

Entwurf eines agilen Requirements Engineering Vorgehens mit Fokus auf Gebrauchstauglichkeit für das Allianz Business System

Bachelorarbeit

Eingereicht von: Andreas Fay

Matrikelnummer 2680515
E-Mail a.fay@gmx.de

*Betreuer: Prof. Dr. Barbara Paech (Universität Heidelberg)
Christoph Weiß (Allianz Deutschland AG)*

Heidelberg, den 30. Juni 2010



Kurzbeschreibung

Um einen agilen Requirements Engineering-Prozess mit Fokus auf Gebrauchstauglichkeit für das Allianz Business System zu etablieren, erläutert diese Arbeit zunächst den IST-Zustand. Dabei sticht vor allem die ungenügende Einbindung von Benutzerinnen und Benutzern hervor. Bisher werden diese erst am Ende des Prozesses eingebunden. Um dies zu beheben, werden zwei bekannte Ansätze aus der Literatur näher betrachtet und deren „Best Practices“ vereint. Auch für die Integration von klassischem Requirements Engineering und agilen Methoden werden aus der Literatur die Praktiken extrahiert. Diese beiden Sammlungen an Praktiken sind die Grundlage für die Behebung der identifizierten Defizite, um das Ziel eines agilen Requirements Engineering-Prozesses mit Fokus auf Gebrauchstauglichkeit für das Allianz Business System zu erreichen.

Abstract

As a prerequisite for establishing an agile requirements engineering-process with a focus on usability for the Allianz Business System, this paper first investigates the current situation. The main problem found is the insufficient user-involvement. Users are only involved at the end of the process. To solve this, two well-known approaches from literature are contemplated and their best practices are merged. Similar to this, best practices for the integration of traditional requirements engineering and agile methods are extracted from literature as well. Both of these collections of best practices serve as a basis for rectifying the identified deficiencies, to achieve the goal of the agile requirements engineering-process with a focus on usability for the Allianz Business System.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Hintergrund und Ziel dieser Arbeit	1
1.2	Gliederung der Arbeit.....	1
2	Grundlagen.....	3
2.1	Software Process Improvement.....	3
2.1.1	Konzept.....	3
2.1.2	Capability Maturity Model Integration (CMMI)	4
2.1.3	Goal-Question-Metric (GQM).....	5
2.1.4	Zusammenfassung	6
2.2	Requirements Engineering und agile Methoden.....	6
2.2.1	Requirements Engineering	6
2.2.2	Agile Methoden.....	8
2.2.3	Integration von RE und agilen Methoden	10
2.2.4	Zusammenfassung	14
2.3	Gebrauchstauglichkeit	15
2.3.1	Lauesen	17
2.3.2	Geis	21
2.3.3	Gemeinsamkeiten und Unterschiede.....	24
2.3.4	Zusammenfassung	25
3	IST-Zustand bei Allianz.....	26
3.1	RE-Prozess.....	26
3.1.1	ABS Vorgehensmodell.....	26
3.1.2	Rollen.....	28
3.1.3	Werkzeuge.....	29
3.1.4	Dokumente	31
3.1.5	Erläuterung des Prozesses.....	32
3.2	Bewertung.....	37
3.2.1	GQM-Plan.....	38
3.2.2	Kriterien	39
3.2.3	Messung	40
3.2.4	Schlussfolgerungen	43
3.3	Agile Methoden	44
3.3.1	Anpassungen der Prozesse durch Agile.....	45
4	Vorschlag für SOLL-RE-Prozess	46
4.1	Integration von RE und agilen Methoden.....	46
4.2	Integration der Methoden von Lauesen und Geis.....	48
4.3	RE-Prozess.....	50
4.3.1	Rollen.....	50
4.3.2	Werkzeuge.....	51
4.3.3	Dokumente	52

4.3.4	Erläuterungen des Prozesses.....	52
5	Fallstudie	58
5.1	Bisherige Dokumente.....	58
5.2	Neuerungen	61
6	Zusammenfassung & Ausblick.....	65
A	Glossar.....	67
B	Abbildungsverzeichnis.....	70
C	Tabellenverzeichnis.....	71
D	Abkürzungsverzeichnis.....	72
E	Literaturverzeichnis.....	73
F	Anhang.....	76
F.1	Adonis-Legende.....	76
F.2	IST-RE-Prozess in Adonis	78
F.2.1	Hauptprozess	78
F.2.2	Unterprozess „IST-Prozesse analysieren und Anforderungen an diese erheben	81
F.2.3	Unterprozess „Basisanforderungen ausarbeiten“.....	82
F.2.4	Unterprozess „Anforderungen überprüfen“	83
F.2.5	Unterprozess „Anforderungen implementieren“	84
F.3	SOLL-RE-Prozess in Adonis.....	86
F.3.1	Hauptprozess	86
F.3.2	Unterprozess „IST-Prozesse analysieren und Anforderungen an diese erheben“	90
F.3.3	Unterprozess „Basisanforderungen ausarbeiten und Nutzungsanforderungen erfassen“	91
F.3.4	Unterprozess „Anforderungen überprüfen“	92
F.3.5	Unterprozess „Anforderungen implementieren“	93
F.4	Fragen & Antworten aus Interviews mit Allianz-Mitarbeiterinnen und -Mitarbeitern.	96
F.5	Kriterienkatalog	100
G	Ehrenwörtliche Erklärung	102

