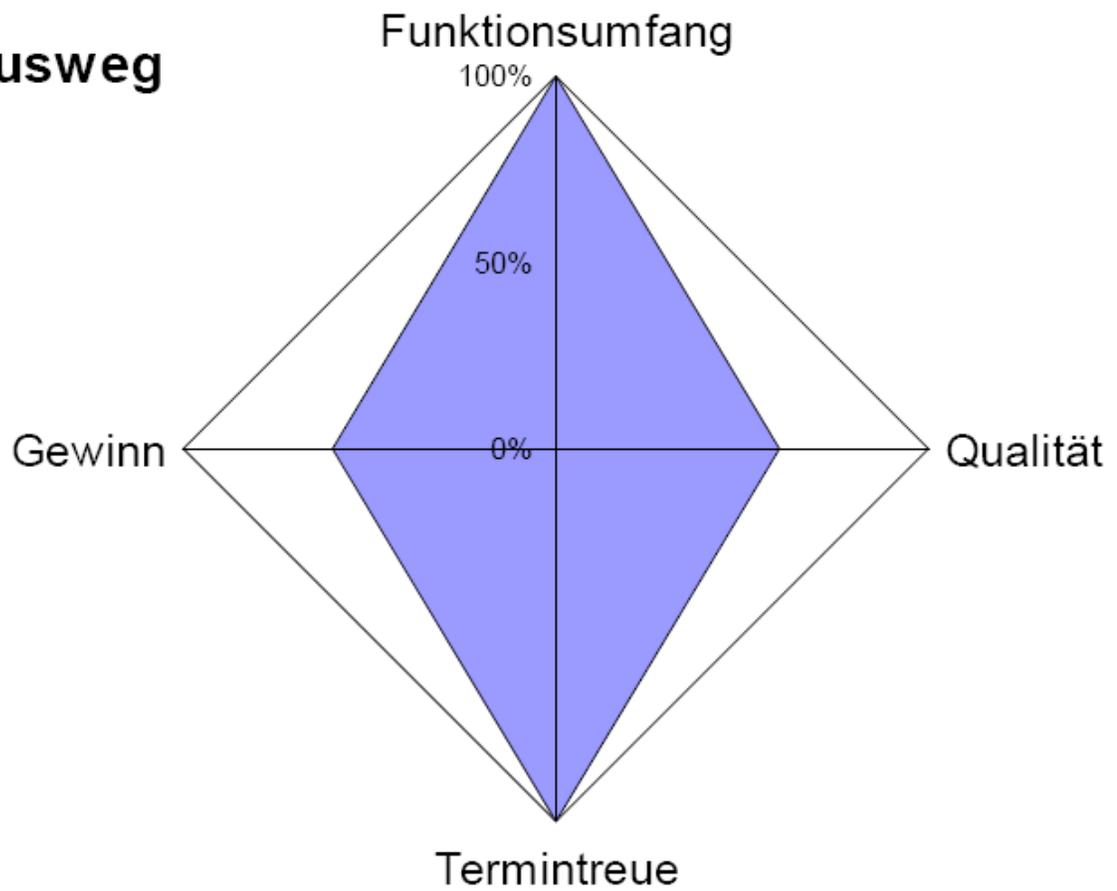


Abbildung 9 Nachbesserungen schmälern den Gewinn

(Quelle: http://www.infolab.de/presse/pressespiegel/t3n_scrum_2008_06_01.pdf)

Bei einem Prestigeprojekt (vgl. Abbildung 10) wird versucht die Qualität gegen 100 % gehen zu lassen, aufgrund der benötigten Aufwände/Nachbesserungen geht der Gewinn gegen Null.

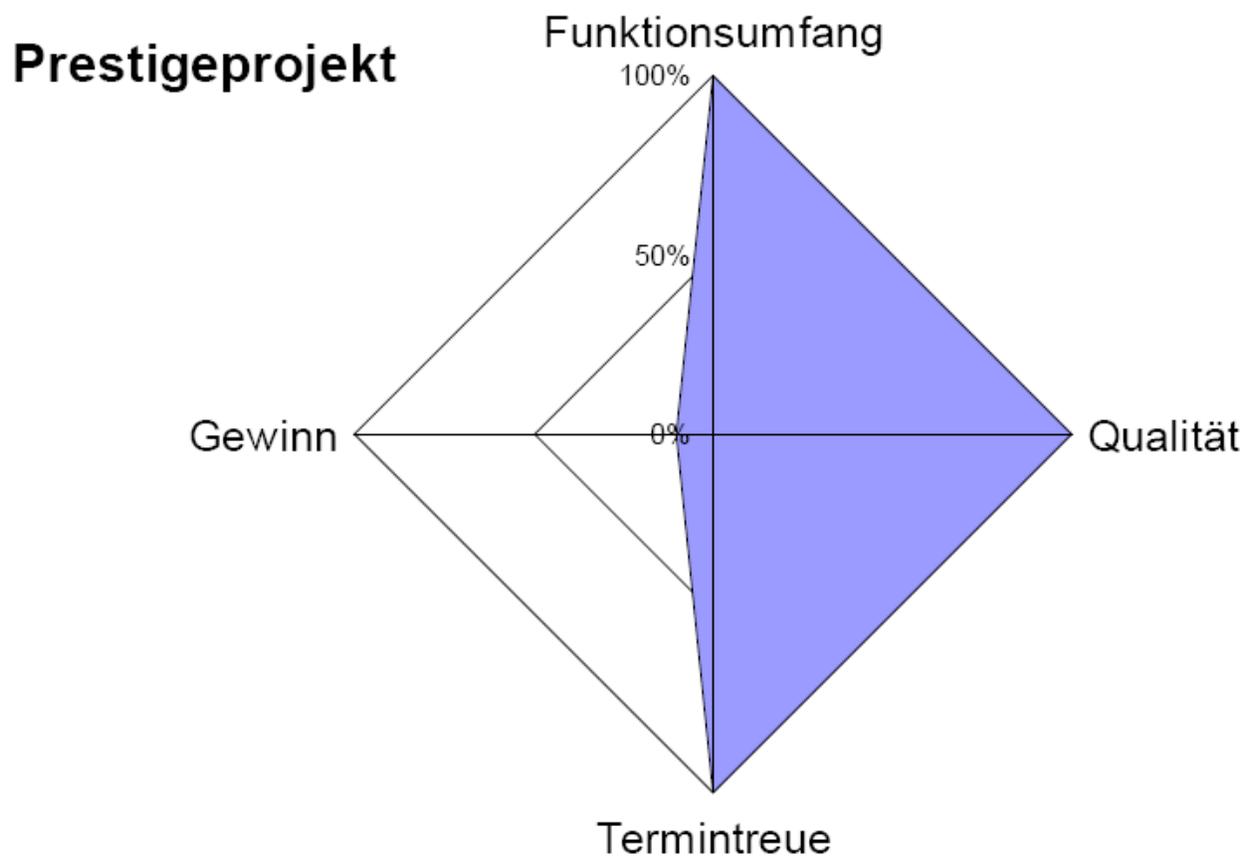


Abbildung 10 Prestigeprojekt

(Quelle: http://www.infolab.de/presse/pressespiegel/t3n_scrum_2008_06_01.pdf)

2.4.7.2 Agile Methoden

Bei agilen Methoden wird laut Grote (2008) eher versucht den Funktionsumfang einzuschränken, um die anderen Zielgrößen bei 100 Prozent zu belassen. Das Projekt wird in der vertraglich festgelegten Qualität und den festgelegten Kosten termingerecht fertiggestellt. Der Funktionsumfang kann möglicherweise etwas verringert sein, der Kunde hat jedoch bei Scrum die Möglichkeit über die Priorisierung der einzelnen Funktionen im Product Backlog zu entscheiden (vgl. Abbildung 11).

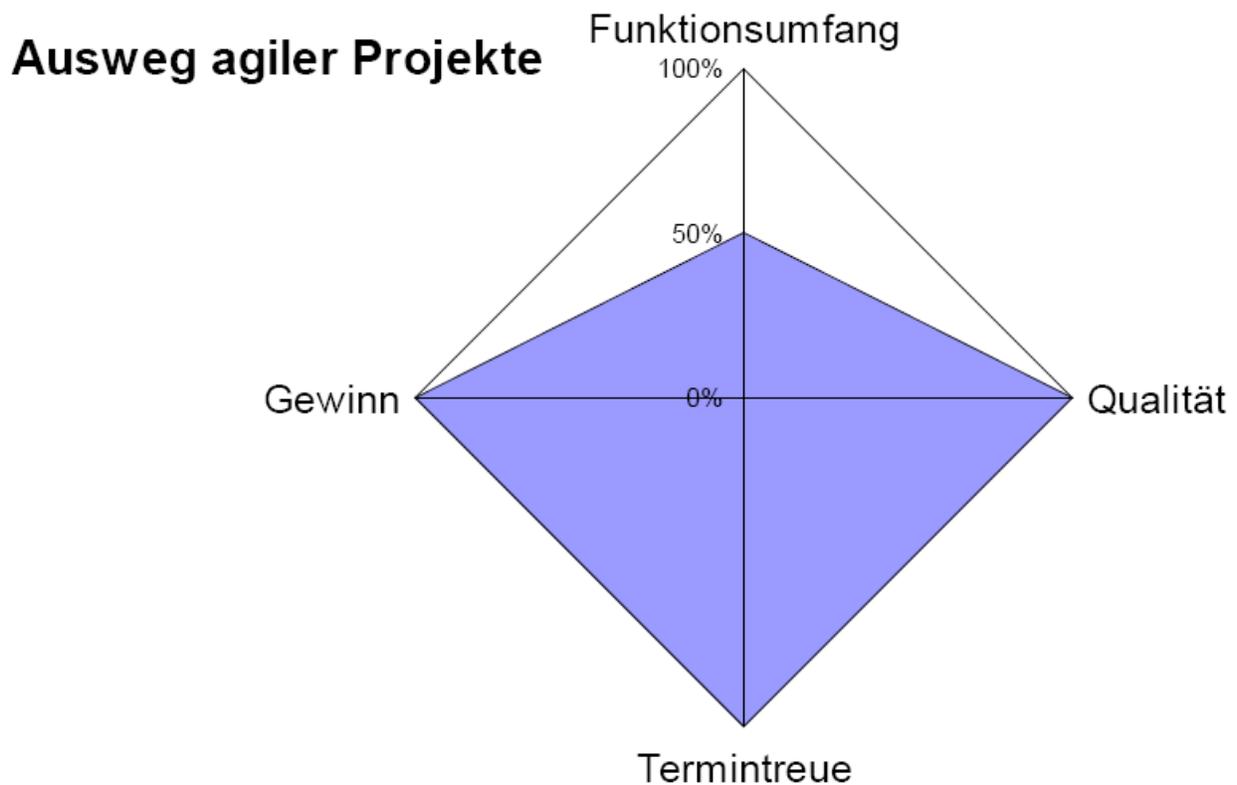


Abbildung 11 Ausweg agiler Projekte

2.4.7.3 Fazit

Die Entscheidung, welche Art von Entwicklungsmethode für ein bestimmtes Projekt zu wählen ist, ist sehr wesentlich von der Planbarkeit bestimmt, welche nun selbst wieder von der Stabilität der Anforderungen abhängt. Fowler (2003) schreibt dazu: „Everything else in software development depends on the requirements. If you cannot get stable requirements you cannot get

a predictable plan.” Dies bedeutet also, dass im Fall von stabilen Anforderungen bevorzugt eine klassische Methode angewendet werden soll, hingegen bei sich ständig ändernden Anforderungen eine agile empfehlenswert wäre.

Hier möchte ich aber noch erwähnen dass die Praxis mir gezeigt hat dass auch bei der klassischen Methode Anforderungen nachträglich geändert werden, der Unified Process zum Beispiel berücksichtigt das Aufnehmen von weiteren Anforderungen.

3 Prozesse in der Praxis

3.1 Der CVT Prozess

Dieser Prozess beschreibt die Interaktion mit diversen Ansprechpartnern eines europäischen Kunden.

Zunächst möchte ich erklären dass unser Kunde ein Mobilnetzbetreiber ist und mit dem Begriff „Kunde“, welcher im weiteren Verlauf verwendet wird, dieses Unternehmen gemeint ist, nicht jedoch der Endbenutzer der seinerseits wieder Kunde des Mobilnetzbetreibers ist (vgl. Abbildung 12).

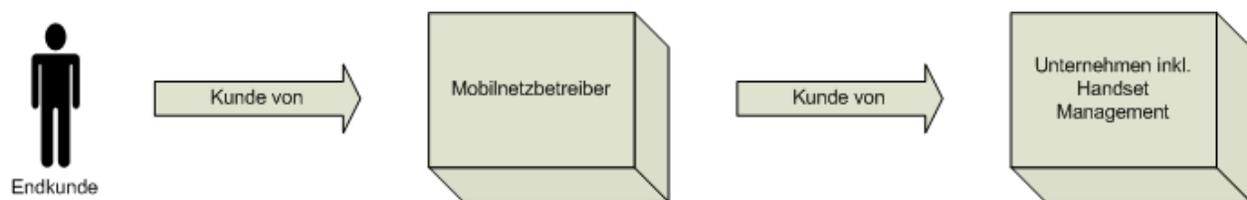


Abbildung 12 Beziehung zwischen Endkunde – Mobilnetzbetreiber – Unternehmen inkl. Handset Management

CVT steht für Content Verification Testing, verschiedene Medien Typen werden auf ihre Funktionalität mit einem Handset getestet. Dies wird für den Kunden, dem Mobilnetzbetreiber durchgeführt, damit alle in den Markt kommenden Handsets des Netzbetreibers auch mit der Applikation unseres Unternehmens funktionieren, und der Endkunde dies problemlos benutzen kann. Hierfür ist das Handset Management zuständig. Wichtig ist hierbei die Funktionalität gegenüber der Medientypen/Software zu überprüfen und falls es im Rahmen des Möglichen liegt (Adaption der Software) dies zu gewährleisten. Da es sich hier um einen schnellebigen Markt handelt, ist jedes Handset als zeitkritisch zu behandeln, deshalb schnelle Durchlaufzeiten sehr erwünscht.

Im weiteren wird auch der Begriff des Tracking Tools verwendet, hierbei handelt es sich um eine Web-basierende Drittfirmensoftware, welche das Speichern von Kommentaren und Dokumenten, versehen mit einem Zeitstempel, ermöglicht. Hier besteht die Möglichkeit Unternehmensintern, als auch mit dem Kunden zu kommunizieren. Einträge können nur für zuständige Personen intern sichtbar gemacht werden, aber auch für den Kunden. So kann man auch interne Diskussionen führen, ohne dass der Kunde hierbei involviert werden soll. Zugang zu den Tracking Tools haben alle Mitarbeiter die hier beteiligt sind, quer über die Organisationseinheiten Technical Operations, Development, Client Management, Content Services, Quality Services, Service Management, Account Management und Handset Management (vgl. Abbildung 13).

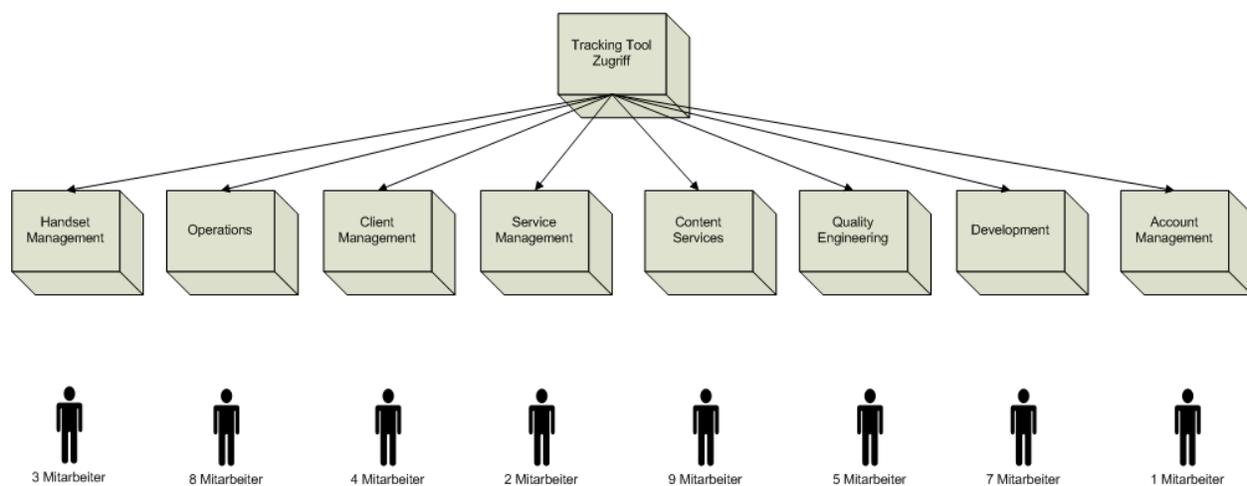


Abbildung 13 Organisationseinheiten mit Tracking Tool Zugang

Das Handset Management Team ist in keinem Scrum Team vertreten, jedoch bei der Definition der Anforderungen und beim Testen neuer Komponenten integriert. Auch werden unserem Handset Management Team die neuen, fertigen Softwareteile vom Client Management Team präsentiert.

Die folgenden Prozessabbildungen wurden im Zuge Diplomarbeit erstellt.