1 Einleitung

1.1 Hintergrund und Ziel dieser Arbeit

Sowohl Kundinnen und Kunden, als auch Benutzerinnen und Benutzer einer Software zufriedenzustellen, ist in modernen Unternehmen eines der Hauptziele und einer der wichtigsten Erfolgsfaktoren. Dies gilt natürlich auch dann, wenn die Kunden und Benutzer Mitarbeiterinnen und Mitarbeiter des Unternehmens sind, das auch die Software herstellt. Im Fall von Allianz mit dem Allianz Business System (ABS) ist genau das der Fall. Das ABS zielt darauf ab, die Arbeit der Sachbearbeiterinnen und Sachbearbeiter der Versicherung zu erleichtern, indem es alle bisherigen Bestandsysteme der verschiedenen Versicherungszweige in einem System zusammenfasst. Die enorme Komplexität der verschiedenen Systeme und auch deren große Anzahl macht deren Vereinigung zu einer schwierigen Aufgabe. Auch länderspezifische Einzelheiten müssen im ABS berücksichtigt werden, denn nicht jeder Versicherungsmarkt gleicht dem anderen. Die deutsche Allianz (Allianz Deutschland AG oder kurz ADAG) hat zur Anpassung des ABS an die deutschen Gegebenheiten das sogenannte ABS Vorgehensmodell entwickelt, das die Erweiterung der im ABS bereits vorhandenen Arbeitsabläufe systematisch beschreibt. Dieses Modell basiert auf dem V-Modell XT, einem einschlägigen Softwareentwicklungsmodell. Um unter anderem dem anfangs erwähnten Ziel der Zufriedenstellung der Kundinnen und Kunden sowie Benutzerinnen und Benutzer Rechnung zu tragen, wurde (und wird derzeit noch) das ABS Vorgehensmodell in den Entwicklungsphasen auf ein agiles Vorgehen umgestellt. Diese Arbeit setzt im ABS Vorgehensmodell noch weiter vorne, bei der Phase des Requirements Engineering, an und hat zum Ziel auch diese agil zu gestalten und dazu auch Aspekte der Gebrauchstauglichkeit zu berücksichtigen. Letzteres zielt noch mehr als die „Agilisierung“ auf die Erhöhung der Zufriedenheit der Benutzerinnen und Benutzer ab, welche mit dem ABS arbeiten müssen. Denn im Requirements Engineering werden die Grundsteine für den späteren Entwurf und die Umsetzung von Software gelegt und daher ist es in dieser Phase enorm wichtig Gebrauchstauglichkeit nicht zu vernachlässigen, um am Ende auch ein Produkt zu erhalten, welches die Benutzerinnen und Benutzer zufriedenstellt.

1.2 Gliederung der Arbeit

Um zunächst wichtige Voraussetzungen für die Betrachtungen in dieser Arbeit sicherzustellen, wird in Kapitel 2 ausführlich über Grundlagen gesprochen. Dazu gehört zu allererst Theorie zur Prozessoptimierung im Umfeld von Software allgemein. Es wird behandelt, wie man Prozessoptimierung praktisch anwenden kann und letztlich auch in dieser Arbeit angewendet wird. Im Folgenden werden die Vokabeln Requirements Engineering und agile Methoden erläutert. Bei Requirements Engineering werden die typischen Aspekte beleuchtet, im Fall der agilen Methoden werden die für diese Arbeit relevanten Methoden vorgestellt. Anschließend wird betrachtet, wie man Requirements Engineering und agile Methoden miteinander vereinen kann und welche Problemfelder dabei zu berücksichtigen sind. Als Ergebnis wird eine Liste von Praktiken präsentiert, die ein Requirements Engineering-Prozess allgemein unterstützen sollte, um sowohl agil zu sein, aber auch die dadurch entstehenden Mängel im Requirements Engineering abzudecken. Kapitel 2 betrachtet als dritten und letzten Punkt die Gebrauchstauglichkeit. Dazu werden zunächst zugrunde liegende Normen behandelt. In der Folge werden dann die Vorgehensmodelle von Soren Lauesen und Thomas Geis zur Schaffung eines gebrauchstauglichen interaktiven Softwaresys-

tems näher betrachtet. Als Ergebnis wird neben Gemeinsamkeiten und Unterschiede der beiden Ansätze eine Liste von Praktiken präsentiert, die ein Requirements Engineering-Prozess allgemein unterstützen sollte, um Aspekte der Gebrauchstauglichkeit zu berücksichtigen.

Kapitel 3 beschäftigt sich mit dem jetzigen IST-Zustand bei Allianz. Das umfasst zunächst eine Beschreibung des ABS Vorgehensmodells und wo genau der Requirements Engineering-Prozess darin zu finden ist. Anschließend folgt eine genaue Beschreibung des Requirements Engineering-Prozesses selbst – das schließt die darin vorkommenden Rollen, Systeme, Dokumente und natürlich auch die Abfolge der Aktivitäten ein. Die Beschreibungen des Prozesses basieren auf einem Prozessmodell, welches im Rahmen dieses Kapitels zunächst erstellt und im Anhang beigefügt wurde. Basierend auf diesem Prozess wird eine Bewertung des IST-Zustands durch mit Allianz abgestimmten Kriterien vorgestellt, die dazu dient feststellen zu können, an welcher Stelle Handlungsbedarf besteht und welche Teile wie stark optimiert werden können. Zum Abschluss des Kapitels wird beschrieben, wie agil bei Allianz verstanden wird und wo und weshalb agile Methoden bereits zum Einsatz kommen.

In Kapitel 4 wird zu den Aspekten von Requirements Engineering und agilen Methoden, wie auch zu Gebrauchstauglichkeit anhand der in Kapitel 2 identifizierten Praktiken eine bestmögliche Lösung für Allianz diskutiert. Dazu werden die in Kapitel 3 identifizierten Schwachstellen des jetzigen IST-Zustandes versucht zu beheben, indem genauer darauf eingegangen wird, welche Änderungen am Requirements Engineering-Prozess durchgeführt werden sollten. Dies wird anschließend in einem Prozess-Modell visualisiert und im Detail beschrieben.

Der in Kapitel 4 ermittelte SOLL-Requirements Engineering-Prozess wird in Kapitel 5 in einer Fallstudie untersucht. Dazu werden von einem bei Allianz durchgeführten Projekt zunächst beispielhaft einige Dokumente des IST-RE-Prozesses vorgestellt. Daraufhin wird illustriert, wie sich die im SOLL-RE-Prozess neuen Dokumente und Aktivitäten dazu einreihen.

Kapitel 6 fasst schließlich die Erkenntnisse dieser Arbeit und die Vorteile des vorgeschlagenen SOLL-RE-Prozesses zusammen und gibt einen kurzen Ausblick über das mögliche weitere Vorgehen.

2 Grundlagen

Dieses Kapitel befasst sich mit den für diese Arbeit relevanten Grundlagen. Dazu wird in Unterkapitel 2.1 das Konzept von Software Process Improvement zusammen mit relevanten Modellen und Frameworks zu diesem Thema vorgestellt. Unterkapitel 2.2 erläutert genauer, was man unter Requirements Engineering und agile Methoden versteht und welche Problemfelder bei einer Integration der beiden zu behandeln sind. In Abschnitt 2.3 werden die Grundlagen zu Gebrauchstauglichkeit erläutert.

2.1 Software Process Improvement

Im Rahmen der Überarbeitung eines vorhandenen Prozesses zur Entwicklung eines SOLL-Prozesses, wie es in dieser Arbeit der Fall ist, stellt die Theorie von Software Process Improvement (kurz SPI) eine fundamentale Grundlage dar. SPI vermittelt Techniken und Methoden zur systematischen Verbesserung eines Prozesses. Im Folgenden wird, basierend auf [1] und [2], zunächst das Konzept von SPI vermittelt (Abschnitt 2.1.1), anschließend den für Allianz interessanten Standard CMMI (Abschnitt 2.1.2) und schließlich in Abschnitt 2.1.3 mit Goal-Question-Metric (kurz GQM) das Vorgehensmodell, an welchem sich diese Arbeit orientiert, genauer erläutert.

2.1.1 Konzept

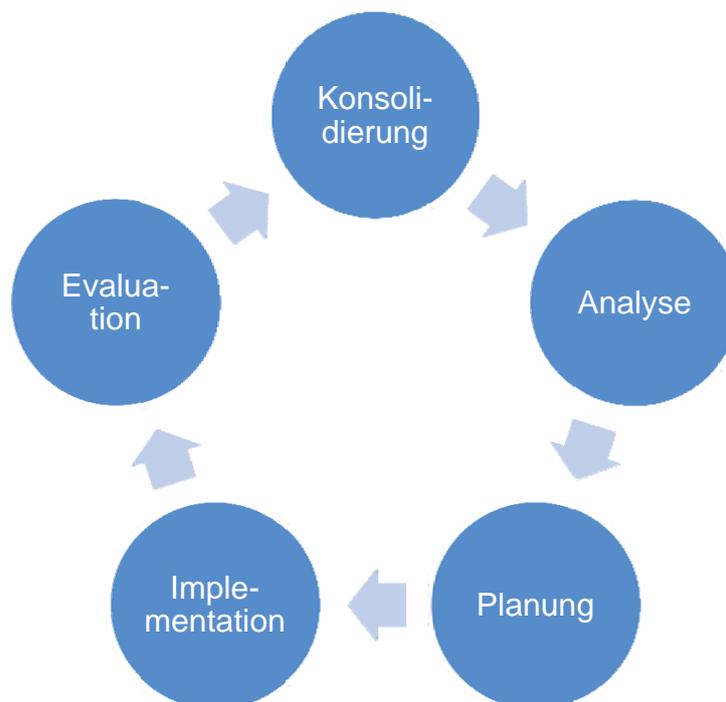


Abb. 1: Konzeptueller Prozess für SPI

Dem Konzept hinter SPI liegt ein fünfteiliger zyklischer Prozess zugrunde (Abb. 1). Bei der, idealerweise ersten, Konsolidierungsphase soll der Prozess, bevor er verbessert werden kann, erst einmal erfasst und dokumentiert werden. Falls der Prozess so noch gar nicht gelebt wird, sollen zunächst Maßnahmen getroffen werden, um dies zu ändern (z.B. Schulungen). Die Optimierung beginnt dann mit einer Analyse, d.h. eine explizite Messung der bisherigen Prozessleistung und gleichzeitig eine Identifikation von dabei auftretenden Mängeln. In der darauffolgenden Planung wer-

den Konzepte ausgearbeitet, die versuchen sollen gefundene Mängel zu beheben und/oder Verbesserungspotentiale auszuschöpfen. Während der Implementation sollen die zuvor erarbeiteten Pläne umgesetzt und schließlich in der Evaluation auf Erfolg geprüft werden. Die Implementation hat dabei zumeist eine Erhöhung der Unordnung zur Folge, weshalb nach deren Evaluation eine erneute Konsolidierung empfohlen wird.

2.1.2 Capability Maturity Model Integration (CMMI)

CMMI ist ein Modell-basierter Standard zur Bewertung eines Softwareprozesses gemäß einer Reifegradskala, entwickelt vom Software Engineering Institute (SEI) [2]. Die Reifegrade sind aufeinander aufbauend wie folgt definiert:

1. Initial
2. Managed
3. Defined
4. Quantitatively Managed
5. Optimizing

Diese Einordnung erfolgt anhand der Kriterien (Practices), welche für jede Ebene festgelegt sind. Um ganz konkret einen Softwareprozess in eine der Ebenen einzuordnen, werden zunächst die Kriterien der zweiten Stufe auf Erfüllung geprüft. Werden diese nicht erreicht, wird der Softwareprozess in die erste Stufe eingeordnet, da diese ohne Beschränkung erreichbar ist. Ansonsten wird geprüft, ob der Softwareprozess den Kriterien der dritten Stufe genügt usw. Je höher ein Softwareprozess also eingeordnet werden kann, desto reifer ist er.

Die eigentliche Verbesserung ist implizit gegeben, denn um eine Ebene im Reifegrad zu steigen, sind die damit verbundenen Kriterien, die dem Softwareprozess eine höhere Qualität beimessen, zu erfüllen.

CMMI wurde basierend auf dem Vorgänger Capability Maturity Model (CMM) und in Anlehnung an SPICE (siehe unten) entwickelt. Dies wurde notwendig, da es für viele Unternehmen nicht mehr möglich war, mit einem Standard alle Probleme zu behandeln. So wurden für CMM Speziallösungen entwickelt, die den Bedürfnissen der Unternehmen gerecht wurden, z.B. SW-CMM (CMM für Software) oder IPD-CMM (CMM für integrierte Produktentwicklung). Erst CMMI versuchte diese Speziallösungen wieder zu vereinen und zu einem gemeinsamen Modell zu integrieren. CMMI ist etwas dynamischer als das Vorgängermodell. Es bietet neben der klassischen Stufenbasierten Sichtweise auch eine sogenannte „Continuous Representation“. Diese zielt darauf ab einzelne Kernbereiche direkt verbessern zu können. Beide Sichtweisen gehen aber von denselben Kernpunkten aus und legen mit einer Priorisierung derer nahe, welche Bereiche in welcher Reihenfolge verbessert werden sollten.

Die Allianz verwendet CMMI als Referenz bei der Qualitätssicherung der Softwareentwicklungsprozesse.

Software Process Improvement and Capability dEtermination (SPICE)

Die als SPICE bezeichnete Norm ISO/IEC 15504 ist ein drei-Phasen-Modell zu SPI [1]. Die erste umfasst die Prozessbeurteilung, die zweite Prozessverbesserung und die dritte die Ermittlung des Prozesspotentials. SPICE wurde in Anlehnung an CMM entwickelt, ist aber besser mit CMMI vergleichbar, da sich CMMI an SPICE orientiert.

Im Gegensatz zu CMMI ist SPICE ausschließlich als „Continuous Representation“ anzusehen, da es eine sehr feingranulare Aufteilung der Kernbereiche in Softwareprozessen vorgibt, die sich an ISO 12207 anlehnt [3]. Für jeden dieser Bereiche wird ähnlich wie in CMMI ein Reifegrad vergeben. Die möglichen Reifegrade sind:

1. Performed
2. Managed
3. Established
4. Predictable
5. Optimising

In SPICE existiert ein sechster Reifegrad, der 0., da, anders als bei CMMI, für Stufe 1 bereits Bedingungen gestellt werden [3].