3.1.1 Prozessabbildung

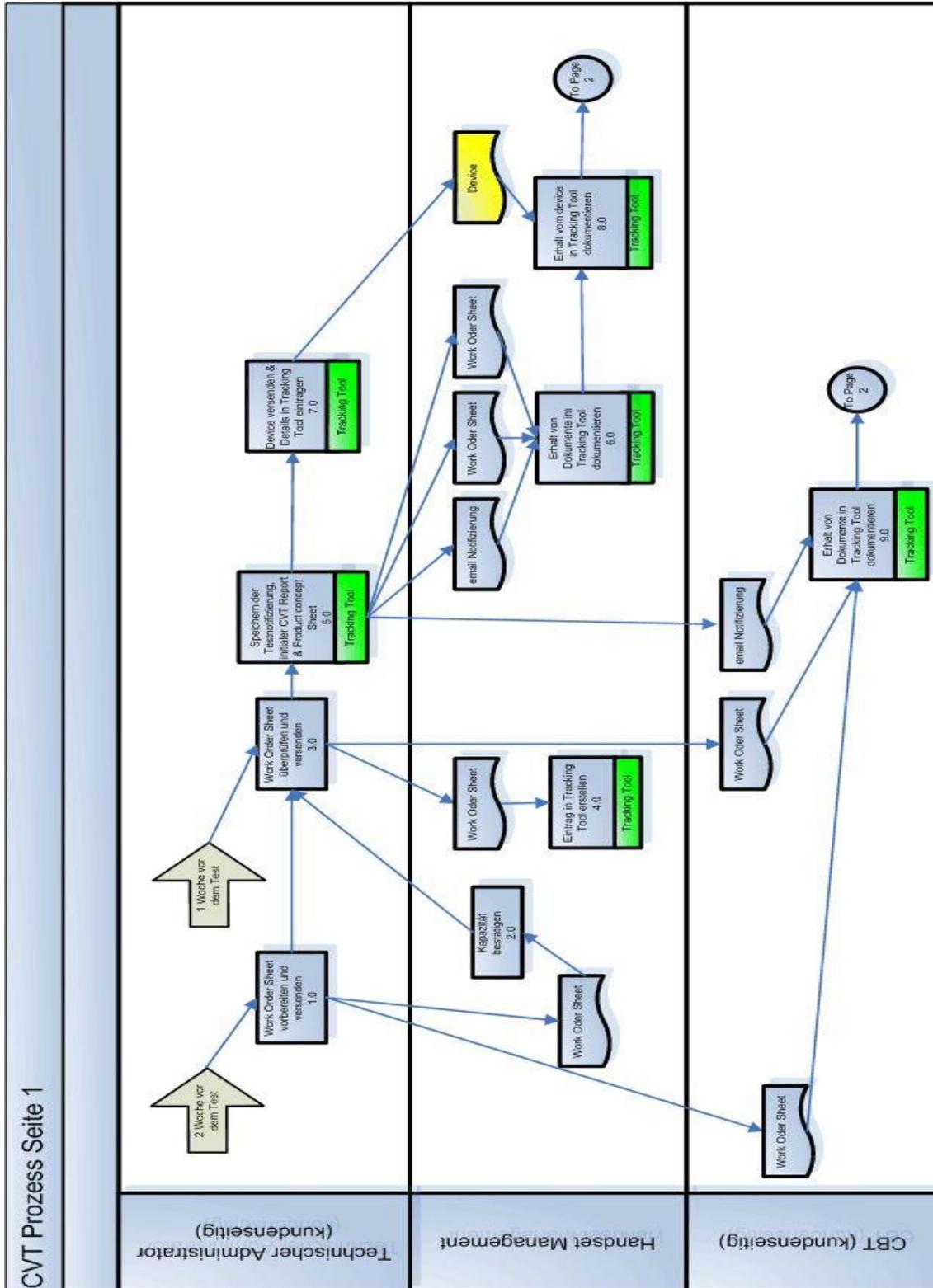


Abbildung 14 CVT Prozess Seite 1

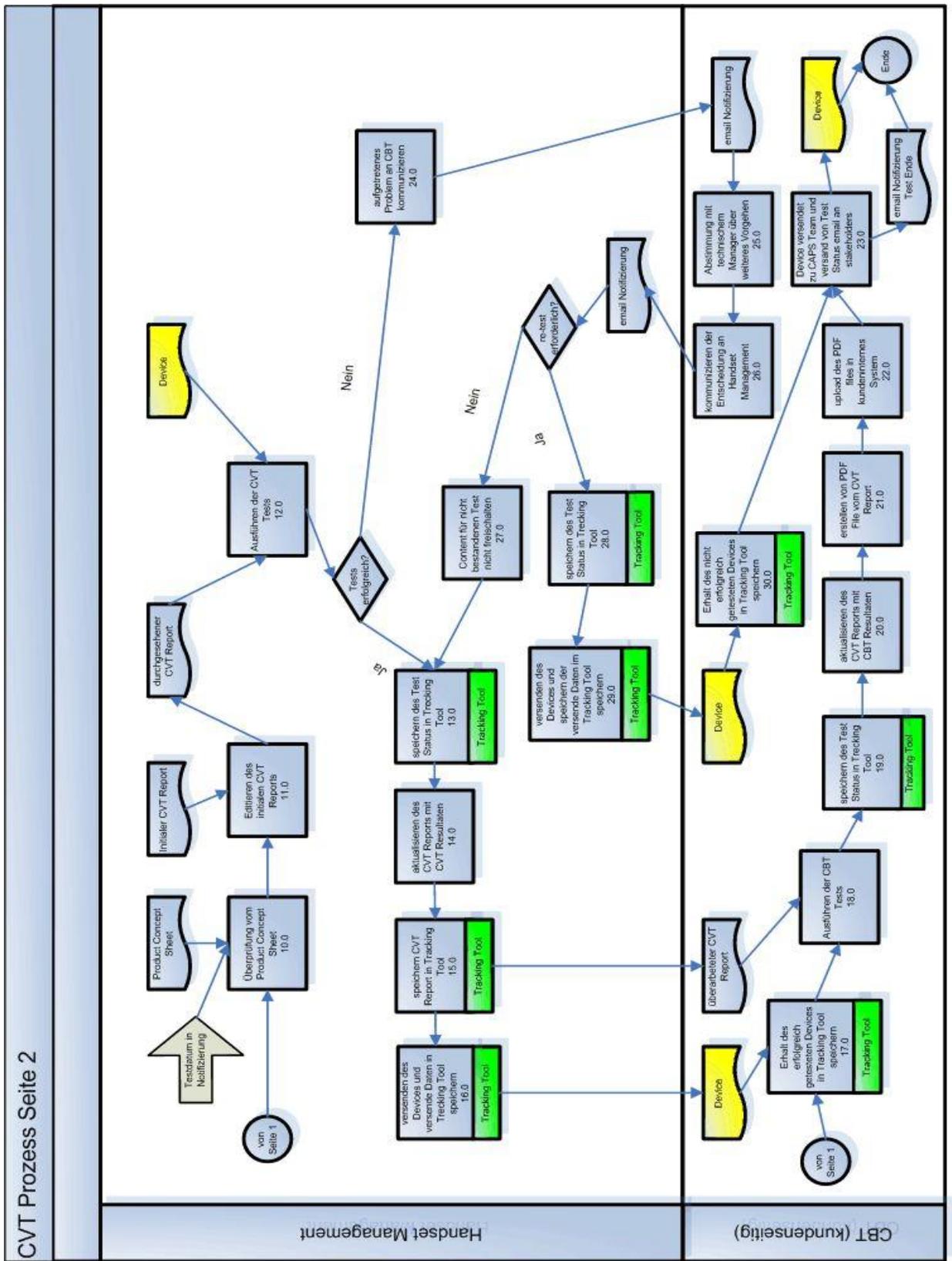


Abbildung 15 CVT Prozess Seite 2

3.1.2 Prozessbeschreibung

3.1.2.1 Abläufe im Vorfeld des Tests

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
1.0	Work Order Sheet vorbereiten und versenden	Work Order Sheet vorbereiten und an beteiligte Stellen senden	Technischer Administrator (kundenseitig)	Work Order Sheet muss vorbereitet und an Handset Management und CBT (kundenseitig) gesendet werden	2 Wochen vor dem Versand des Mobiltelefons
2.0	Kapazität bestätigen	Kapazität wird bestätigt	Handset Management	Es werden die freien Kapazitäten des Handset Managements bestätigt	2 Wochen vor dem Versand des Mobiltelefons
3.0	Work Order Sheet überprüfen und versenden	Work Order Sheet überprüfen und an beteiligten Stellen senden	Technischer Administrator (kundenseitig)	Work Order Sheet muss überprüft und an Handset Management und CBT (kundenseitig) gesendet werden	1 Woche vor dem Versand des Mobiltelefons
4.0	Eintrag in Tracking Tool erstellen	Den Erhalt im Tracking Tool dokumentieren	Handset Management	Erhalt des Work Order Sheets im Tracking Tool mit Datum und Uhrzeit speichern	1 Woche vor dem Versand des Mobiltelefons
5.0	Speichern der Testnotifizierung, initialer CVT Report & Product Concept	Das Speichern der Testnotifizierung, initialer CVT Report &	Technischer Administrator (kundenseitig)	Testnotifizierung, initialer CVT Report & Product Concept Sheet werden in das Tracking Tool gespeichert	1 Woche vor dem Versand des Mobiltelefons

	Sheet	Product Concept Sheet			
6.0	Erhalt von Dokumenten im Tracking Tool dokumentieren	Sicherstellung das Dokumente im Tracking Tool gespeichert wurden	Handset Management	Im Tracking Tool mit Datum und Uhrzeit den Erhalt der Testnotifizierung, initialer CVT & Product Concept Sheet speichern	1 Woche vor dem Versand des Mobiltelefons
7.0	Device versenden und Details im Tracking Tool eintragen	Device wird versendet und Details im Tracking Tool eingetragen	Technischer Administrator (kundenseitig)	Das Device wird an Handset Management versendet, Details in das Tracking Tool gespeichert	Bis eine Woche vor Beginn der Tests
8.0	Erhalt vom Device im Tracking Tool dokumentieren	Den Erhalt des Devices im Tracking Tool speichern	Handset Management	Im Tracking Tool das Datum und die Uhrzeit vom Erhalt des Devices speichern.	Bis eine Woche vor Beginn der Tests
9.0	Erhalt von Dokumenten im Tracking Tool dokumentieren	Den Erhalt des Work Order Sheets und der E-Mail Notifizierung im Tracking Tool speichern	CBT (kundenseitig)	Im Tracking Tool mit Datum und Uhrzeit den Erhalt des Work Oder Sheets und der E-Mail Notifizierung speichern	1 Woche vor dem Versand des Mobiltelefons

3.1.2.2 Abläufe während der Tests

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
10.0	Überprüfung des Product Concept Sheets	Product Concept Sheet wird überprüft	Handset Management	Product Concept Sheet wird überprüft und verstanden	Vor Beginn der Tests
11.0	Editieren des initialen CVT Reports	Initialer CVT Report wird editiert	Handset Management	Initialer CVT Report für die Tests editieren	Vor Beginn der Tests
12.0	Ausführen der CVT Tests	Die CVT Tests werden durchgeführt und Ergebnisse im CVT Report gespeichert	Handset Management	Im Product Concept Sheet angegebene Medien Typen testen, die Ergebnisse in CVT Report vermerken	
13.0	Speichern des Teststatus im Tracking Tool	Der Teststatus wird im Tracking Tool gespeichert	Handset Management	Im Tracking Tool das Datum, die Uhrzeit und Status der Tests speichern	Nach Beendigung des CVT Tests
14.0	Aktualisieren des CVT Reports mit CVT Resultaten	Der CVT Report wird mit den Testdaten aktualisiert	Handset Management	Die Testdaten werden in den CVT Report eingetragen, der Report wird aktualisiert	Nach Beendigung des CVT Tests
15.0	Speichern CVT Report im Tracking Tool	Der CVT Report wird im Tracking Tool gespeichert	Handset Management	Der aktuelle CVT Report wird im Tracking Tool gespeichert	Nach Beendigung des CVT Tests

16.0	Versenden des Devices und Versanddaten im Tracking Tool speichern	Das Device wird versendet und die Versanddaten werden im Tracking Tool gespeichert	Handset Management	Das Device wird an den Kunden zurückgesendet, die Versanddaten wie Datum, Uhrzeit und Tracking Number werden im Tracking Tool gespeichert	Nach Beendigung des CVT Tests
17.0	Erhalt des erfolgreich getesteten Devices im Tracking Tool speichern	Der Erhalt des erfolgreich getesteten Devices wird im Tracking tool gespeichert	CBT (kundenseitig)	Im Tracking Tool das Datum und die Uhrzeit vom Erhalt des Devices speichern.	Nach Beendigung des CVT Tests
18.0	Ausführen der CBT Tests	CBT Tests werden ausgeführt	CBT (kundenseitig)	CBT Tests werden kundenseitig ausgeführt	Nach Beendigung des CVT Tests
19.0	Speichern des Teststatus in Tracking Tool	Der Test Status wird im Tracking Tool gespeichert	CBT (kundenseitig)	Die CBT Tests werden im Tracking Tool gespeichert	Nach Beendigung des CVT & CBT Tests
20.0	Aktualisieren des CVT Reports mit CBT Resultaten	Der CVT Report wird mit den CBT Daten aktualisiert	CBT (kundenseitig)	Die Testdaten werden in den CVT Report eingetragen, der Report aktualisiert	Nach Beendigung des CVT & CBT Tests
21.0	Erstellen von PDF File vom CVT Report	Aus dem CVT Report wird ein PDF File generiert	CBT (kundenseitig)	Ein PDF File wird aus dem CVT Report generiert	Nach Beendigung des CVT & CBT Tests
22.0	Upload des PDF Files in	Das PDF File wird in ein	CBT (kundenseitig)	Das PDF File wird in ein kundeninternes	Nach Beendigung des CVT & CBT

	kundeninternes System	kundeninternes System geladen		System geladen	Tests
23.0	Device versendet an CAPS Team und Versand des Teststatus an Stakeholder	Das Device wird an das CAPS Team versendet, der Teststatus an die Stakeholder	CBT (kundenseitig)	Das kundenseitige CBT Team sendet das Device an das ebenfalls kundenseitige CAPS Team, der Test Status wird an alle Stakeholder gesendet	Nach Beendigung des CVT & CBT Tests
24.0	Aufgetretenes Problem an CBT kommunizieren	Das aufgetretene Problem wird an CBT kommuniziert	Handset Management	Die aufgetretenen Probleme werden via E-Mail an das kundenseitige CBT Team mit Datum, Uhrzeit und Fehlerbeschreibung gesendet	Gleicher Tag wie die Probleme auftreten
25.0	Abstimmung mit technischem Manager über weiteres Vorgehen	Das Problem wird mit dem technischen Manager betreffend weiterem Vorgehen besprochen	CBT (kundenseitig)	Der Kunde berät intern wie mit dem Problem umgegangen werden soll, ob man das Device mit vermindertem Funktionsumfang auf den Markt bringt, oder die Betriebssoftware überarbeitet werden muss	
26.0	Kommunizieren der Entscheidung an Handset	Die Entscheidung wird Handset Management	CBT (kundenseitig)	Der Kunde informiert das Handset Management via E-Mail über seine	

	Management	mitgeteilt		Entscheidung	
27.0	Content für nicht bestandene Tests nicht freischalten	Der Content für nicht bestandene Tests wird nicht freigeschaltet	Handset Management	Die Content-Kategorien der fehlgeschlagenen Tests werden nicht für das Device auf der Produktions Plattform freigeschaltet	Nach der Kundenentscheidung
28.0	Speichern des Teststatus im Tracking Tool	Der Teststatus wird im Tracking Tool gespeichert	Handset Management	Speichern im Tracking Tool dass dieses Device erneut getestet werden muss (nach einem Betriebssoftware update)	
29.0	Versenden des Devices und speichern der Versanddaten im Tracking Tool	Das Device wird versendet und die Versanddaten werden im Tracking Tool gespeichert	Handset Management	Das Device wird an den Kunden zurückgesendet, die Versanddaten wie Datum, Uhrzeit und Tracking Number werden im Tracking Tool gespeichert	

3.1.3 Mögliche Probleme

3.1.3.1 Handset-Versandprobleme

Aufgrund des Handsetversands kann es zu Verzögerungen oder gar zum Verlust kommen. Hier gilt es sofort nachzuprüfen, ob das Gerät verzögert eintrifft, oder es tatsächlich verloren gegangen ist. In diesem Fall kommt folgender Prozess zum Einsatz.