Für die Verbesserung eines Prozesses ist es, wie in CMMI, notwendig insofern den Prozess zu ändern, als dass er die Auflagen der nächst höheren Stufe erfüllt. Die dafür notwendigen Bewertungen müssen von einem externen Experten erfolgen [2].

Reifegrad von Requirements Engineering Prozessen

Sommerville und Sawyer stellen in [4] ein auf CMM basierendes Reifegradmodell von Requirements Engineering-Prozessen vor. Dazu haben sie die in CMM definierten Reifegrade teilweise neu definiert.

Der erste Reifegrad, „Initial“, trifft auf Organisationen zu, die keinen definierten Requirements Engineering Prozess haben. Sie haben Probleme damit gute Anforderungsdokumente innerhalb gegebener Zeit- und Kostenrahmen zu erstellen. Sie sind sehr abhängig von fähigen und erfahrenen Mitarbeitern.

Die zweite Stufe, „Repeatable“, betrifft Unternehmen, die Standards für ihre Anforderungsdokumente definiert und ein Vorgehen zum Anforderungsmanagement eingeführt haben. Betreffende Unternehmen, die diese Stufe erfüllen, haben meistens ein gleichmäßig qualitativ hochwertiges Anforderungsdokument, welches entsprechend der Planung (Kosten und Zeit) erstellt wurde.

Die dritte Stufe, „Defined“, wird durch einen explizit definierten RE-Prozess beschrieben, der geprägt von bewährten Techniken und Praktiken ist. Zusätzlich gibt es eine kontinuierliche Prozessoptimierung.

2.1.3 Goal-Question-Metric (GQM)

Der GQM-Ansatz nach Basili [5] dient der zielorientierten Messung von Softwareprodukten oder -prozessen. Anders als die zuvor skizzierten Frameworks ist GQM nicht Modell-basiert, sondern lässt eine relativ freie Messung zu, wobei das zugrunde liegende Modell zwischen drei Ebenen unterscheidet:

1. Conceptual
2. Operational
3. Quantitative

Die erste Ebene ist die der Ziele (Goal). Hier werden der Zweck der Messung, das zu untersuchende Objekt, das genaue Problem der Messung und die Sicht, aus der die Messung vorgenommen wird, festgehalten. Diese kurzen, schwer messbaren, Ziele werden in der zweiten Ebene, den Fragen (Question), präzisiert. Anhand der Fragen soll es möglich sein bestimmen zu können, ob das/die zugehörige/n Ziel/e erreicht

wurde/n. Um dies genau feststellen zu können, werden die Fragen in Metriken (Metric) in Ebene drei verfeinert.
Eine ausführliche Beschreibung gibt [6].

2.1.4 Zusammenfassung

Für die Verbesserung des Requirements Engineering-Prozesses bei Allianz wird GQM eingesetzt, da es im Vergleich zu CMMI (oder ähnlichen Modellen) feingranularer ist, d.h. präziser formulieren kann, wie genau ein Prozess bemessen wird und so leichter Schlussfolgerungen gezogen werden können. CMMI stellt dennoch eine wichtige Grundlage dar, da es für GQM Ziele und Fragen liefert, die auch damit zusammenhängen, dass Allianz CMMI als Referenz für die Qualitätssicherung von Softwareprozessen verwendet.

2.2 Requirements Engineering und agile Methoden

Neben den Grundlagen zur Verbesserung von Prozessen bedarf diese Arbeit einiger Grundlagen zu Requirements Engineering und agilen Methoden. Im Folgenden werden die Kernpunkte sowohl von klassischem Requirements Engineering (Kapitel 2.2.1), als auch die von agilen Methoden (Kapitel 2.2.2) herausgestellt. Nach dieser getrennten Vorbetrachtung werden in Kapitel 2.2.3 beide Konzepte zusammen betrachtet. Dabei steht die Frage nach dem derzeitigen Stand der Forschung im Mittelpunkt.

2.2.1 Requirements Engineering

Requirements Engineering (RE) ist ein kooperativer, iterativer, inkrementeller Prozess, der sich mit der Identifikation und Dokumentation von Anforderungen für ein Softwaresystem und dessen Kontext beschäftigt. Es ist das Ziel von RE Anforderungen vor der eigentlichen Entwicklung des Softwaresystems zu erfassen und damit spätere, kostenintensive Anpassungen zu verhindern. RE basiert dabei auf zwei wichtigen Annahmen [7; 8]:

- Je später im Softwareentwicklungsprozess Fehler gefunden werden, desto kostspieliger ist deren Behebung [8].
- Es ist möglich eine stabile Menge von Anforderungen vor dem nächsten Inkrement des Systementwurfs und der Implementation zu bestimmen [7].

In RE werden vier große Aktivitäten ausgeführt [8]: Erhebung, Dokumentation, Prüfung und Abstimmung sowie Verwaltung von Anforderungen. Im Folgenden wird jede dieser Aktivitäten basierend auf [8] kurz erläutert.

2.2.1.1 Erhebung von Anforderungen

Die Erhebung von Anforderung ist die Grundlage für das RE. Hierbei handelt es sich um den Versuch ein möglichst vollständiges und fehlerfreies Verständnis des Softwaresystems und von dessen Kontext zu erhalten und in Form von Anforderungen zu beschreiben. Dazu werden zumeist Stakeholder, Dokumente und bereits in Betrieb stehende Alt- oder Konkurrenzsysteme hinzugezogen. Dabei sehr beliebte Mittel sind Interviews oder Beobachtungen von Stakeholdern oder Perspektiven-basiertes Lesen von Dokumenten.

Anforderungen werden für gewöhnlich zu Beginn eines Projekts, also auch vor dem Entwurf oder der Implementation, erhoben, da sie die Grundlage für jene bilden. Aus diesem Grund ist die Erhebung der Anforderungen ein besonders erfolgskritischer Faktor. Werden in diesem Stadium Fehler begangen, so kann der Aufwand zur Korrektur derselben in einer späteren Projektphase bis zu 100-mal höher sein [8]. Gerade wegen dieser enormen Verantwortung ist und bleibt RE zusammen mit der Erhebung von Anforderungen eine für den Projekterfolg unverzichtbare Tätigkeit.

2.2.1.2 Dokumentation von Anforderungen

Das Ziel der Dokumentation der Anforderungen ist eine systematisch organisierte Kommunikation zwischen den Stakeholdern. Die Dokumentation dient für gewöhnlich als rechtliche Grundlage zwischen Auftraggeber und -nehmer. Aber auch die Schnittstelle zwischen Anforderungsingenieurinnen und -ingenieuren sowie Entwicklerinnen und Entwicklern wird so eindeutig und konsistent definiert. Zudem dient die Dokumentation besonders bei komplexen Systemen dazu, die Übersicht wahren zu können. Durch die üblicherweise elektronische Form der Dokumentation ist ein weiterer Nutzen die Zugänglichkeit und die Verfolgbarkeit von Änderungen.

Die Dokumentation schließt für gewöhnlich z.B. auch Protokolle von Interviews etc., aber auch Modelle des Softwaresystems mit, ein. Generell ist die Gesamtdokumentation als gesammeltes Wissen anzusehen, welches neuen Mitarbeiterinnen und Mitarbeitern hilft in das Projekt einzusteigen und idealerweise auch für Folgeprojekte wichtige Erfahrungswerte liefern kann.

Für die Strukturierung von Dokumenten werden Richtlinien angewendet, um Inkonsistenzen möglichst vorzubeugen und eine gewisse Einheitlichkeit zu erreichen. Die Dokumentation sollte darüber hinaus mit der später folgenden Implementation konsistent sein; sie findet als Grundlage in der noch später folgenden Testphase Verwendung.

2.2.1.3 Prüfung und Abstimmung von Anforderungen

Die Aktivität „Prüfung und Abstimmung von Anforderungen“ ist die Grundlage zur Sicherstellung von Qualitätsrichtlinien, die an Anforderungen gestellt werden. Darunter fallen z.B. Notwendigkeit, Konsistenz, Vollständigkeit, Realisierbarkeit oder Abgestimmtheit (siehe dazu [7]). Zur Prüfung können Techniken wie Walkthrough, Perspektiven-basiertes Lesen, Prototyping oder Checklisten eingesetzt werden.

Treten Konflikte auf, so wird im Rahmen der Abstimmung von Anforderungen z.B. eine Entscheidungsmatrix aufgestellt, die mögliche Lösungsalternativen gegen Kriterien abwägt. Generell ist aber auch der Anforderungsingenieur gefordert, um Konflikte allgemeinerer Art zu lösen.

2.2.1.4 Verwaltung von Anforderungen

Die Verwaltung von Anforderungen ist eine während des gesamten Projektes gegenwärtige Aktivität. Ihr Gegenstand ist die zweckmäßige Attributierung von Anforderungen (das ist das Festhalten von Zusatzinformationen, wie z.B. eindeutige Identifikationsnummer oder Autor), Definition von Sichten auf die Anforderungen, die Priorisierung, die Verfolgbarkeit sowie die Versionierung von Anforderungen.

Moderne Werkzeuge sind eine wesentliche Hilfe bei der Ausführung dieser Tätigkeit, da sie Zuständigen viel Arbeit abnehmen und automatisch für viele der gewünschten Effekte sorgen.

2.2.2 Agile Methoden

Das Agile Manifest [9], 2001 durch 17 Pioniere agiler Methoden formuliert, legt die Grundsätze jeder aktuellen agilen Methode fest. Das sind:

- Individuen und Interaktionen stehen über Prozessen und Werkzeugen,
- funktionierende Software steht über umfassender Dokumentation,
- Zusammenarbeit mit Kunden steht über Verträgen und
- Umgang mit Änderungen steht über der Befolgung von Plänen

Basierend auf [10] werden im Folgenden die für diese Arbeit relevanten agilen Methoden Scrum, eXtreme Programming und Lean Development kurz erläutert. Diese Auswahl beruht auf dem bei Allianz vorherrschenden Verständnis von „Agile“ (siehe 3.3).

2.2.2.1 Scrum

Scrum ist ein Projektmanagement Framework zur Entwicklung von Software [10]. Dabei wird zwischen drei Rollen unterschieden: (1) „Das Team [...] führt alle Arbeiten aus, die zur Umsetzung der Anforderungen im Produktinkrement notwendig sind. Das Team ist bevollmächtigt und autonom. Es organisiert seine Arbeit selbst und ist interdisziplinär besetzt. Das Team besteht aus maximal neun Mitgliedern“ [11]. (2) „Der Product Owner ist [...] verantwortlich für den Erfolg des Projekts. Der Product Owner beschreibt die Anforderungen und managt das Product Backlog. Er erstellt den Releaseplan und die Releaseberichte. Er managt die Interessenvertreter und arbeitet eng mit dem Team während der gesamten Projektlaufzeit zusammen.“ [11]. (3) „Der ScrumMaster hilft Scrum richtig anzuwenden. Er unterstützt das Team und

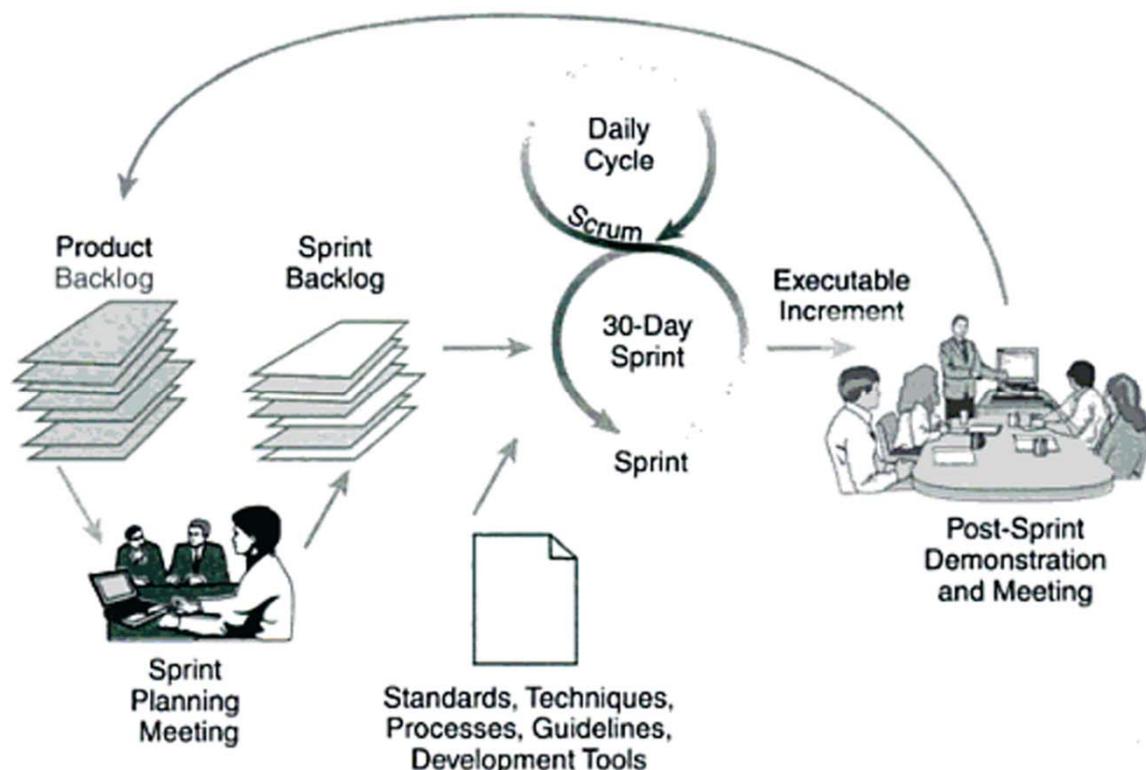


Abb. 2: Scrum Prozess nach [10]