3.1.3.1.1 Prozessabbildung

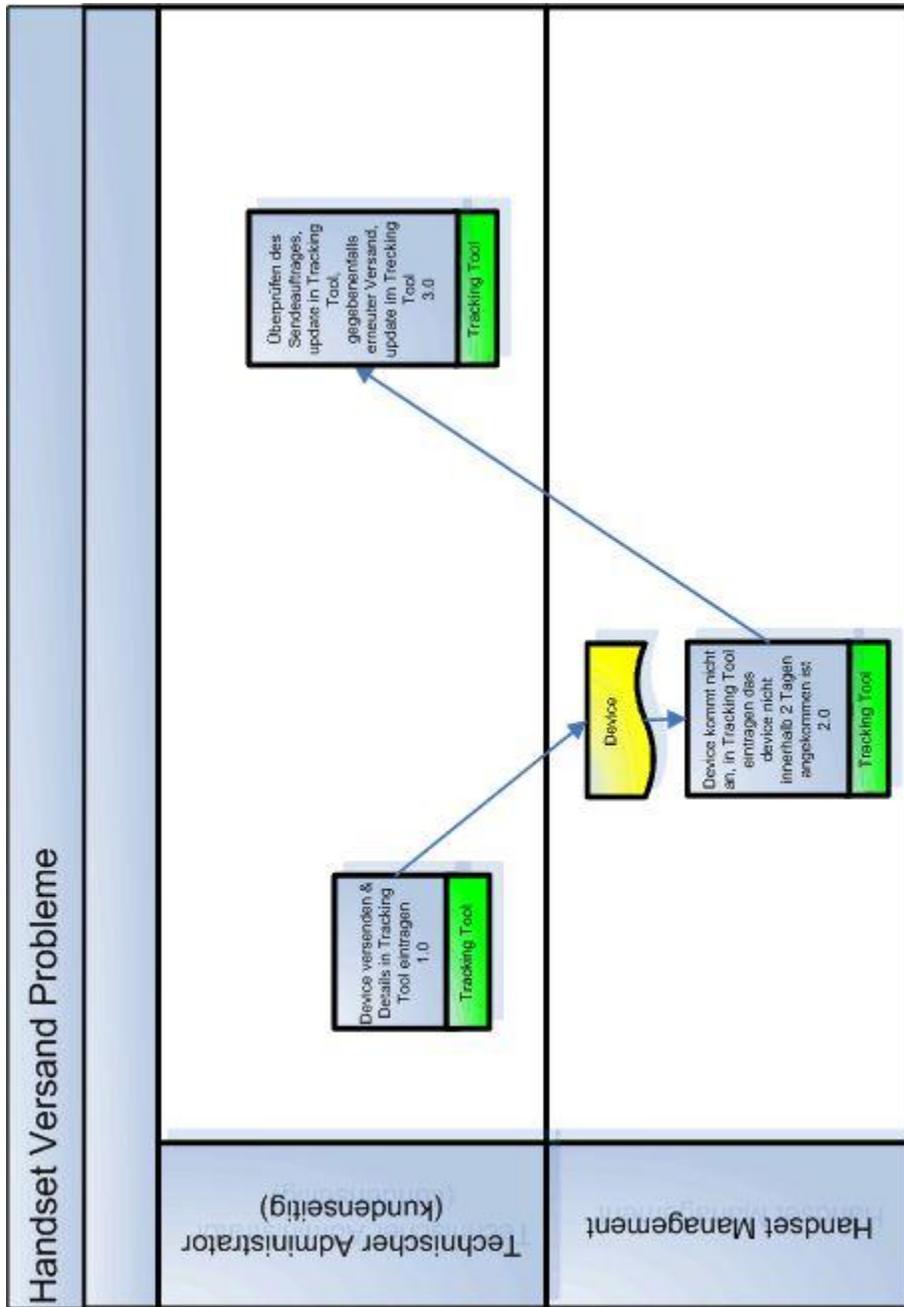


Abbildung 16 Handset Versand Probleme

3.1.3.1.2 Prozessbeschreibung

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
1.0	Device versenden und Details im Tracking Tool eintragen	Device wird versendet und Details werden im Tracking Tool eingetragen	Technischer Administrator (kundenseitig)	Das Device wird an Handset Management versendet, Details im Tracking Tool gespeichert	Bis eine Woche vor Beginn der Tests
2.0	Device kommt nicht an, in Tracking Tool eintragen dass Device nicht innerhalb von 2 Tagen angekommen ist	Im Tracking Tool vermerken - Device ist nicht innerhalb von 2 Tagen angekommen	Handset Management	Es wird im Tracking Tool vermerkt dass das Device nicht planmäßig erhalten wurde.	Bis eine Woche vor Beginn der Tests
3.0	Überprüfen des Sendeauftrages, Update im Tracking Tool, gegebenenfalls erneuter Versand, Update im Tracking Tool	Der Sendeauftrag wird überprüft, das Device gegebenenfalls erneut versendet, Update im Tracking Tool durchgeführt	Technischer Administrator (kundenseitig)	Der Sendeauftrag wird überprüft, kontaktieren des Versandunternehmens wegen Verzögerung. Bei Verlust wird ein neues Device versendet, Update im Tracking Tool	Bis eine Woche vor Beginn der Tests

3.1.3.1.3 Agiler Ansatz

Der agile Ansatz in diesem Problemfall besteht darin, die zuständige Person kundenseitig zu kontaktieren. Dies passiert beim regulärem Prozess erst durch einen kundeninternen weiteren Prozess, was einigermaßen viel Zeit in Anspruch nimmt. Der neue Ansatz hingegen hat sich als effektiv und zeitsparend erwiesen. Eine andere Möglichkeit besteht noch darin, direkt das Versandunternehmen zu kontaktieren um zu prüfen, ob die Zustellung noch am selben Tag erfolgt. Bei der vorgegebenen Prozessstruktur wird die Kontaktaufnahme zum Versandunternehmen vom Kunden erledigt, jedoch hat die Erfahrung gezeigt, dass oftmals das Gerät schon geliefert wurde bis die Anfrage des Kunden erfolgt. Dies ist wieder auf den Zeitverlust aufgrund der kundeninternen Struktur zurückzuführen. Das Nachfragen ist eine Tätigkeit, die, sofern ein Kontakt oder Zugang vorhanden ist, eine Arbeit ist welche auch selbst durchführbar ist, was meiner Meinung nach dem agilen Gedanken entspricht. Man sucht die direkte Interaktion, übernimmt gegebenenfalls mehr Aufgaben als man eigentlich müsste, nur um das Zeitintervall so kurz wie möglich zu halten.

3.1.3.1.3.1 Prozessabbildung

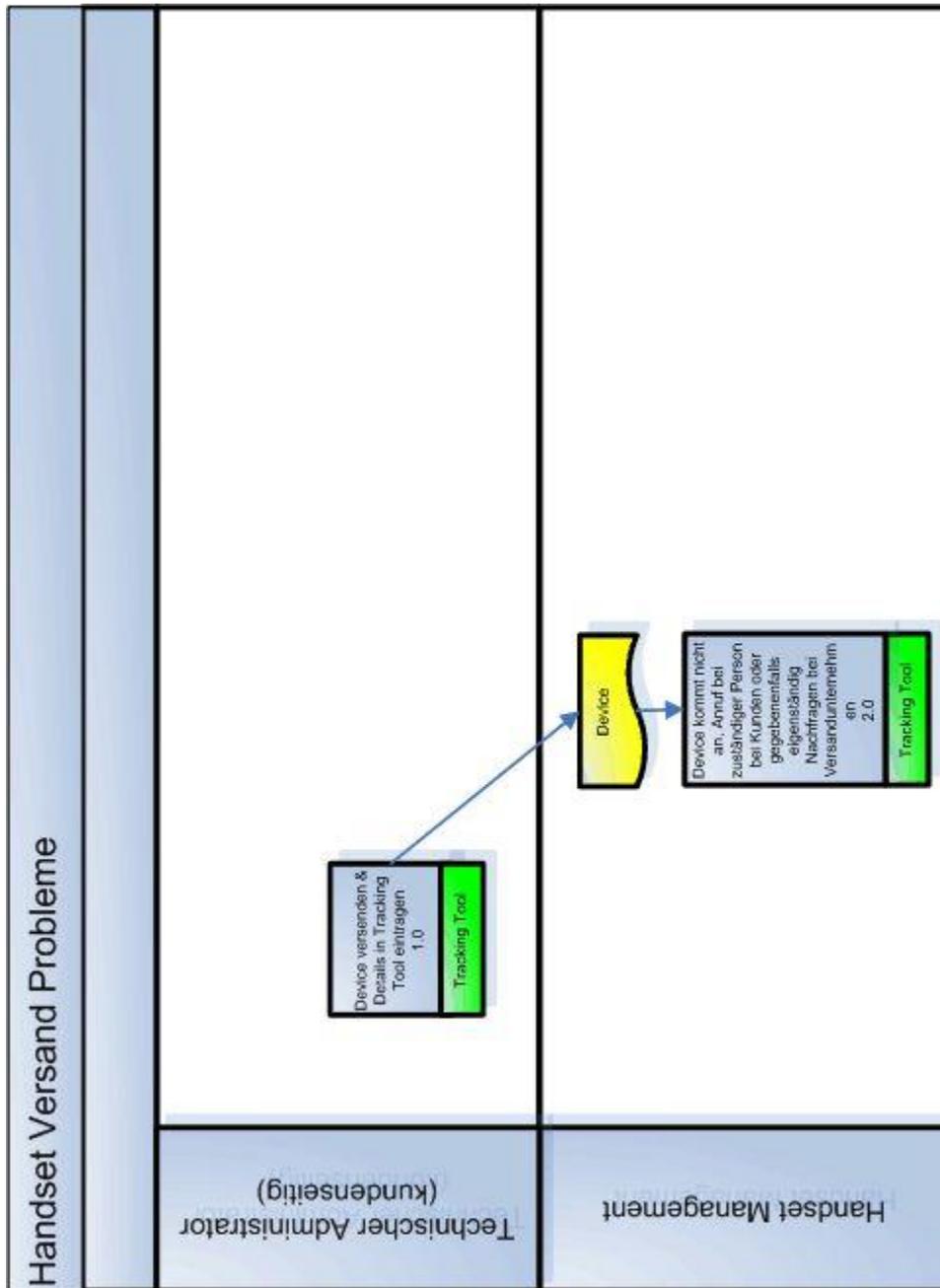


Abbildung 17 Handset Versand Probleme Agil

3.1.3.1.3.2 Prozessbeschreibung

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
1.0	Device versenden und Details im Tracking Tool eintragen	Device wird versendet und Details werden im Tracking Tool eingetragen	Technischer Administrator (kundenseitig)	Das Device wird an Handset Management versendet, Details werden im Tracking Tool gespeichert	Bis eine Woche vor Beginn der Tests
2.0	Device kommt nicht an, Anruf beim Kunden oder gegebenenfalls eigenständiges Nachfragen bei Versandunternehmen	Zuständige kundenseitige Person kontaktieren, gegebenenfalls eigenständiges Nachfragen bei Versandunternehmen	Handset Management	Die zuständige kundenseitige Person wird kontaktiert, falls möglich auch eigenständiges Nachfragen bei Versandunternehmen. Bei Verlust muss jedenfalls Kunde verständigt werden	Bis eine Woche vor Beginn der Tests

3.1.3.2 Fehlende Dokumente im Handset Management Prozess

Aus diversen Gründen kann es passieren, dass kundenseitig nicht alle Dokumente zeitgerecht zur Verfügung stehen, da jedes von einer anderen Abteilung entwickelt und verwaltet wird. Der technische Administrator hängt die einzeln erhaltenen Dokumente in das Tracking Tool. Falls eine dieser notwendigen Informationen fehlt, muss unverzüglich versucht werden an sie zu gelangen, um so schnell wie möglich mit dem Testen beginnen zu können. In diesem Fall kommt folgender Prozess zum Einsatz.

3.1.3.2.1 Prozessabbildung

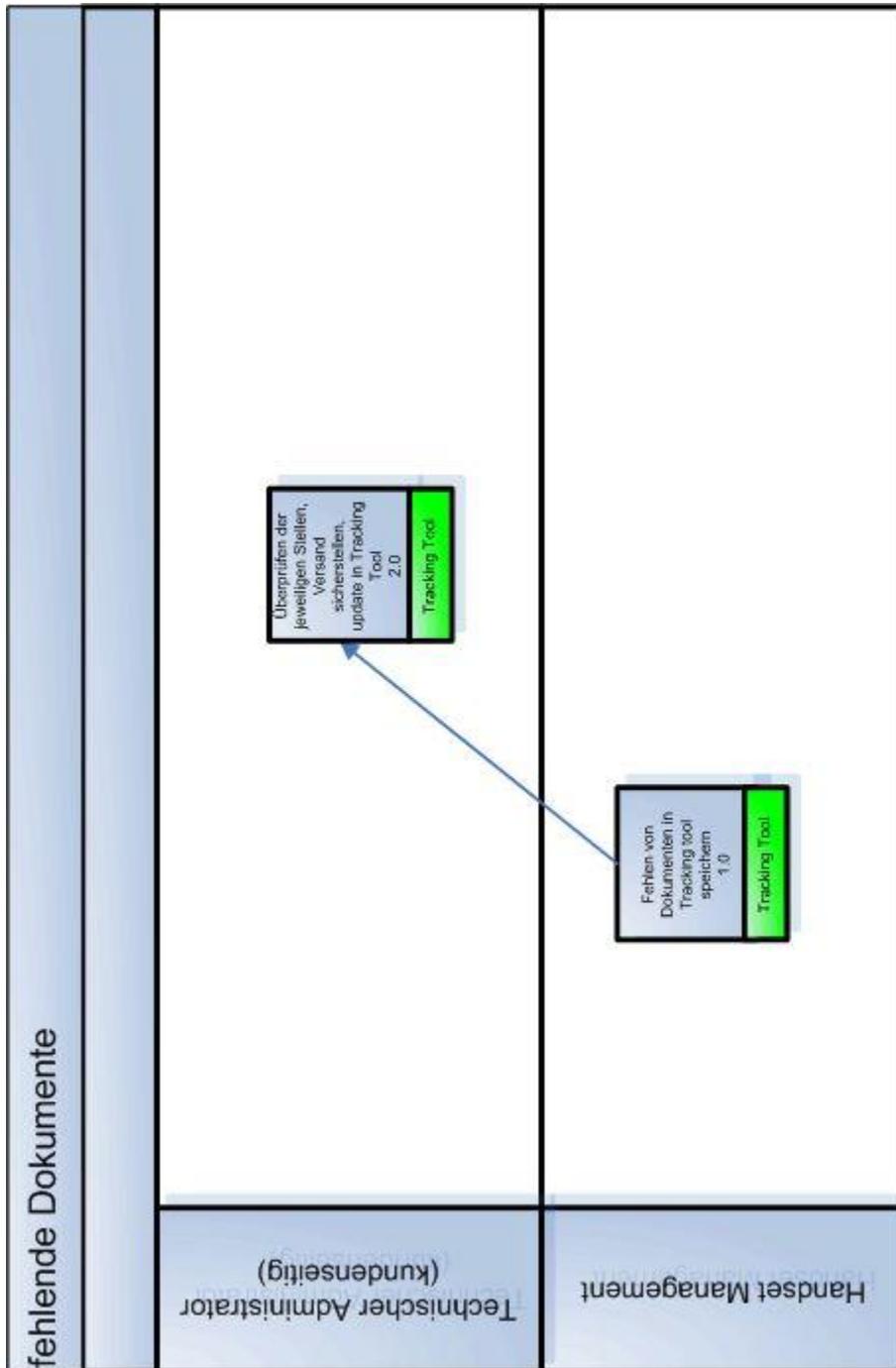


Abbildung 18 Fehlende Dokumente im Handset Management Prozess

3.1.3.2.2 Prozessbeschreibung

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
1.0	Fehlen von Dokumenten im Tracking Tool vermerken	Das Fehlen der Dokumente wird im Tracking Tool vermerkt	Handset Management	Das Fehlen von Dokumenten wird im Tracking Tool vermerkt; es wird dokumentiert, was genau fehlt.	Bis eine Woche vor Beginn der Tests
2.0	Überprüfen der jeweiligen Stellen, Versand sicherstellen, Update im Tracking Tool	Kundenseitiges Überprüfen der jeweiligen verantwortlichen Stellen, Versand sicherstellen, Update im Tracking Tool	Technischer Administrator (kundenseitig)	Kundenseitiges Überprüfen und nachfragen bei den verantwortlichen Stellen, Versand wird sichergestellt, Update im Tracking Tool	Bis eine Woche vor Beginn der Tests

3.1.3.2.3 Agiler Ansatz

Der agile Ansatz in diesem Problemfall besteht ebenfalls darin, die zuständigen Personen oder Stellen kundenseitig zu kontaktieren. Im regulären Fall würde eine zentrale Stelle die Überprüfung potentiell fehlender Dokumente übernehmen und gegebenenfalls eine Nachforschung einleiten, was einigermaßen viel Zeit in Anspruch nimmt. Der direkte Kontakt hat sich als effektiv und zeitsparend erwiesen. Da man die Personen in den jeweiligen Kundenstellen auch aufgrund des regen gewachsenen Kontakts schon persönlich kennt, wissen diese auch um die Notwendigkeit der einzelnen Dokumente, und dass jede Wartezeit die Produkteinführung auf dem Kundenmarkt erheblich beeinflussen kann. Des weiteren ist dieser Kontakt auch wichtig, um zumindest eine Zeitabschätzung erhalten zu können, was in dem regulären kundeninternen Prozess nicht vorgesehen wäre. Mit dieser Vorgehensweise kann man versuchen den Flaschenhals zu umgehen, um so in der Zwischenzeit andere Tests durchzuführen. Würde man keine Abschätzung haben, wären auch andere Tests nicht so einfach vorzuziehen, da man nicht weiß, ob die fehlenden notwendigen Informationen vielleicht bald zur Verfügung stehen.

Hier verhindert die agile Methodik auch Stillstand oder Verwirrung im Testbereich und erleichtert so die Strukturierung der weiteren Vorgehensweise.

3.1.3.2.3.1 Prozessabbildung

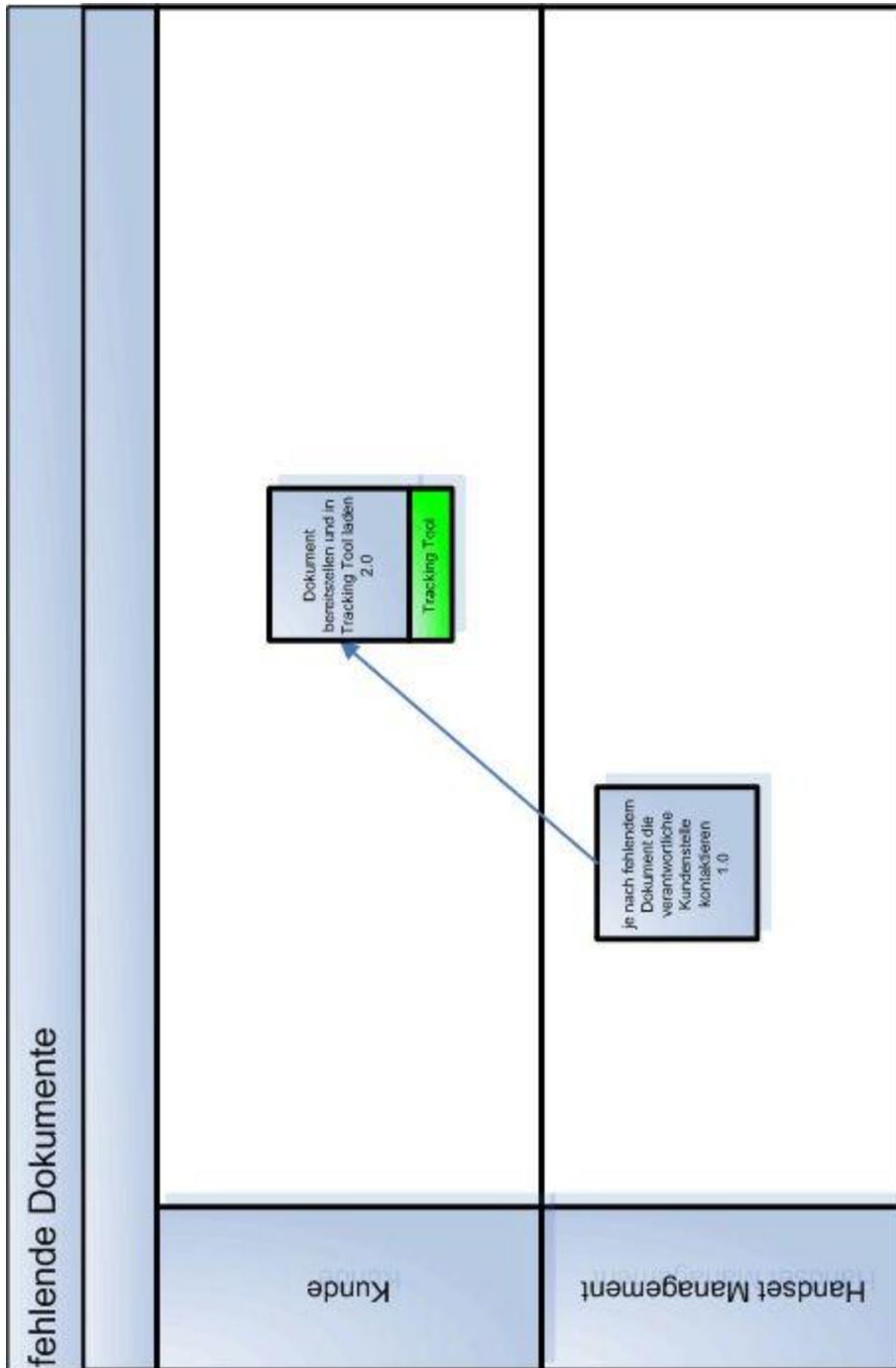


Abbildung 19 Fehlende Dokumente im Handset Management Prozess Agil

3.1.3.2.3.2 Prozessbeschreibung

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
1.0	Je nach fehlendem Dokument die verantwortliche Kundenstelle kontaktieren	Bei Fehlen von Dokumenten wird die verantwortliche Kundenstelle kontaktiert	Handset Management	Bei Fehlen von Dokumenten wird die verantwortliche Kundenstelle kontaktiert	Bis eine Woche vor Beginn der Tests
2.0	Dokument bereitstellen und ins Tracking Tool laden	Das fehlende Dokument wird bereitgestellt und ins Tracking Tool geladen	Kunde	Das fehlende Dokument wird bereitgestellt und ins Tracking Tool geladen; gegebenenfalls wird ein Zeitintervall genannt, bis wann es vorhanden sein sollen	Bis eine Woche vor Beginn der Tests

3.1.3.3 Downloadfehler

Es besteht die Möglichkeit, dass während der Tests Downloadfehler auftreten. Hier gilt zu klären, ob es interne oder externe Gründe hat. Dies ist möglich, indem man die Log-Files des Systems überprüft, um die Fehlerquellen einzugrenzen. Gegebenenfalls sind dann interne Folgeprozesse vonnöten, um den Fehler zu beheben, oder eine Rückmeldung an den Kunden, damit hier ihre intern zuständigen Stellen informiert werden können.

Interne Fehler können aufgrund von Softwaremigration und somit fehlerhafter Software entstehen, oder ein Hardware Problem ist der Grund für das Nicht-Funktionieren der benötigten Downloadfunktionen.

Kundenseitig kommen mehrere Möglichkeiten für Fehler in Betracht, wie zum Beispiel Probleme mit dem Mobiltelefonnetz und -empfang, ebenso wie Roamingprobleme oder einer Billingsoftware-Problematik. Wie man hier sehen kann, ist die Komplexität eines solchen Fehlers schon sehr groß, und es muss versucht werden ihn so schnell wie möglich zu lösen, um die Produkteinführung an den Endkunden nicht zu verzögern.

Wichtig ist hierbei auch die Unterscheidung zwischen internen und externen Tracking Tool. Das Abgleichen dieser beiden Werkzeuge ist für die Kommunikation vonnöten, jedoch entsteht damit auch ein Zeitverlust.

3.1.3.3.1 Prozessabbildung

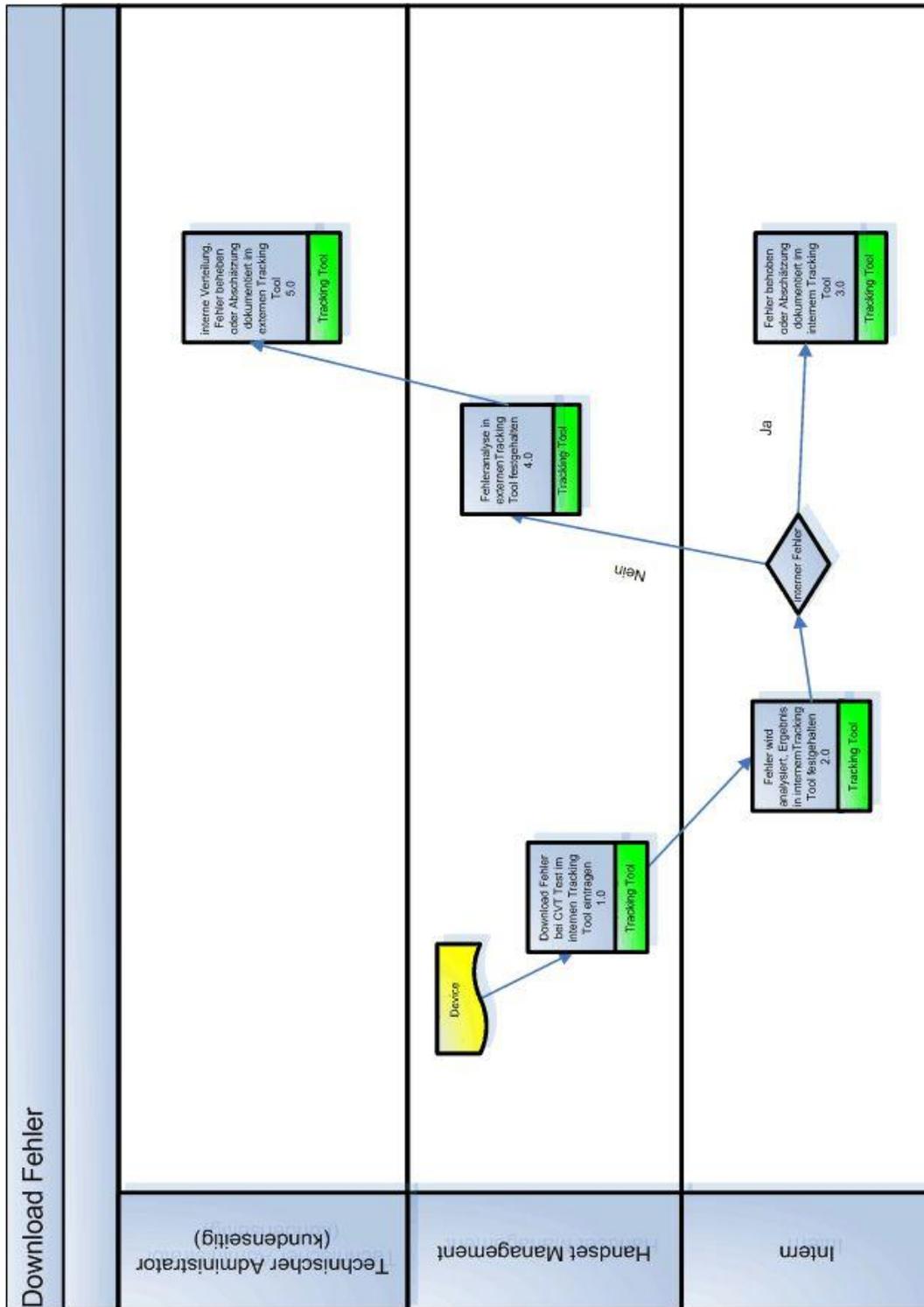


Abbildung 20 Download Fehler

3.1.3.3.2 Prozessbeschreibung

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
1.0	Downloadfehler beim CVT Test im internen Tracking Tool eintragen	Der Downloadfehler beim CVT Test wird im internen Tracking Tool eingetragen	Handset Management	Der Downloadfehler beim CVT Test wird im internen Tracking Tool mit allen benötigten Daten die für die Analyse notwendig sind eingetragen. Anschließend wird dieser Case einer internen Gruppe zugeordnet, die entscheidet welches interne Team am besten mit der Fehleranalyse betraut werden soll	Gleicher Tag wie die Probleme auftreten
2.0	Fehler wird analysiert, Ergebnis im internen Tracking Tool festgehalten	Der Fehler wird analysiert, das Ergebnis im internen Tracking Tool festgehalten.	Intern	Der Fehler wird analysiert, das Ergebnis im internen Tracking Tool mit allen nötigen Informationen der Untersuchung (Log Files,...) festgehalten.	
3.0	Fehler beheben, Abschätzung dokumentieren im internen Tracking Tool	Der Fehler wird behoben oder die Abschätzung im internen Tracking Tool	Intern	Der Fehler wird behoben oder eine Abschätzung gegeben, wann dies geschieht. Dies wird	

		dokumentiert		dann im internen Tracking Tool dokumentiert und festgehalten	
4.0	Fehleranalyse im externen Tracking Tool festhalten	Die Fehleranalyse wird im externen Tracking Tool festgehalten	Handset Management	Die Fehleranalyse wird im externen Tracking Tool festgehalten	
5.0	Interne Verarbeitung, Fehler beheben oder Abschätzung dokumentieren im externen Tracking Tool	Interne Verarbeitung beim Kunden, Fehlerbehebung oder Abschätzung im externen Tracking Tool dokumentieren	Technischer Administrator (kundenseitig)	Interne Verarbeitung beim Kunden mit Weiterleitung an die zuständigen Abteilungen mit dem Ergebnis der ersten Fehleranalyse; Fehlerbehebung oder Abschätzung wann dies passiert, wird im externen Tracking Tool dokumentiert	

3.1.3.3 Agiler Ansatz

Bei diesem Problemfall besteht der agile Ansatz darin, die Zeitabläufe zu verkürzen. Die Fehlerbehebung kann die Einführung des Devices auf dem Markt massiv verzögern, was weder im Kunden-, noch im Firmeninteresse ist. Es ist wichtig den funktionierenden Zustand so schnell wie möglich wiederherzustellen, weshalb es das Bestreben aller ist, dies einfach und zeitsparend zu gestalten. Es wird versucht, nach Auftreten des Fehlers, aufgrund von Erfahrungswerten oder eigenen Möglichkeiten der Überprüfung so viel wie möglich über den Fehler herauszufinden, um der internen Stelle alle zur Verfügung stehenden Informationen mitteilen zu können. Aufgrund von gewachsenen Strukturen und dem damit verbundenen Verständnis von Kollegen über die Wichtigkeit solcher Probleme, besteht die Möglichkeit, mit den selbst herausgefundenen Ergebnissen direkt zu den internen Ansprechpartnern und Stellen zu gehen, um mit deren Wissen dies bestätigen oder falsifizieren zu lassen. Auch hier versucht jeder Mitarbeiter sein Wissen, das manchmal auch seine Abteilungsgrenze überschreitet, bestmöglich einzusetzen, um dem Problem so schnell wie möglich auf den Grund gehen und es beheben zu können. Bei einfachen und bekannten Fehlern wird hier somit der zusätzliche Aufwand der internen Zuweisung und dem Denken nur für den eigenen Bereich umgangen. Es versucht jeder Beteiligte mit seinem ganzen Wissen aktiv teilzunehmen. Natürlich wird bei komplexeren Themen und Problemfällen der ursprüngliche Prozess mit dem Tracking Tool beibehalten, um auch dieses Wissen für andere oder neue Mitarbeiter festzuhalten, sie dem Unternehmen zu erhalten. Diese Vorgehensweise dient auf keinen Fall zur Untergrabung von Prozessen, sondern dazu diese in speziellen Fällen durch verbesserte Effizienz zu beschleunigen, das vorhandene Wissen der Mitarbeiter als Ganzes zu nutzen.

Wenn der Fehler nicht im internen Bereich liegt, wird das damit verbundene Wissen (Log-Files,...) in das externe Tracking Tool gespeichert. Auch hier kann man eine leichte Veränderung durch den agilen Ansatz erkennen, denn aufgrund des vorhandenen Wissens und der Expertise in diesem Bereich kann oftmals schon eine bestimmte Richtung vorgegeben werden, in welchem Kundenbereich sich der Fehler verbergen könnte. Dies wird zusätzlich via E-Mail an den technischen Administrator kundenseitig gesendet, oder, da die kundenseitigen Ansprechpartner aufgrund gewachsener Kommunikation schon bekannt sind, gleich an diese zuständigen

Personen weitergeleitet, um in beidseitigem Einverständnis die Zeitspanne bis zur Behebung des Problems so kurz wie möglich zu gestalten. Auch bekommt man auf diesem Wege eine schnelle Abschätzung, wie lange die Behebung dauern wird, um in der Zwischenzeit diesen blockierenden Zeitraum konstruktiv mit anderen Arbeiten verplanen zu können. Der direkte Kontakt hat sich auch in diesem Fall als effektiv und zeitsparend erwiesen, das persönliche Interagieren vorrangig gegenüber der reinen Prozessstruktur, diese zwar nicht ersetzend, aber ergänzend.