stellt die direkte Zusammenarbeit von Product Owner und Team sicher. Der Scrum-Master beseitigt Hindernisse und hilft dem Team, seine Produktivität kontinuierlich zu steigern.“ [11]. Diese drei Rollen vollziehen den in Abb. 2 dargestellten Prozess von Ken Schwaber und Jeff Sutherland. Er beginnt mit der Erstellung des Product Backlogs durch den Product Owner. Das Product Backlog enthält eine nach Priorität geordnete Liste von Anforderungen (nicht-funktionale und funktionale), deren jeweiliger Aufwand durch das Team geschätzt wurde. Die Priorisierung der Anforderungen nimmt der Product Owner zusammen mit relevanten Stakeholdern vor. Die Anforderungen mit der höchsten Priorität werden entsprechend der Aufwandsschätzungen in ein Sprint Backlog übernommen, welches festschreibt, welche Anforderungen im nächsten Sprint zu realisieren sind. Die Dauer eines Sprints liegt normalerweise zwischen einer und vier Wochen. In dieser Zeit arbeitet das Team an der Realisierung, an der Qualitätssicherung und, falls nötig, zusammen mit dem Product Owner an der Ausarbeitung von Anforderungen. Die Ausarbeitung folgt dabei immer dem Prinzip, nicht mehr als nötig zu spezifizieren („just enough“). Ziel ist es nach dem Sprint ein Stück auslieferbarer Software zu haben. Der Product Owner fungiert als hierbei direkter Ansprechpartner und Interessensvertreter der Kundin bzw. des Kunden.

Während eines Sprints werden in der Regel tägliche Scrum-Meetings abgehalten (Daily Scrum), in welchen das Team die Arbeit unter sich aufteilt und über Fortschritt und Probleme diskutiert wird. Neben dem Product Owner ist der ScrumMaster involviert, um etwaige Probleme mit der Organisation zu klären. Zudem können während des Sprints neue Anforderungen auftreten, die in das Product Backlog mit aufgenommen werden. Eine Veränderung des Sprint Backlogs während eines Sprints ist nicht möglich.

Nach einem Sprint werden die Ergebnisse diskutiert und die gemachten Erfahrungen des Teams reflektiert. Anforderungen aus dem Product Backlog werden nun durch den Product Owner neu priorisiert und der Aufwand der Anforderungen mit hoher Priorität durch das Team detailliert geschätzt. Das Team entscheidet auf Basis der Priorisierung der verbleibenden Anforderung und der eigenen Schätzung über die Notwendigkeit und den Umfang eines nächsten Sprints.

Um Scrum für größere Projekte und Hierarchien von Projekten einsetzbar zu machen, gibt es die Möglichkeit des sogenannten „scrum of scrums“. Hierbei werden Hierarchien von Scrum-Teams realisiert, die vorgeben, dass jeweils eine Vertreterin bzw. ein Vertreter eines „echten“ Teams zusammen in einem organisatorischen Team arbeiten, um so einen übergreifenden Austausch zu erreichen. Diese Schachtelung kann auf diese Weise beliebig fortgeführt werden. Für die Rolle des Product Owners wird diese Schachtelung ebenfalls angewendet, um so übergreifende Interessen zu berücksichtigen.

2.2.2.2 eXtreme Programming (XP)

XP ist ein Werte-basierter Ansatz, der Kommunikation, Einfachheit, Feedback, Mut und Qualitätsarbeit in den Vordergrund stellt [10]. Auf diesen Werten aufbauend bietet XP eine Sammlung an Praktiken, die ein Entwicklungsprozess bereichern kann. Als erstes wird dabei das „Planning Game“ genannt. Hier werden Anforderungen in Form sogenannter User Stories, das sind kurze Beschreibungen von jeweils einer Anforderung, meist in Papierform auf einem kleinen Kärtchen, gesammelt, durch die Entwicklerinnen und Entwickler bewertet und durch die Kunden-Vertreterin oder den Kunden-Vertreter priorisiert. Wie auch bei dem Product Owner aus Scrum, wird von einem stets eingebundenen „on-site customer“ ausgegangen.

Die Realisierung der User Stories erfolgt in kurzen Iterationen von ca. drei Wochen, wobei sie oftmals durch vor der Implementation der Funktionalität realisierte Testfälle (test first) geprägt ist. Mit dem sogenannten „Pair Programming“ werden zwei Programmierinnen und Programmierer an einem Computer mit der Realisierung betreut, wobei sich davon eine Erhöhung der Codequalität erhofft wird. Außer des Quellcodes und der Testfälle wird von XP keine weitere Dokumentation gefordert.

XP eignet sich nach Aussagen der Schöpfer Kent Back, Ward Cunningham und Ron Jeffries nur für kleine Teams (weniger als zehn Personen), obwohl es bereits in größeren Projekten eingesetzt wurde.

2.2.2.3 Lean Development (LD)

LD ist eine sehr strategische agile Methode [10]. Sie zielt darauf ab, die Erstellung einer änderungstoleranten Software mit ein Drittel des menschlichen Aufwands, ein Drittel der Entwicklungszeit, ein Drittel der Investition in Werkzeuge und Methoden und ein Drittel des Aufwands zur Anpassung an neue Marktumgebungen als sonst üblich zu realisieren. Dabei verweist der Schöpfer Bob Charette auf vier erfolgskritische Faktoren: (1) Die schnelle Erstellung von Wert für den Kunden, (2) die Erstellung änderungstoleranter Software, (3) die Erstellung ausschließlich notwendiger Funktionalität und (4) Aggressivität, Starrsinn und der Glaube an die Ziele von LD.

Aus der oben genannten Strategie wird in der Praxis meist das generelle Prinzip der Vermeidung von unnötigem Aufwand extrahiert und kurz mit „Lean“ bezeichnet.

2.2.3 Integration von RE und agilen Methoden

Agile Methoden sind im Vergleich zu traditionellen Entwicklungsprozessen weniger Dokumenten-bezogen. Das liegt nach [7] insbesondere an zwei hervorzuhebenden Unterschieden:

- Agile Methoden zeichnen sich durch ihre explizite Anpassungsfähigkeit aus. Traditionelle Methodiken verlassen sich auf Planung zur möglichst frühen, vollständigen Erfassung des Softwaresystems.
- Agile Methoden sind Werte-orientiert. Klassische Methoden sind Prozess-orientiert.