3.1.3.3.1 Prozessabbildung

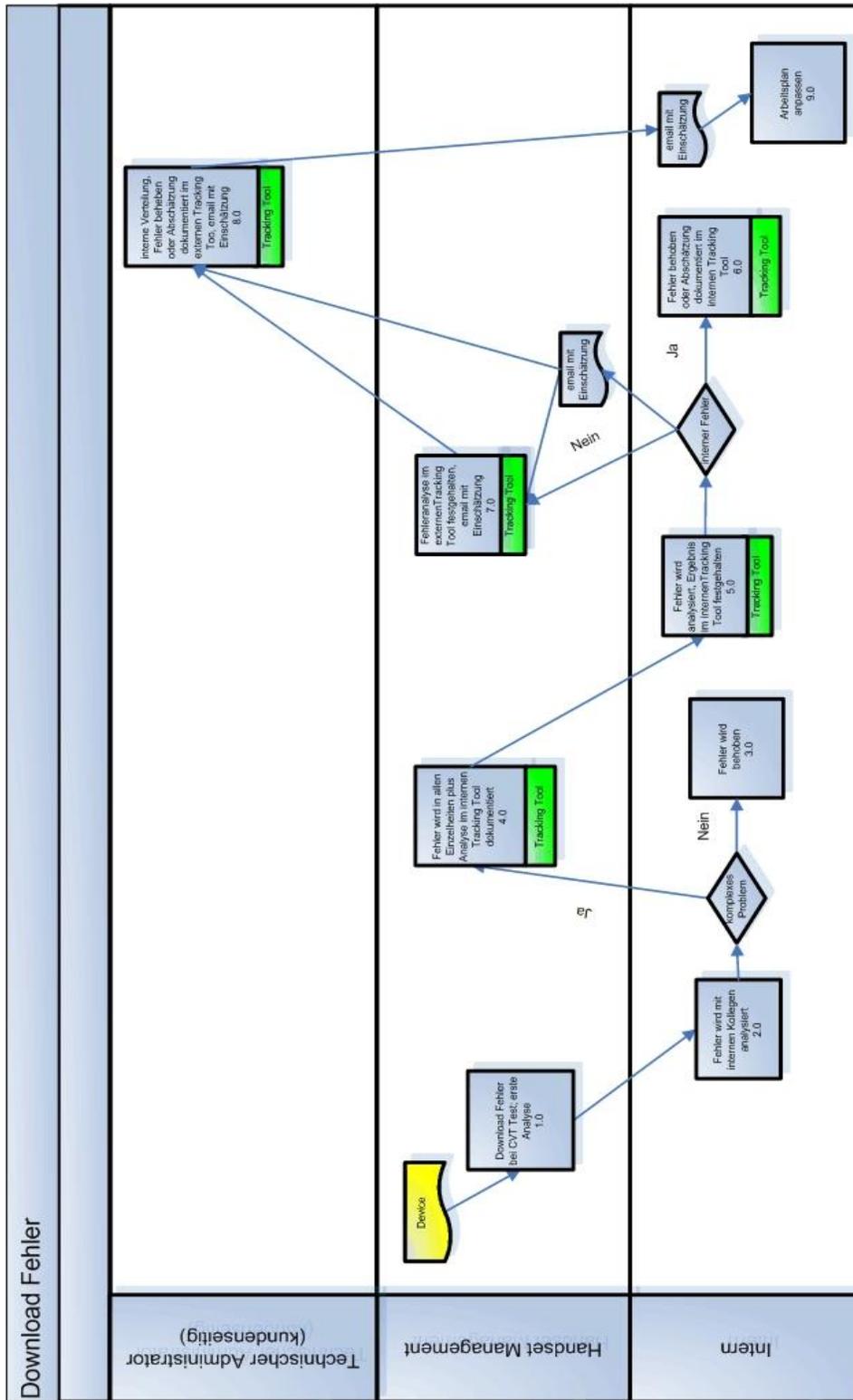


Abbildung 21 Download Fehler Agil

3.1.3.3.2 Prozessbeschreibung

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
1.0	Downloadfehler bei CVT Test ; erste Analyse	Der Downloadfehler beim CVT Test wird vorab analysiert	Handset Management	Der Downloadfehler beim CVT Test wird von Handset Management vorab durch Erfahrungswerte und nach eigenen Möglichkeiten der Überprüfung so weit wie möglich analysiert	Gleicher Tag wie die Probleme auftreten
2.0	Fehler wird mit internen Kollegen analysiert	Der Fehler wird mit internen Kollegen analysiert	Intern	Der Fehler wird mit internen Kollegen analysiert, gesamtes Wissen zur Fehlerbehebung wird eingebracht	Gleicher Tag wie die Probleme auftreten
3.0	Fehler wird behoben	Der Fehler wird behoben	Intern	Der Fehler wird durch die intern durchgeführte Analyse behoben	Gleicher Tag wie die Probleme auftreten
4.0	Fehler und Analyse werden in allen Einzelheiten im internen Tracking Tool dokumentiert	Der Fehler und seine Analyse werden in allen Einzelheiten im internen Tracking Tool dokumentiert	Handset Management	Der Fehler und seine Analyse wird in allen Einzelheiten im internen Tracking Tool dokumentiert	Gleicher Tag wie die Probleme auftreten
5.0	Fehler wird analysiert, Ergebnis im internen Tracking	Der Fehler wird analysiert, das Ergebnis wird im internen Tracking	Intern	Der Fehler wird analysiert, das Ergebnis im internen Tracking Tool mit allen nötigen	

	Tool festgehalten	Tool festgehalten		Informationen der Untersuchung (Log Files,...) festgehalten.	
6.0	Fehler wird behoben oder Abschätzung im internen Tracking Tool dokumentiert	Der Fehler wird behoben oder eine Abschätzung im internen Tracking Tool dokumentiert	Intern	Der Fehler wird behoben oder eine Abschätzung gegeben, wann dies geschieht. Dies wird dann im internen Tracking Tool dokumentiert	
7.0	Fehleranalyse wird im externen Tracking Tool festgehalten, E-Mail mit Einschätzung	Die Fehleranalyse wird im externen Tracking Tool festgehalten, ein E-Mail mit Fehlereinschätzung wird gesendet	Handset Management	Die Fehleranalyse wird im externen Tracking Tool festgehalten, ein E-Mail mit Fehlereinschätzung wird gesendet	
8.0	Interne Verteilung, Fehler beheben oder Abschätzung im externen Tracking Tool dokumentiert, E-Mail mit Abschätzung	Interne Verteilung beim Kunden, Fehler wird behoben oder Abschätzung im externen Tracking Tool dokumentiert, E-Mail mit Abschätzung	Technischer Administrator (kundenseitig)	Interne Verteilung beim Kunden, Fehler wird behoben oder Abschätzung im externen Tracking Tool dokumentiert; falls Ansprechpartner bekannt ist wird ein E-Mail mit Abschätzung mitgesendet	
9.0	Arbeitsplan anpassen	Arbeitsplan wird angepasst	Handset Management	Arbeitsplan wird gegebenenfalls durch Einschätzung im E-Mail angepasst	

3.1.3.4 Medientyp-Fehler

Eine weitere Fehlerquelle können die diversen Medientypen darstellen. Hier können unterschiedliche Fehler beobachtet werden, wie zum Beispiel dass das File nicht abgespielt oder gespeichert werden kann. Auch hier gilt es zu klären, ob dies interne oder externe Gründe hat. Dies ist möglich, indem man die Log-Files des Systems durchschaut, um Fehlerquellen einzugrenzen. Gegebenenfalls sind anschließend interne Folgeprozesse vonnöten, um zu entscheiden, ob der Fehler behoben werden kann, oder doch eine Meldung an den Kunden erforderlich ist. Interne Fehler können aufgrund fehlerhafter Software, aber auch durch beschädigte Medientypen auftreten

Kundenseitig kann es zu Problemen mit der Softwareversion des Handsets kommen - dies muss dann mit dem Hersteller besprochen werden. Der Kunde trifft im Anschluss eine Entscheidung, ob der Test einfach als nicht bestanden eingetragen oder beendet und auf eine neue Softwareversion gewartet werden soll. In diesem Fall würde das Device wieder an den Kunden zurückgesendet werden; die Tests starten dann nach dem Eintreffen des neuen Endgerätes mit durchgeführtem Softwareupdate erneut.

Wie man hier schon sehen kann, ist die Komplexität auch aufgrund einer möglichen Beteiligung des Herstellers an solchen Problemen sehr groß. Es muss daher versucht werden, so schnell wie möglich eine Lösung zu finden, um die Produkteinführung nicht weiter zu verzögern.

Wichtig ist auch hierbei die Unterscheidung zwischen internen und externen Tracking Tools. Unser Unternehmen arbeitet mit verschiedenen dieser Tools, wie auch der Kunde. So benutzt der Kunde zur internen Kommunikation in diesen Prozessen ein anderes Tracking Tool (externes) als wir intern und für die Kundenkommunikation (internes). Das Abgleichen dieser beiden Werkzeuge ist für die Kommunikation vonnöten, jedoch entsteht damit auch ein Zeitverlust.

3.1.3.4.1 Prozessabbildung

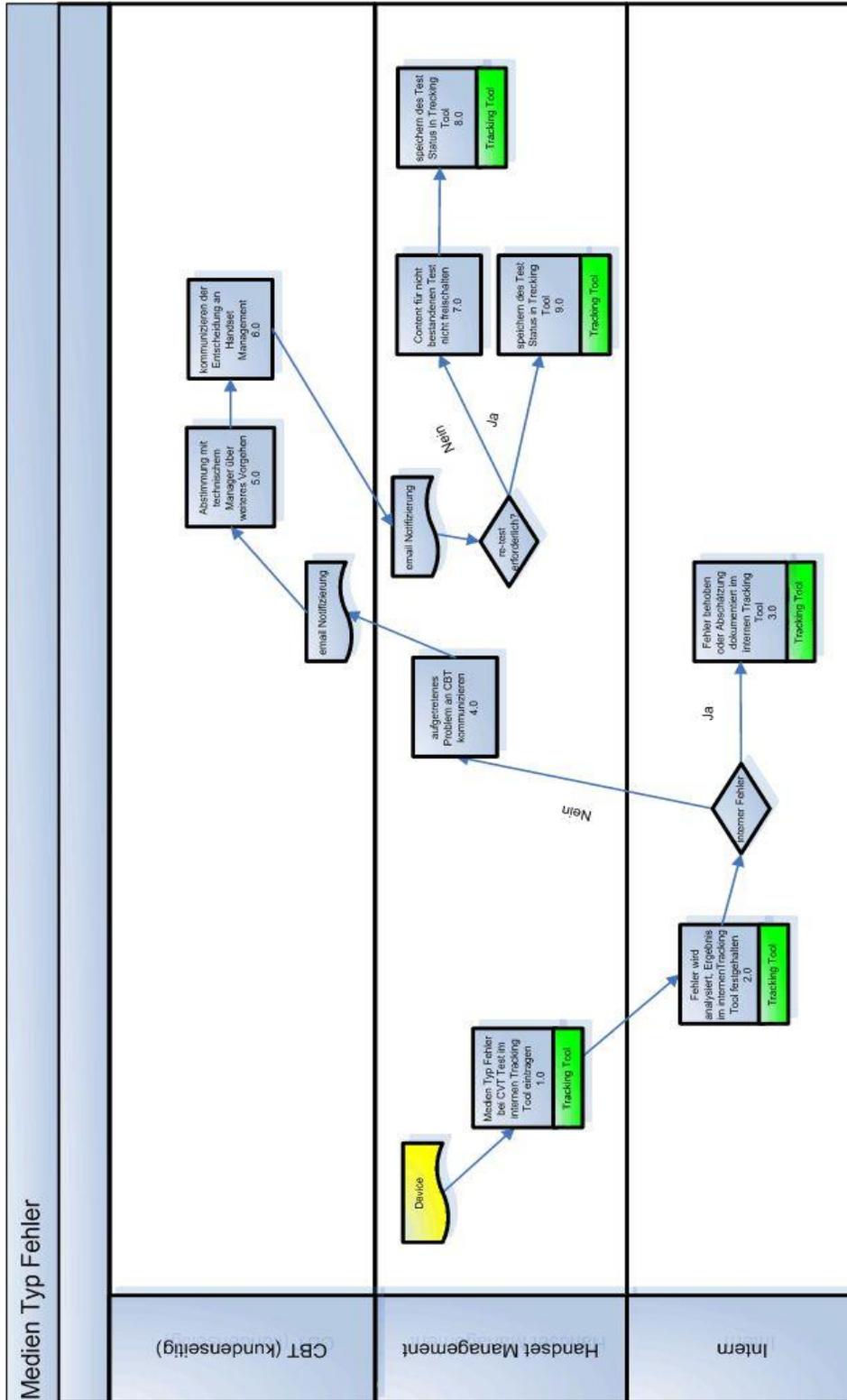


Abbildung 22 Medien Typ Fehler

3.1.3.4.2 Prozessbeschreibung

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
1.0	Medientyp-Fehler beim CVT Test im internen Tracking Tool eintragen	Der Medientyp-Fehler beim CVT Test wird im internen Tracking Tool eingetragen	Handset Management	Der Medientyp-Fehler beim CVT Test wird im internen Tracking Tool, mit allen benötigten Daten die für die Analyse notwendig sind, eingetragen. Anschließend wird dieser Case einer internen Gruppe zugeordnet, die entscheidet welches interne Team am besten mit der Fehleranalyse betraut werden soll	Gleicher Tag wie die Probleme auftreten
2.0	Fehler wird analysiert, Ergebnis im internen Tracking Tool festgehalten	Der Fehler wird analysiert und das Ergebnis im internen Tracking Tool festgehalten.	Intern	Der Fehler wird analysiert und das Ergebnis im internen Tracking Tool mit allen nötigen Informationen dieser Untersuchung (Log Files,...) festgehalten.	
3.0	Fehler beheben oder Abschätzung im internen Tracking Tool dokumentieren	Der Fehler wird behoben oder eine Abschätzung im internen Tracking	Intern	Der Fehler wird behoben oder eine Abschätzung gegeben, wann dies geschieht. Dies wird dann im	

		Tool dokumentiert		internen Tracking Tool dokumentiert	
4.0	Aufgetretenes Problem an CBT kommunizieren	Das aufgetretene Problem wird an CBT kommuniziert	Handset Management	Die aufgetretenen Probleme werden via E-Mail an das kundenseitige CBT Team mit Datum, Uhrzeit und Fehlerbeschreibung gesendet	
5.0	Abstimmung mit technischem Manager über weiteres Vorgehen	Das Problem wird mit dem technischen Manager betreffend weiterem Vorgehen besprochen	CBT (kundenseitig)	Der Kunde berät intern, wie mit dem Problem umgegangen werden soll - etwa ob das Device mit vermindertem Funktionsumfang auf den Markt gebracht, oder die Betriebssoftware überarbeitet werden soll	
6.0	Kommunizieren der Entscheidung an Handset Management	Die Entscheidung wird Handset Management mitgeteilt	CBT (kundenseitig)	Der Kunde informiert Handset Management via E-Mail über seine Entscheidung	
7.0	Content für nicht bestandene Tests nicht freischalten	Der Content für nicht bestandene Tests wird nicht freigeschaltet	Handset Management	Die Contentkategorie der fehlgeschlagenen Tests wird nicht für das Device auf der Produktionsplattform freigeschaltet	Nach der Kundenentscheidung
8.0	Speichern des Teststatus im	Der Teststatus wird im Tracking Tool	Handset	Im Tracking Tool wird das Datum, die Uhrzeit	Nach Beendigung

	Tracking Tool	gespeichert	Management	und Status der Tests gespeichert	des CVT Tests
9.0	Speichern des Teststatus im Tracking Tool	Der Teststatus wird in Tracking Tool gespeichert	Handset Management	Im Tracking Tool wird gespeichert, dass dieses Device erneut getestet werden muss (nach einem Betriebssoftware-Update)	

3.1.3.4.3 Agiler Ansatz

Der agile Ansatz bei diesem Problemfall besteht wieder darin, die Zeitabläufe zu verkürzen. Die Fehlerbehebung kann auch in diesem Fall die Einführung des Devices auf dem Markt massiv verzögern, was weder im Kunden-, noch im Firmeninteresse ist. Es sollte auch hier so schnell wie möglich das Problem gelöst werden. Deshalb ist das Bestreben aller dies einfach und zeitsparend zu gestalten. Auch hier wird versucht, nach Auftreten des Fehlers aufgrund von Erfahrungswerten und nach eigenen Möglichkeiten der Überprüfung so viel wie möglich über den Fehler herauszufinden, um anschließend der internen Stelle alle zur Verfügung stehenden Informationen mitteilen zu können. Auch in diesem Falle ist es aufgrund von gewachsenen Strukturen und dem damit verbundenen Verständnis von Kollegen für die Wichtigkeit solcher Probleme möglich, mit den selbst herausgefundenen Ergebnissen direkt zu den internen Ansprechpartnern und Stellen zu gehen, um dies mit ihrem Wissen bestätigen oder falsifizieren zu lassen. Abteilungsübergreifendes Wissen wird auch in diesem Fall genutzt, um das reibungslose Testen wiederherzustellen. Somit wird hier bei einfachen und bekannten Fehlern ebenfalls der zusätzliche Aufwand der internen Zuweisung, und auf den eigenen Bereich beschränktes Denken umgangen. Jeder Beteiligte versucht mit seinem ganzen Wissen aktiv teilzunehmen. Natürlich wird bei komplexeren Themen und Problemfällen der ursprüngliche Prozess mit dem Tracking Tool beibehalten, um dieses Wissen auch für andere oder neue

Mitarbeiter festzuhalten, sie also dem Unternehmen zu erhalten. Diese Vorgehensweise soll auf keinen Fall Prozesse untergraben, sondern diese in speziellen Fällen durch verbesserte Effizienz beschleunigen, das vorhandene Wissen also als Ganzes nutzen.

Wenn der Fehler nicht im internen Bereich liegt, wird das damit verbundene Wissen (Log-Files,...) an das kundeninterne CBT Team via E-Mail kommuniziert. Auch hier kann man eine leichte Veränderung mit agilem Ansatz erkennen, denn aufgrund des vorhandenen Wissens und der Expertise in diesem Bereich, kann oftmals schon eine bestimmte Richtung vorgegeben werden in welchem Bereich sich der Fehler verbergen könnte - ob es etwa an der Plattform oder am Endgerät liegt. Auch bekommt man auf diesem Wege schnelleres Feedback ob der Test wiederholt werden, das Gerät also an den Kunden zurückgesendet und auf ein Softwareupdate gewartet, oder der Content nicht aktiviert werden soll, um in der Zwischenzeit diesen blockierenden Zeitraum konstruktiv mit anderen Arbeiten verplanen zu können. Der direkte Kontakt hat sich auch in diesem Fall als effektiv und zeitsparend erwiesen, das persönliche Interagieren vorrangig gegenüber der reinen Prozessstruktur, diese nicht ersetzend, sondern ergänzend.

3.1.3.4.3.1 Prozessabbildung

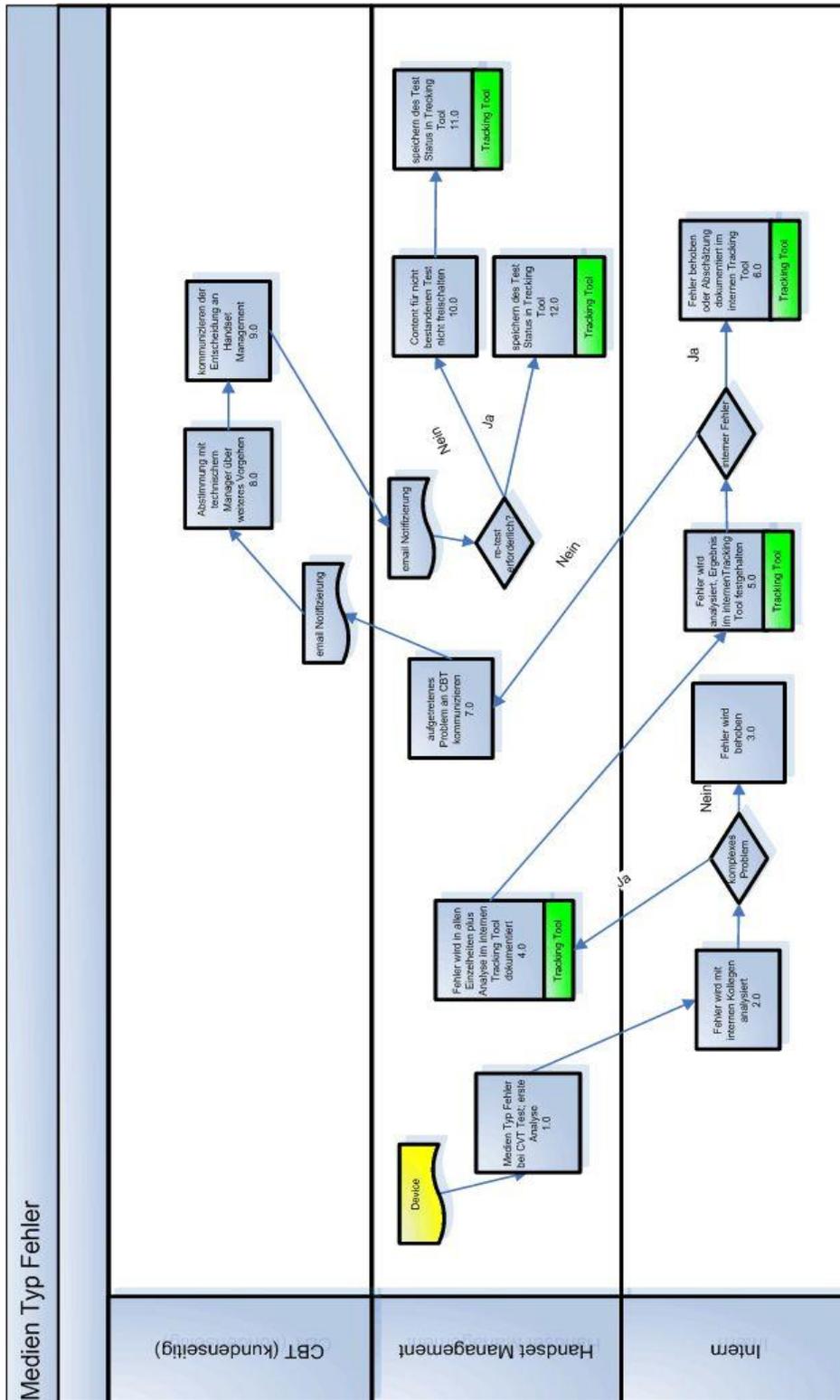


Abbildung 23 Medien Typ Fehler Agil

3.1.3.4.3.2 Prozessbeschreibung

Abschnitt	Abschnittname	Beschreibung	Rolle	Was gemacht werden muss	Qualitätskriterium
1.0	Downloadfehler bei CVT Test ; erste Analyse	Der Downloadfehler beim CVT Test wird vorab analysiert	Handset Management	Der Downloadfehler beim CVT Test wird von Handset Management vorab durch Erfahrungswerte und nach eigenen Überprüfungsmöglichkeiten so weit wie möglich analysiert	Gleicher Tag wie die Probleme auftreten
2.0	Fehler wird mit internen Kollegen analysiert	Der Fehler wird mit internen Kollegen analysiert	Intern	Der Fehler wird mit internen Kollegen analysiert - gesamtes Wissen zur Fehlerbehebung wird eingebracht	Gleicher Tag wie die Probleme auftreten
3.0	Fehler wird behoben	Der Fehler wird behoben	Intern	Der Fehler wird durch die intern durchgeführte Analyse behoben	Gleicher Tag wie die Probleme auftreten
4.0	Fehler und Analyse werden in allen Einzelheiten im internen Tracking Tool dokumentiert	Der Fehler und seine Analyse werden in allen Einzelheiten im internen Tracking Tool dokumentiert	Handset Management	Der Medientyp-Fehler und seine Analyse werden in allen Einzelheiten im internen Tracking Tool dokumentiert	Gleicher Tag wie die Probleme auftreten
5.0	Fehler wird analysiert, Ergebnis im internen Tracking	Der Fehler wird analysiert, das Ergebnis im internen Tracking	Intern	Der Fehler wird analysiert, das Ergebnis im internen Tracking Tool mit allen nötigen	

	Tool festgehalten	Tool festgehalten.		Informationen von der Untersuchung (Log Files,...) festgehalten.	
6.0	Fehler beheben oder Abschätzung im internen Tracking Tool dokumentieren	Der Fehler wird behoben oder eine Abschätzung im internen Tracking Tool dokumentiert	Intern	Der Fehler wird behoben oder eine Abschätzung gegeben, wann dies geschieht. Dies wird dann im internen Tracking Tool dokumentiert	
7.0	Aufgetretenes Problem an CBT kommunizieren	Das aufgetretene Problem wird an CBT kommuniziert	Handset Management	Die aufgetretenen Probleme werden via E-Mail an das kundenseitige CBT Team mit Datum, Uhrzeit, Fehlerbeschreibung und Fehlereinschätzung gesendet	
8.0	Abstimmung mit technischem Manager über weiteres Vorgehen	Das Problem wird mit dem technischen Manager betreffend weiterem Vorgehen besprochen	CBT (kundenseitig)	Der Kunde berät intern, wie mit dem Problem umgegangen werden soll - etwa ob man das Device mit vermindertem Funktionsumfang auf den Markt bringt, oder die Betriebssoftware überarbeitet werden muss; die Fehlereinschätzung von Handset Management beschleunigt diesen	

				Prozess	
9.0	Kommunizieren der Entscheidung an Handset Management	Die Entscheidung wird Handset Management mitgeteilt	CBT (kundenseitig)	Der Kunde informiert Handset Management via E-Mail über seine Entscheidung	
10.0	Content für nicht bestandenen Test nicht freischalten	Der Content für den nicht bestandenen Test wird nicht freigeschaltet	Handset Management	Die Contentkategorie des fehlgeschlagenen Tests wird nicht für das Device auf der Produktionsplattform freigeschaltet	Nach der Kundenentscheidung
11.0	Speichern des Teststatus im Tracking Tool	Der Teststatus wird im Tracking Tool gespeichert	Handset Management	Im Tracking Tool das Datum, die Uhrzeit und den Status der Tests speichern	Nach Beendigung des CVT Tests
12.0	Speichern des Teststatus im Tracking Tool	Der Teststatus wird im Tracking Tool gespeichert	Handset Management	Im Tracking Tool speichern, dass dieses Device erneut getestet werden muss (nach einem Betriebssoftware update)	

4 Persönliche Einschätzung agiles Management im Bereich des internationalen Handset Managements

4.1 Aussage dieser Arbeit

Die Abänderung der herkömmlichen Prozesse nach agilen Ansätzen hat absolut positive Auswirkungen auf die bisherigen Arbeitsvorgänge gehabt und wurde von den Mitarbeitern ohne Probleme angenommen und gelebt. Ich habe mich schon oftmals gefragt, aus welchem Grund diese Arbeitsweise funktioniert, warum dieses scheinbar „wilde“ Handeln, das Umändern von Prozessen, einen so positiven Effekt auf den Output haben kann. Erst als ich die ersten Artikel über agiles Arbeiten, agile Projekte und ihre Unterschiede zu herkömmlichen, klassischen Arbeitsweisen gelesen hatte, konnte ich sehen, dass auch hier ein System vorliegt und Parallelen entdecken. Dass etwas das vielleicht auf den ersten Blick anders, unkonventionell, antiautoritär, nicht alltäglich, unangepasst, unüblich, ungewöhnlich, ungewohnt, ja, sogar nicht vorstellbar wirkt, doch so eine messbare Verbesserung bringen kann. Ich wollte verstehen, wie sich diese Art und Weise zu denken und zu arbeiten erklären lässt, und fand für mich im agilen Arbeiten die Erklärung.

4.2 Verbesserungen und Wissensgewinne

Ein wesentlicher Punkt, den wir damit verbessern konnten, ist die zeitliche Verkürzung der Arbeitsschritte. Da jedem Mitarbeiter die Wichtigkeit dieser Problemlösungen und Problematiken vor Augen gehalten wurde, versucht nun jede beteiligte Person sein ganzes Wissen und seine kompletten Fähigkeiten mit einzubringen. Natürlich bedeutet dies etwas mehr Aufwand für den Einzelnen, wenn man mehr als seine ursprünglichen Aufgabengebiete im Auge behält - jedoch ist dieses Vorgehen für das Team natürlich unglaublich förderlich und steigert die

Produktivität. Besonders der Teamgeist, das Zusammengehörigkeitsgefühl und die soziale Interaktion wurden im Zuge der agilen Einflüsse weiter verbessert und ausgebaut. Natürlich ist so ein enges Zusammenarbeiten, das zweifellos entstanden ist, und auch das Entlasten der Kollegen, meiner subjektiven Meinung nach, nur bei einer gewissen Grundsympathie möglich. Auch aus eigener Erfahrung kann ich sagen, dass sobald ein Mitarbeiter sich nicht in ein Team sozial integrieren kann, die Arbeitsaufteilung und das -klima, besonders bei agilen Methoden, sehr schwer und belastet werden. Dies beschränkt sich nicht nur auf den einzelnen Mitarbeiter, der sich vielleicht schwieriger ins Team integrieren oder überhaupt soziale Beziehungen aufbauen kann, sondern hat auch Auswirkungen auf die Teamleistung an sich. Erfahrungswerte zeigen dass, sobald sich ein Mitarbeiter nicht ganz im Team integriert, nicht nur bezogen auf sein Wissen, Handeln oder seine Fähigkeiten, sondern besonders auf seine sozialen Kompetenzen, dies auf jeden Fall Produktivität kostet - mehr noch als bei traditionellen Entwicklungs- und Prozessmethoden. Dieses Feedback ist besonders wichtig für das Personalmanagement, für Vorstellungsgespräche. Aus diesem Grund werden auch Teammitglieder eingeladen, bei diesen Gesprächen beizusitzen, um auch ihren Eindruck über die Kandidaten abgeben zu können.

Natürlich ist dieses agile Vorgehen auch im Interesse des Kunden, um das Handset so schnell wie möglich auf dem Markt einführen zu können. Auch hier ließ sich der agile Ansatz recht problemlos mit den diversen kundenseitigen Ansprechpartnern einführen. Kritische Stimmen verstummten sehr schnell aufgrund der verbesserten Performance.

Ich finde es auch wichtig festzuhalten, dass aufgrund des Nicht-Benutzens des Tracking Tools bei bekannten Fehlern der Firma kein Wissen verloren geht. Diese Fälle wurden schon zu einem früheren Zeitpunkt dokumentiert und sind daher bekannt. Man verwendet nur das bereits erworbene Wissen daraus, um das Problem zu beheben, anstatt dieses nochmals festzuhalten. Neue, veränderte oder kombinierte Fehler, die in dieser Form noch nicht aufgetreten sind, werden natürlich prozessnormiert in den Tracking Tools festgehalten, um das Wissen zu sichern und so bei einem Wiederauftreten desselben Problems Zeit zu sparen und so effizient wie möglich zu handeln.

4.3 Soziale Verbesserungen

Einer der Punkte, der mir anfangs nicht wirklich aufgefallen ist, ist die allgemeine Verbesserung der Kommunikation und des sozialen Umgangs - sowohl innerhalb des Teams, als auch mit den externen Ansprechpartnern. Wie schon im vorigen Kapitel von mir beschrieben, ist die Grundchemie, die es in einem Team geben muss, das Verstehen mit den Kollegen, ein für mich wesentlicher Punkt bei unseren agilen Methoden. Jedoch sehe ich auch aus meinen Erfahrungen, dass dies den Zusammenhalt innerhalb des Teams wesentlich stärkt, da man ja auch teilweise Aufgaben der anderen mit übernimmt. Da auch dies nicht einseitig passieren kann, stellt man im Gegensatz dazu auch anderen Kollegen sein Wissen hilfreich zur Verfügung - ein ständiges Miteinander und auch persönliches Interagieren statt einem Nebeneinander mit nur förmlicher Kommunikation.

Was man früher allein durch Teamevents wie Weihnachtsfeiern oder ähnliches erreichen konnte, wächst hier auf eine natürliche Art und Weise, wird von diesen Firmenveranstaltungen aber noch zusätzlich gestärkt. Natürlich basiert dieses Wissen nur auf meiner persönlichen Erfahrung.

4.3.1 Kritische Diskussion

Diese tägliche, enge Zusammenarbeit kann natürlich auch Probleme mit sich bringen - vielleicht sogar schneller und stärker als dies bei herkömmlichen Arbeitsformen der Fall wäre. Man stellt sehr schnell Sympathie und Antipathie zu Personen fest, mit der man lernen muss umzugehen. Aufgrund der professionellen Einstellung aller wird dies normalerweise auch zu keinem Problem, jedoch kann es dann mit fortlaufender Interaktion auch hier zu Spannungen kommen, die der Produktivität nicht dienlich sind, weil sie das aktive aufeinander Zugehen und Helfen beeinträchtigen.

Ein Lösungsansatz hierzu wäre das einbeziehen des Teams, oder zumindest von einigen Teammitgliedern in diverse Vorstellungsgespräche von möglichen neuen Mitarbeitern, damit auch von dieser Seite konstruktives Feedback gegeben werden kann.

4.4 Diskursive Reflexion

In diesem Kapitel wurden Kollegen, die ebenfalls mit Agilen Techniken arbeiten, befragt. Es wurde extra ein Querschnitt durch verschiedene Abteilungen gezogen, so waren 11 Mitarbeiter vom Client Management, Quality Services, dem Service Management, der Development Abteilung, sowie aus den Management Bereichen beteiligt (2 Frauen und 9 Männer). Es handelte sich um teilstrukturierte Interviews, jedes einzelne dauerte zwischen 10 und 20 Minuten. Die Fragen wurden bewusst in Richtung der Kommunikation und ihrer Entwicklungsmöglichkeit gestellt, da dies für mich einen wesentlicher Faktor der Agilität darstellt, und aus meiner Erfahrung heraus wichtig für das Funktionieren eines Teams ist. Ich wollte mit den Fragen absichtlich keine Ja/Nein Antworten erhalten, um so die Kollegen ebenfalls aus ihrem Erfahrungsschatz erzählen zu lassen. Eine Gruppendiskussion, die ich zuerst führen wollte, habe ich wieder verworfen da ich von jeder einzelnen Person ihre ganz eigene Meinung wissen wollte, ohne diese in einer Diskussion in der Gruppe vielleicht verfälschen zu lassen.

4.4.1 Fragestellungen

Folgende Fragen wurden den Kollegen im Rahmen des Interviews gestellt:

1. Wie wichtig finden Sie die Kommunikation mit anderen Mitarbeitern auch in punkto ihrer Transparenz, bzw. wie wichtig bewerten Sie positive oder negative Eindrücke bei neuen Kollegen?
2. Wie beurteilen Sie Geduld und Toleranz und Unvoreingenommenheit im Umgang mit Ihren Kollegen, bzw. wie wichtig ist Ihnen der respektable Umgang?

3. Wie bedeutend finden Sie exaktes Zuhören und Antworten sowie Geradlinigkeit bei Antworten in Gesprächen mit Kollegen?
4. Wie wichtig finden Sie gefühlte Intuitionen in der technischen Agilität? Hat Ihr Bauchgefühl Einfluss auf Ihre Entscheidungen?
5. Würden Sie es für wichtig und sinnvoll erachten Kommunikation anhand von Modellen zu verstehen und zu lernen?
6. Sind Sie der Meinung dass die Entwicklung von Agilität und Kommunikation festgegossen ist, oder gibt es auch hier die Möglichkeit für stetiges Lernen?

4.4.2 Ergebnisse

Die Interviews ergaben eine Vielzahl von Übereinstimmungen der Meinungen, auch konnte man wirklich interessante Gedanken festhalten, welche ich im Anschluss präsentieren möchte

4.4.2.1 Ergebniss Frage 1

Wie wichtig finden Sie die Kommunikation mit anderen Mitarbeitern auch in punkto ihrer Transparenz, bzw. wie wichtig bewerten Sie positive oder negative Eindrücke bei neuen Kollegen?