Der erste Unterschied entspricht genau der Lösung, nach der viele Unternehmen aktuell suchen. Sie haben das Problem, dass sie aufgrund von Kosten- und Zeitdruck den Kontext des zu entwickelten Softwaresystems nicht mehr gänzlich verstehen können und die Kundin bzw. der Kunde keine klare Vorstellung von ihren bzw. seinen Wünschen hat. Insofern bieten agile Methoden dort den Ausweg, die Schwäche traditioneller Vorgehensweisen von sich ändernden Anforderungen zu einer Stärke umzuwandeln.

Der zweite Unterschied ist ähnlich veranlagt. Agile Methoden versuchen die Kundin bzw. den Kunden, wie auch alle sonstigen Beteiligten stärker einzubeziehen und reduzieren dafür die in klassischen Ansätzen in großen Ausmaß gelebten, aber auch sehr kostenintensiven Bereiche, wie die Dokumentation. Doch letztere ist insbesondere bei großen und langfristigen Projekten unverzichtbar und muss in einem gesunden Ausmaß geführt werden.

Bisherige Ansätze das RE klassischer Methoden mit agilen Methoden zu integrieren, versuchen die oben kurz skizzierten Stärken und Zielrichtungen der beiden Methodiken zu vereinen. Basierend auf den in Kapitel 2.2.1 genannten Phasen des RE wer-

den im Folgenden Probleme und mögliche Lösungen vorgestellt, die bei einer Integration von klassischem RE und agilen Methoden auftreten können.

2.2.3.1 Erhebung von Anforderungen

Ein wesentlicher Vorteil, so wird es zumindest in der Philosophie agiler Methoden propagiert, sei die Kunden-zentrierte Entwicklung. Die bzw. der idealerweise stets verfügbare, mit Entscheidungsgewalt versehene Repräsentantin bzw. Repräsentant der Kundinnen und Kunden wird bei XP „on-site customer“ [12], bei Scrum „Product Owner“ [13] genannt. Diese Person hat den Freiraum, Entscheidungen bezüglich Anforderungen selbst zu treffen. Zur Entscheidungsfindung gibt es seitens Scrum und XP allerdings keine Vorgaben. Es bleibt on-site customer bzw. Product Owner selbst überlassen, wie sie bzw. er dies organisiert. Für kleine und mittelgroße Projekte ist diese Tatsache tragbar, für größere Projekte ist es für on-site customer bzw. Product Owner jedoch unumgänglich, alle Stakeholder bei der Entscheidungsfindung einzubinden, wie Ktata und Lévesque in [14] festgestellt haben. Generell gesehen und als eine Voraussetzung agiles Vorgehen skalierbar zu machen, ist es ein Trugschluss zu glauben, dass eine einzige Person für alle Kundengruppen sprechen kann. Insofern ist es unverzichtbar eine Stakeholder-Analyse aus dem klassischen RE durchzuführen, um zu Beginn eines Projektes alle relevanten Beteiligten zu identifizieren. Für den weiteren Verlauf schlagen Ktata und Lévesque im Rahmen einer Erweiterung zu Scrum namens „Business-Scrum“ vor, ein „steering committee“ zu etablieren, das aus mehreren Personen besteht, die die Interessen der Kundinnen und Kunden vertreten und welche dem Product Owner bei der Identifikation, Analyse und Priorisierung von Anforderungen helfen sollen [14]. Nawrocki et al. stellen in [12] ein für XP modifiziertes Modell vor, dass analog mehrere Kundenrepräsentierende vorsieht, welche aber durch einen „senior customer“ ergänzt werden, dem eine zwischen den Repräsentierenden vermittelnde Rolle zufällt. Denkt man als Beispiel an ein größeres Unternehmen in der Rolle des Kunden, so könnte die bzw. der Vorsitzende ein Vorstandsmitglied sein, wobei sonstigen Vertretende Fachbereichsleiterinnen und Fachbereichsleiter sind. Durch diese Erweiterung der agilen Methoden ist es möglich, den Vorteil des stets verfügbaren „on-site customers“ wirklich auszunutzen, ohne dabei die Interessen einzelner Stakeholder zu übersehen.

Die Erhebung der Anforderung kann im Rahmen der in agilen Methoden üblichen initialen Planung, z.B. „Planning Game“ in XP, durch weitere Stakeholder bereichert werden. In diesem Zuge bietet sich auch die Priorisierung der Anforderungen an. Nawrocki et al. schlagen dazu in ihrem modifizierten XP-Planning Game vor, allen Kundenvertreterinnen und -vertretern ein festgelegtes Budget an Arbeitszeit zur Verfügung zu stellen, mit dem es ihnen möglich ist User Stories zu „kaufen“, die ihnen besonders wichtig erscheinen. Auf diese Art sollen vor allem die für alle Interessensgruppen wichtigsten Anforderungen berücksichtigt werden. Gleichzeitig kann so das Problem, dass Software einen Großteil an ungenutzten Features besitzt, eingedämmt werden. Bei dem Anteil ungenutzter Features an der Gesamtmenge der Features sprechen Kotonya und Sommerville in [15] von einem Prozentsatz von 47%. In [14] referenzierte Werte der Standish Group von 2002 zeigen mit 45% ähnliche Beobachtungen. Dabei ist es wichtig, das „just enough“-Prinzip von agilen Methoden wirklich zu leben und als niedrig priorisierte Anforderungen bewusst *nicht* zu implementieren.

Nicht-funktionale Anforderungen sind ein wichtiges Mittel, um Belange der Qualität an ein Softwaresystem auszudrücken und nicht Funktionen oder Daten, wie es funktionale Anforderungen tun [16]. Klassisches RE betrachtet daher die Erhebung von Anforderung als Identifikation von nicht-funktionalen, wie auch funktionalen Anforderungen. Agile Methoden hingegen zielen hauptsächlich darauf ab, mit dem Kunden darüber zu reden, „was“ er denn gerne haben möchte, nicht „wie“. Die aus XP bekannten User Stories beispielsweise sind reine Beschreibungen funktionaler Anforderungen, für nicht-funktionale Anforderungen gibt es kein Mittel, um sie angemessen zu erfassen. Nicht-funktionale Anforderungen haben zudem meist hohe Auswirkungen auf das

Softwaresystem, so dass sie die Implementation funktionaler Anforderungen direkt beeinflussen. Ergänzend wäre es wünschenswert, dass nicht-funktionale Anforderungen möglichst früh in ausreichendem Detailgrad bekannt sein und auch nicht mehr verändert werden würden, so wie es klassisches RE versucht [7; 17].