Bei dieser Frage gab es eigentlich die einstimmige Meinung dass Kommunikation in unserem Beruf eines der Wichtigsten Grundlagen ist, hier möchte ich auf zwei Zitate zurückgreifen, welche sinnbildlich dafür waren. „Ohne Kommunikation würde unser Job gar nicht funktionieren“, und „Ich finde Kommunikation deshalb wichtig, weil es die Basis für eine erfolgreiche Zusammenarbeit ist“.

Man ist sich einig dass transparente, aufrichtige Kommunikation mit anderen Mitarbeitern die Grundlage für rasche, partizipative Problemlösungen bildet. Kommunikation ist ein Schlüssel für Transparenz, sie wird durch positive Eindrücke bei neuen Kollegen natürlich vereinfacht, wobei sich negative Eindrücke oft auch als falsch erweisen, wenn man einen Menschen erst besser kennenlernt. Falls nicht, sollte eine akzeptable Zusammenarbeit durch ein Mindestmaß an Professionalität dennoch möglich sein - vorausgesetzt jedoch, dass auch das Gegenüber diese aufbringt.

4.4.2.2 Ergebniss Frage 2

Wie beurteilen Sie Geduld und Toleranz und Unvoreingenommenheit im Umgang mit Ihren Kollegen, bzw. wie wichtig ist Ihnen der respektable Umgang?

Bei dieser Frage sind die Interviewteilnehmer der Meinung, dass man den Mitarbeitern ausreichend Zeit geben muss, sich mit ihrer Arbeit vertraut zu machen, jedoch sollte dabei auch den Lernfortschritt im Auge behalten werden. Sollte sich dieser nicht und nicht einstellen wollen und die gleichen Fehler wieder und wieder auftreten, kann die Geduld auch einmal zu Ende sein. Gerade dieser Punkt war vielen Befragten wichtig, dass sie Geduld aufbringen, aber auch diese dann recht schnell zu einem Ende kommen kann. Toleranz und Respekt sollten in der Arbeit selbstverständlich sein - unbewusste Voreingenommenheit wird sich zwar nicht gänzlich vermeiden lassen, doch sollte man sie soweit wie möglich reduzieren. Auch hier möchte ich eine Aussage erwähnen: „Der respektable Umgang untereinander ist meines Erachtens eine Grundvoraussetzung für eine erfolgreiche Zusammenarbeit und zwar unabhängig von der hierarchischen Ordnung.“

4.4.2.3 Ergebniss Frage 3

Wie bedeutend finden Sie exaktes Zuhören und Antworten sowie Geradlinigkeit bei Antworten in Gesprächen mit Kollegen?

Da bei Projekten generell ein hoher Zeitdruck herrscht, wird auch auf effiziente Kommunikation Wert gelegt. Daher sollte vor allem bei Meetings eine gute Kommunikationskultur herrschen, exaktes Zuhören/Antworten fördert die Kommunikation mit Kollegen. Geradlinige Antworten und auch Eingestehen von Unklarheiten sind essentiell, um einen Fehler durch dieses Unverständnis nicht weiter anwachsen zu lassen. Hier möchte ich folgendes als Kommentar erwähnen: „Geradlinigkeit hat meines Erachtens mehr mit der Persönlichkeit als mit der Sache zu tun.“ Oftmals wurde hier bemängelt dass wirklich exaktes Zuhören und Antworten sowie Geradlinigkeit nur von den wenigsten Menschen beherrscht wird, es hier immer ein grosses Verbesserungspotential gibt.

4.4.2.4 Ergebniss Frage 4

Wie wichtig finden Sie gefühlte Intuitionen in der technischen Agilität? Hat Ihr Bauchgefühl Einfluss auf Ihre Entscheidungen?

Bei dieser Frage gab es zweierlei Gruppen von Meinungen.

Erstens, dass bei technischen Projekten das Bauchgefühl keine optimale Basis für Entscheidungen ist. Hat ein Projektmitarbeiter aufgrund seiner Erfahrung eine Präferenz für bestimmte Lösungen, muss er die anderen Mitarbeiter auch sachlich von den Vorteilen überzeugen können.

Zweitens, Bauchgefühl kommt aus dem Unterbewusstsein und hat bei manchen Kollegen einen großen Einfluß auf Entscheidungen, da es sie in die Lage versetzt rasch zu entscheiden und dies zumeist richtig sei. Nach Maßgabe der verfügbaren Zeit muss diese Entscheidung durch Sachargumente unterlegt werden. Es wurde hier aber extra betont dass Bauchgefühl nur im Zusammenhang mit Erfahrung Einfluss nehmen kann. Als Beispiel möchte ich hier „natürlich

hat Bauchgefühl einen Einfluss auf Entscheidungen. Ich versuche dennoch Entscheidungen aufgrund von objektiv betrachteten Umständen zu treffen.“, anführen

4.4.2.5 Ergebniss Frage 5

Würden Sie es für wichtig und sinnvoll erachten Kommunikation anhand von Modellen zu verstehen und zu lernen?

Hier sind die Kollegen der Meinung, dass durch die praktische Arbeit im Team man am besten die Kommunikation und den Umgang mit Kollegen lernt, Modelle können aber zusätzliches theoretisches Wissen bringen. Natürlich ist man auch der Meinung dass Kommunikation immer verbesserbar ist. Wichtig ist den Kollegen auch den Leuten die Angst vor mehreren Personen zu sprechen, oder die durch sprachliche Barrieren entstehen kann, im Team zu nehmen. Als Beispiel möchte ich hier folgende Aussage anführen: „Modelle sind hilfreich für ein besseres Verständnis wissenschaftlich bekannter Zusammenhänge und um Abhängigkeiten reproduzierbar zu machen um zu optimieren. Neue Zusammenhänge wird man durch Modelle allerdings nicht entdecken.“

4.4.2.6 Ergebniss Frage 6

Sind Sie der Meinung dass die Entwicklung von Agilität und Kommunikation festgegossen ist, oder gibt es auch hier die Möglichkeit für stetiges Lernen?

In diesem Punkt gab es eine ganz starke Einstimmigkeit der Mitarbeiter. Agilität und Starrheit sind zwei nicht zu vereinbarende Begriffe. Agilität und agile Kommunikation haben sich daher ständig weiter zu entwickeln und den Gegebenheiten eines sich verändernden Alltags anzupassen, bedingt auch durch unterschiedliche Projekte, unterschiedliche Teams und unterschiedliche Länder. Hier habe ich folgende Aussage aussagekräftig gefunden: „Da sich

sowohl der Mensch mit seinem Wissen und Bewusstsein kontinuierlich verändert und die Umwelt ebenfalls, ist stetiges Lernen die Grundvoraussetzung um für die Zukunft gerüstet zu sein.“

5 Zusammenfassung

Die Erkenntnisse dieser Arbeit sind für das Unternehmen und mich persönlich sehr wichtig. Man sieht die Unterschiede in der Theorie und im Rahmen der Prozesse sehr deutlich und kann dies auf zukünftige Verhandlungen und Prozessdefinitionen mit Kunden einfließen lassen. Des Weiteren wurde auch der wichtige Gesichtspunkt der Teamzusammenstellung herauskristallisiert und besprochen, wie wichtig dies für den Erfolg eines Projektes sein kann. Nur ein Mitarbeiter der nicht mit dem Team zusammen passt kann bei Anwendung der Agilität ein grosses Problem für die Produktivität und Kommunikation sein. Auch die Informationen die meine Kollegen durch die Interviews eingebracht haben sind informativ und von grossem Wert. Natürlich werde ich mich auch nach dieser Arbeit weiter mit diesem Thema beschäftigen, da dies wirklich mein Interesse geweckt hat. Da auch unser Unternehmen in dieser Richtung sehr interessiert ist, wird auch die firmenbeteiligte Weiterentwicklung in punkto Agilität weiter vorangehen, und ich kann auch weiterhin versuchen mein erworbenes Wissen weiter einzubringen.

5.1 Persönliche Einsichten

Als ich bei meinem derzeitigen Arbeitgeber ins Handset Management wechselte, hatten wir viele definierte Prozessabläufe, die sich oftmals als zeitaufwendig gezeigt haben. Zeit, die bei der Einführung eines Mobiltelefones in den Markt sehr kostbar ist. Also machte ich mich über die Jahre und mithilfe meiner Erfahrung daran, diese Prozesse im Interesse der Kunden und meines Arbeitgebers stückweise zu optimieren. Ich wurde von Dr. Georg Wolf darauf hingewiesen, dass unsere über die Jahre gewachsene Arbeitsweise in meinem Team sehr stark der Agilität

entspricht, was mein Interesse für dieses Gebiet weckte. Ich wollte verstehen, warum unsere jetzige Arbeitsweise so viel effektiver war, warum die Kommunikation innerhalb des Teams und des Unternehmens so gut funktionierte, und entschloss mich meine Diplomarbeit über dieses Thema zu verfassen. Da auch das Unternehmen in dem ich beschäftigt bin grosses Interesse an Agilität hat, und diese Prozesstrukturen noch nicht festgehalten wurden, bekam ich auch Unterstützung seitens meines Arbeitgebers, was ich als Vollzeitarbeitskraft als sehr hilfreich empfand. Ich konnte mit diesem Thema auch Frau Professor Motschnig überzeugen und damit war mein Diplomarbeitsthema fixiert. Als Ziele wurden als ersten Teil der Arbeit eine theoretische Gegenüberstellung von klassischen und agilen Methoden festgelegt, um ein Grundverständnis in diesen beiden Richtungen zu schaffen, und mir noch weiteres Wissen in dieser Thematik aneignen konnte. Im Praxisteil war das Ziel diese über teils Jahre hinweg modifizierten und gewachsenen Prozesse festzuhalten, sei es in ihrer klassischen und agilen Form. Hierbei festzustellen warum dies als agil gesehen werden kann, was die Auswirkungen auf den Output und die Mitarbeiter für Auswirkungen hat, warum es dadurch zu Verbesserungen gekommen ist, als Ziel, die Prozesse detailliert festzuhalten, um auch weiteren Personen den Unterschied verdeutlichen zu können. Dieser war wirklich sehr zeitintensiv, da die Prozesse in dieser Art und Weise bisher nicht festgehalten wurde, ich auch immer wieder Rücksprache mit Kollegen halten musste ob sie bei der jeweiligen Ausführung die ihr Gebiet betreffe genauso sehen wie ich, was mich natürlich während der Fussballeuropameisterschaft in Österreich nach der Arbeit doch das eine oder andere Getränk kostete. Nach dem Abbilden dieser Prozesse wurde als Ziel definiert meine persönliche Erfahrung festzuhalten, Wissen, das ich mir im Rahmen der Arbeit in dem Unternehmen und der Diplomarbeit aneignen durfte, und ich als sinnvoll und wichtig erachte erwähnt zu werden, da dies ebenfalls Vorteile für weitere Teams oder Unternehmen bringen könnte. Zusätzlich sollte ich auch noch eine Befragung mit Kollegen durchführen, wie sie zu dem Thema Agilität und Kommunikation stehen, zwei Themen, die wie ich in dieser Arbeit lernte sehr eng miteinander verbunden sind. Hier mussten erstmals passende Fragen erstellt werden, was ich als nicht ganz einfach empfand um diese nicht suggestiven Einfluss nehmen zu lassen. Vorab wollte ich dies anhand einer Gruppendiskussion durchführen, jedoch entschied ich mich dann zu einer persönlichen Befragung um hier jede einzelne Meinung ohne möglicher Beeinflussung festhalten zu können. Einen weiteren wichtigen Punkt den ich während dem Schreiben der Arbeit festgestellt hatte, und der mich auch den einen oder anderen

Schweisstropfen kostete war, dass UML zwar zur Darstellung der obersten Ebene als Eventflussdiagramm brauchbar war, jedoch für diese detaillierte Prozessbeschreibung aufgrund der fehlenden Übersicht nicht geeignet ist, deshalb wählte ich sog. „swim lane diagrams“ als Darstellungsform.. Dies wurde auch in Anhang A festgehalten, da auch dies für mich eine sehr interessante Entdeckung während meiner Arbeit war.

6 Glossar

CVT Abkürzung für „Content Verification Test“: verschiedene Medien Typen werden auf ihre Funktionalität mit einem Handset getestet.

CBT Abkürzung für „Content Browsing Test“: Kundeninternes Team

Product Concept Sheet: Testrelevante Daten

Work Order Sheet: Technische Eckdaten eines Handsets

7 **Abbildungsindex**

Abbildung 1 Wasserfall Modell.....	9
Abbildung 2 Spiralmodell.....	11
Abbildung 3 V - Modell.....	13
Abbildung 4 Änderungskostenkurve: Klassische Softwareentwicklung	16
Abbildung 5 Kostenkurve: Agile Softwareentwicklung.....	22
Abbildung 6 Funktionsweise von Scrum.....	23
Abbildung 7 Drei unterschiedliche Rollen in Scrum.....	24
Abbildung 8 Sparen an der Qualität.....	32
Abbildung 9 Nachbesserungen schmälern den Gewinn.....	33
Abbildung 10 Prestigeprojekt	34
Abbildung 11 Ausweg agiler Projekte.....	35
Abbildung 12 Beziehung zwischen Endkunde – Mobilnetzbetreiber – Unternehmen inkl. Handset Management.....	37
Abbildung 13 Organisationseinheiten mit Tracking Tool Zugang	38
Abbildung 14 CVT Prozess Seite 1	39
Abbildung 15 CVT Prozess Seite 2	40
Abbildung 16 Handset Versand Probleme	48
Abbildung 17 Handset Versand Probleme Agil.....	51
Abbildung 18 Fehlende Dokumente im Handset Management Prozess	53
Abbildung 19 Fehlende Dokumente im Handset Management Prozess Agil.....	56
Abbildung 20 Download Fehler.....	59
Abbildung 21 Download Fehler Agil.....	64
Abbildung 22 Medien Typ Fehler.....	68

Abbildung 23 Medien Typ Fehler Agil.....	73
Abbildung 24 UML Darstellungsversuch.....	95
Abbildung 25 UML Darstellungsversuch.....	97
Abbildung 26 Abstrakte Darstellung in UML.....	99

„Ich habe mich bemüht, sämtliche Inhaber der Bildrechte ausfindig zu machen und ihre Zustimmung zur Verwendung der Bilder in dieser Arbeit eingeholt. Sollte dennoch eine Urheberrechtsverletzung bekannt werden, ersuche ich um Meldung bei mir.“

8 Literatur

Abrahamson Pekka, Salo Outi, Ronkainen Jussi, Warsta Juhani: Agile Software Development Methods - Review And Analysis, 2002, VTT Publications 478

Boehm Barry W.: A Spiral Model of Software Development and Enhancement, 1988, IEEE Computer Society Press

Bleek Wolf-Gideon, Floyd Christiane: Agile Methoden, 2004, Universität Hamburg

Bögli Alex: Die Rolle der Anforderungen in agilen Methoden, 2004, Universität Zürich

Cockburn Alistair: Agile Software Development, 2002, Addison Wesley

Cohn Mike: Agile Estimating and Planning, 2006, Prentice Hall

Gloger Boris: Scrum - Produkte zuverlässig und schnell entwickeln, 2008, Hanser Verlag

Highsmith Jim: Agile Software Development Ecosystems, 2002, Addison Wesley

Kauflin Michael: Agile vs. Klassische Entwicklungsmethoden, 2004, Universität Zürich

Pichler Roman: Scrum: Agiles Projektmanagement erfolgreich einsetzen, 2008, dPunkt

Schwaber Ken, Beedle Mike: Agile Software Development with Scrum, 2001, Prentice Hall

Schwaber Ken: Agile Project Management with Scrum, 2004, Microsoft Press

Schwaber Ken: "The Enterprise and Scrum", 2007, Microsoft Press

Takeuchi Hirotaka, Nonaka Ikujiro: The New Production Development Game, 1986, Harvard Business Review Jan/Feb.: 137-146

Fowler Martin: The New Methodology, 2005,
<http://martinfowler.com/articles/newMethodology.html>

Ambler Scott W., Kostenkurven, 2002
<http://www.agilemodeling.com/essays/costOfChange.htm>

Grote Wolfhart, Agile Softwareentwicklung nach Maß, Juni-August 2008,
http://www.infolab.de/presse/pressespiegel/t3n_scrum_2008_06_01.pdf

Royce Winston: Managing the Development of Large Software Systems, 1970
<http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>

Agile Softwareentwicklung, Wikipedia, 2008
http://de.wikipedia.org/wiki/Agile_Softwareentwicklung

Wasserfallmodell, Wikipedia, 2008
<http://de.wikipedia.org/wiki/Wasserfallmodell>

Spiralmodell, Wikipedia, 2008
<http://de.wikipedia.org/wiki/Spiralmodell>

V-Modell, Wikipedia, 2008
<http://de.wikipedia.org/wiki/V-Modell>

Scrum, Wikipedia, 2008
<http://de.wikipedia.org/wiki/Scrum>

Agiles Projektmanagement

<http://scrum-master.de>

Das Agile Manifest, 2008

<http://www.agilemanifesto.org>

Anhang A Darstellung in UML

Vorab gab es einige Versuche den Prozess in UML darzustellen, um auch die schon definierte Struktur zu verwenden. Leider wurden aber hier die Einschränkungen bei der praktischen Nutzung deutlich. Das Verwenden dieser Prozessmodellierungsmethode ist für Präsentationszwecke in einem Betrieb mit mehr als nur einfachen Abläufen leider nicht geeignet (vgl. Abbildung 24).

In unseren Prozessen wird mit vielen Dokumenten gearbeitet, die nach diversen Aktionen auch ein Update erhalten. Diese Dokumentflüsse machen eine Darstellung in UML, auch ohne Synchronisierungen, sehr unübersichtlich. Auch das Verwenden interner und externer Tracking Tools wurde bei der Entwicklung von UML nicht berücksichtigt. Deshalb haben wir eine allgemeine Form der Darstellung mit unseren Kunden gewählt, um soviel Übersichtlichkeit wie möglich zu bieten. Es soll auch für außenstehende Personen ein Leichtes sein, die Prozesse zu lesen und zu verstehen.

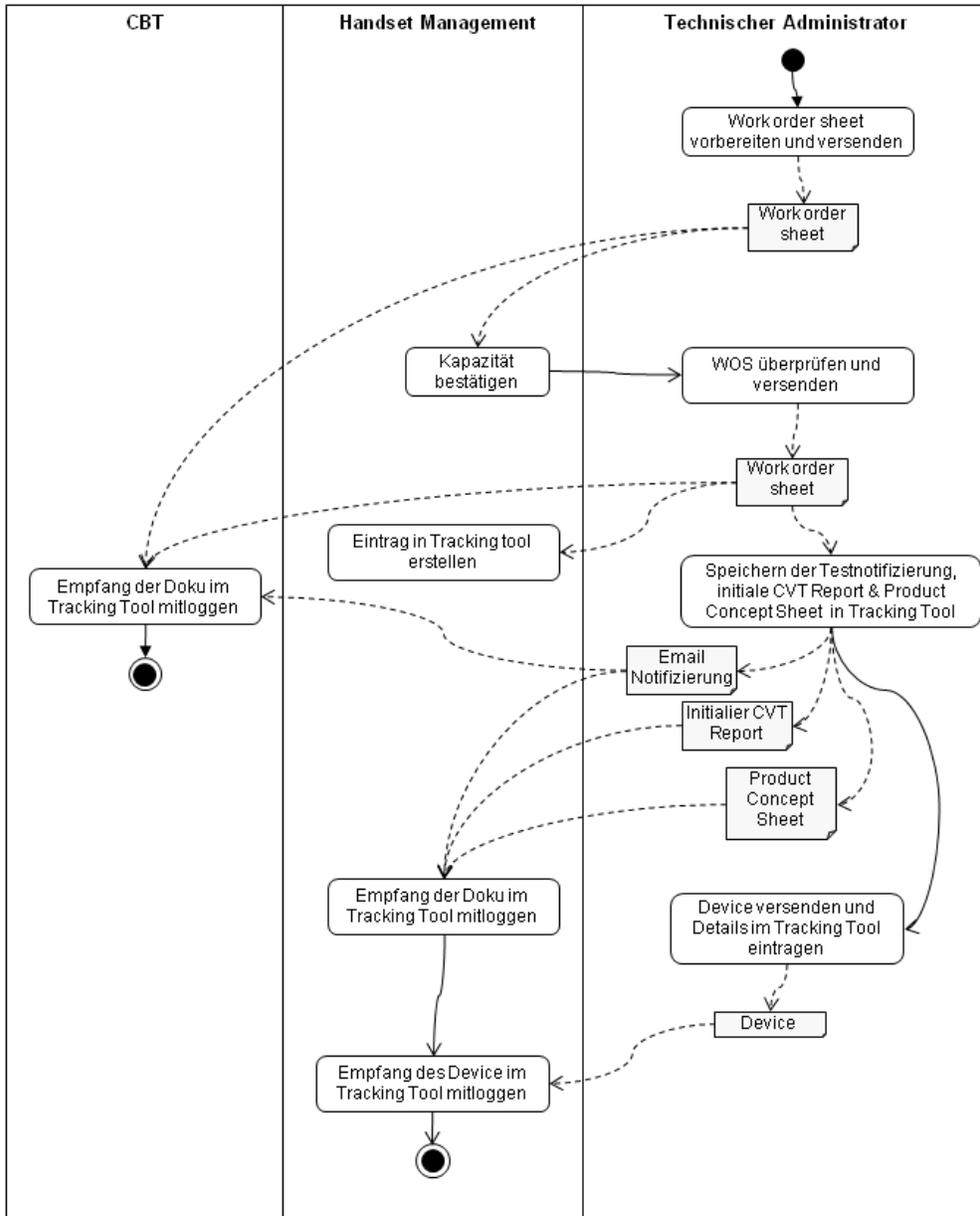


Abbildung 24 UML Darstellungsversuch

Mit Synchronisierungen wird dies nicht nur um einiges unübersichtlicher, sondern auch unleserlicher. Für eine Darstellung die erklärend und beschreibend wirken soll ist dies leider im Kundenbetrieb und zum Bereitstellen von Wissen unbrauchbar. Ebenfalls stellt das Nebenläufige und Aufspaltende der Dokument- und Kontrollflüsse bei diesem Prozess aus meiner Sicht einen weiteren Grund der Unverwendbarkeit dar. Aus der fertigen UML Darstellung könnte man keiner Person, die mit dem Prozess nicht sehr gut vertraut ist, diese Interaktion mit dem Kunden näher bringen. Eine strenge Aktivitätsdiagrammsemantik ist hier nicht sinnvoll einzusetzen.

Als Beispiel (vgl. Abbildung 25) möchte ich hier einen Versuch nach strenger UML Vorgabe anführen

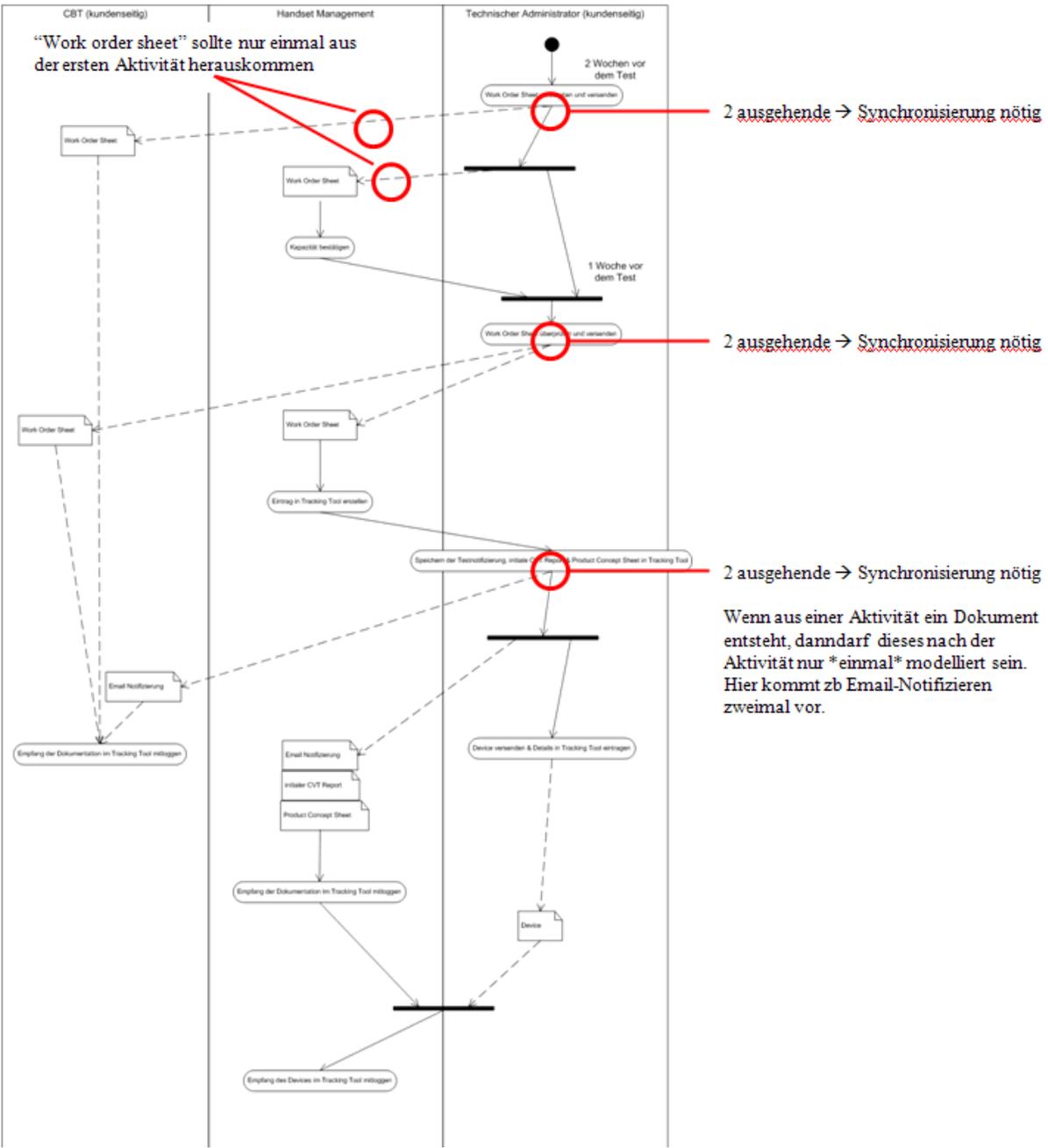


Abbildung 25 UML Darstellungsversuch

Wie man hier erkennen kann, ist schon der Ansatz der genauen UML Verwendung aufgrund der öfters verwendeten Dokumente, Rollen und Tracking Tool Einträge betreffend der Übersichtlichkeit nicht zielführend und für eine genaue Abbildung dieser Art von Prozessen

nicht geeignet. Jedoch kann man UML natürlich für eine etwas abstrahiertere Ansicht verwenden, wie in 2.1.3.1 gezeigt wird.

Für mich persönlich stellt sich UML deshalb zwar als sehr gute Möglichkeit der Prozessdarstellung dar, jedoch war sie nie als Kooperationsmodul gedacht. Die Menge an Wissen, welche ich für diesen Prozess verwendet habe macht die Darstellung in UML unübersichtlich und nicht nachvollziehbar. Versions- und Zeitmanagement wurde in UML nicht vorgesehen, ebenso wenig wie versionierte Dokumente zwischen Personen und Abteilungen oder Tracking Tools.

Abstrakte Darstellung in UML

Eine abstrakte Darstellung war jedoch möglich (vgl. Abbildung 26).

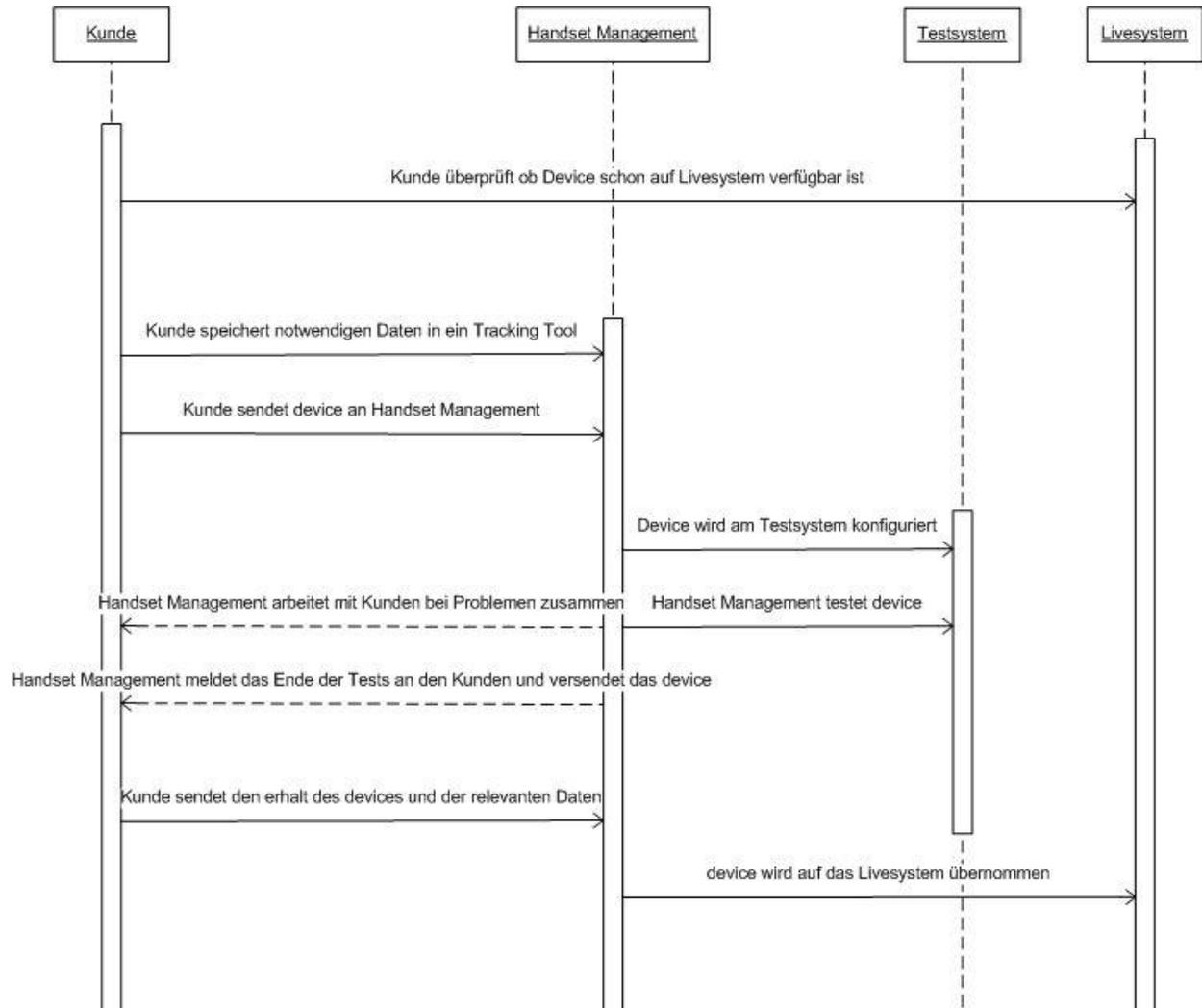


Abbildung 26 Abstrakte Darstellung in UML

Anhang B Abstract & Lebenslauf

Abstract

Die wichtigsten Ziele der Diplomarbeit sind die theoretische Gegenüberstellungen vom klassischen gegenüber dem agilen Methodenansatz der Softwareentwicklung mittels Literaturanalyse, die Aufbereitung von wichtigen Handset Management Prozessen unter diesen Gesichtspunkten, das herausfiltern der gelebten Agilität in diesem Bereich. Da das Unternehmen agile Arbeitsweisen in der Entwicklung übernommen hat, soll auch die gewachsene Agilität im Handset Management verdeutlicht und dokumentiert werden. Auch hier wird versucht den Vorteil des agilen Vorgehens zu verdeutlichen.

Bei der theoretischen Gegenüberstellung werden diese Methoden als Grundlage vorgestellt. Bei der Aufbereitung und Abbildung der Handset Management Prozesse wird der klassische mit dem jeweiligen agilen Ansatz verglichen und genau erklärt, um dem Leser die spezifischen Vorteile der Abteilung Handset Management näher bringen zu können. Daraus resultierend werden die Erfahrungen in diesem Bereich dargestellt, und wie weit diese Auswirkungen auf andere Prozesse wie zum Beispiel Kommunikation oder das Einstellen von neuen Mitarbeitern haben. Danach wird durch ein teilstrukturiertes Interview die Meinung und Einstellung von Kollegen im Unternehmen festgehalten, breit gefächert nach Aufgabenbereich und Position.

The most important goals of the Masters Thesis are the theoretical comparison of classic and agile techniques by literature analysis, the preparation of important Handset Management processes considering those principles, and the specification of the process of extracting the relevant agile information. As the company adopted agile operation in developing software, the grown agility should be traced and documented in the Handset Management area. I also try to capture the advantages of the agile approach.

During the theoretical comparison the methods are presented for background knowledge about this topic. Preparing the Handset Management Processes, I compare and describe in detail the classic with the particular agile approach, showing the reader the specific advantages of the

Handset Management department. As a result, the experience in this area and its impact towards other processes like communication or employing new staff members will be presented. Thereafter, partially structured interviews record the opinion and attitude of colleagues in different functions and positions inside the company.

Lebenslauf

Name: **Oliver Maurovich**

Geburtsdatum: 10.01.1977

Nationalität: Österreich

Bildung: **September 1983 – Juni 1987**
Volksschule Karl-Toldt Weg, 1140 Wien

September 1987 – Juni 1995
Bundesrealgymnasium Wien 14

Seit Oktober 1996
Studium Wirtschaftsinformatik, Universität Wien und Technische
Universität Wien

Berufliche Entwicklung: **Juni 1999 – November 2000**
uboot.com mobile internet services gmbh
Technischer Event Manager

September 2001 – Jänner 2005
ucp morgen (universal communication platform AG)
Produkt- und Projektmanager

Jänner 2005 – Mai 2006
Qpass
Handset Manager

Seit Mai 2006

Qpass – Amdocs Digital Commerce Division

Manager Handset Testing

Seit Jänner 2000

Eventplattform partyteufel.at

CEO

Sprachen:

Englisch, Französisch