Wie Rodriguez et al. in [18] beschreiben, hat der Kunde seinen Fokus besonders auf funktionale Anforderungen gerichtet (siehe dazu Abb. 3). Bei einem besonders Kunden-zentrierten Vorgehen, wie den agilen Methoden, ist es eine zusätzliche Gefahr, gar nicht oder erst spät zu relevanten nicht-funktionalen Anforderungen zu gelangen. Es wird die Verwendung sogenannter „System Stories“ vorgeschlagen, die die allgemeinen Wünsche an ein Softwaresystem beschreiben, welche nicht mit User Stories erfasst werden können [18]. Diese System Stories sollten dabei, genau wie User Stories, im XP-Planning Game erfasst und priorisiert werden.

2.2.3.2 Dokumentation von Anforderungen

Reduktion unnötiger Kostenfaktoren ist das Ziel dem eine ausführliche und vollständige Dokumentation bei agilen Methoden zum Opfer gefallen ist. Zwar empfiehlt unter anderem Scrum die Verwendung eines Anforderungsdokuments in Form eines „Product backlog“ bzw. „Sprint backlog“, doch werden keine Aussagen über das Ausmaß der Dokumentation getroffen und so bleibt dies meist den Entwicklerinnen und Entwicklern überlassen. Die Dokumentation wird zentriert auf klaren, kompakten und gut kommentierten Quellcode. Weiterführende Dokumentation beschränkt sich auf Kernbereiche und ist so leichter anpassbar an in agilen Methoden oft vorkommende Änderungen am Softwaresystem [7].

Im Sinne des Requirementsmanagement zeigt sich zudem, dass agile Methoden, wie Scrum, allein nicht ausreichen und mit Praktiken des klassischen RE Modelle aufgewertet werden sollten [17]. Insbesondere ist es wichtig, die in Form von User Stories ohnehin schon vorhandenen Anforderungen neben der Erfassung auch systematisch zu verwalten und Abhängigkeiten sowohl untereinander als auch zur Implementation festzuhalten.

Nawrocki et al. schlagen in [12] vor, explizite Verantwortlichkeiten zur Dokumentation und Verwaltung von Anforderungen zu verteilen. Hierzu wird nahegelegt, die Rolle

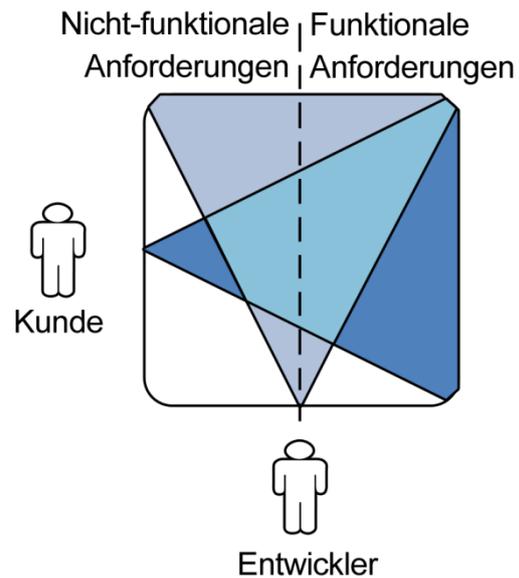


Abb. 3: Sicht auf Softwaresystem nach [18]

des Testers, welche ohnehin stark auf Anforderungen basiert, zur Rolle „tester/analyst“ zu erweitern. Diese sieht vor, Anforderungen systematisch und unter Zuhilfenahme von geeigneten fortgeschrittenen Werkzeugen festzuhalten und zu verwalten, wie man es aus dem klassischen RE kennt [8]. Die Testerin oder der Tester hat so die Chance, gezielt auf Anforderungen zu testen und dies entsprechend zu dokumentieren.

Die Vorteile dieser klareren Vorgehensweise bezüglich Dokumentation liegen auf der Hand. Besonders größere Projekte profitieren von der besseren Dokumentation, da dort die Personalfuktuation höher ist und die Einarbeitung in das Projekt durch die Dokumentation vereinfacht wird. Für langwährende Wartung am Softwaresystem etc. ist eine ausführliche Dokumentation ebenfalls förderlich [7].

Der Nachteil einer ausgedehnteren Dokumentation, viel Zeit und damit Geld zu „verschwenden“, ist im Rahmen agilen Vorgehens nicht so eklatant. Klassischerweise wird versucht, alle in der Zukunft aufkommenden Fragen durch ausreichende Dokumentation im Voraus zu beantworten. Die Folge ist die Spezifikation vieler Dinge, die in diesem Detailgrad nicht notwendig gewesen wäre (da einige Anforderungen z.B. gar nicht implementiert wurden). Agile Methoden hingegen laufen mit dem „just enough“-Prinzip nicht in diese Kostenfalle, können aber trotzdem von den oben erläuterten Vorteilen ausführlicher Dokumentation profitieren.

2.2.3.3 Prüfung und Abstimmung von Anforderungen

„Are we building the right product?“, das ist die Frage, der sich besonders agile Methoden im Sinne der Validierung (Test Driven Development und Ähnliches) stellen [19]. Diese Belange sind auch im klassischen RE wichtig. Scrum z.B. setzt darauf, dass ein Sprint immer ein auslieferbares Produkt hervorbringt, welches idealerweise mit jeglicher Art von Tests geprüft wurde. Doch vernachlässigen agile Methoden die eng mit der Validierung verbundene Verifikation („Are we building the product right?“) [19]. Hier können gerade bei größeren Projekten Konsistenzprobleme auftreten, d.h. Anforderungen, die dasselbe aussagen, aber auf unterschiedliche Art und Weise formuliert sind. Auch Widersprüche zwischen Anforderungen können ein Problem sein. Um dies zu verhindern, schlagen Eberlein und do Prado Leite in [17] vor, agile Methoden um die Instrumentarien des traditionellen RE bezüglich Verifikation zu bereichern. Dies könnte z.B. die Benutzung von Werkzeugen zur automatischen Prüfung von Anforderungen auf Syntax oder die Verwendung von Checklisten bedeuten. Letztlich kann Verifikation zusammen mit den oben genannten Verbesserungsmöglichkeiten zur Dokumentation eingeführt werden. Werkzeuge zur Dokumentation und Verwaltung von Anforderungen unterstützen für gewöhnlich Möglichkeiten zur Verifikation, indem z.B. geeignete Vorlagen zur Erfassung von Anforderungen oder Syntax-Prüfungen von Anforderungsbeschreibungen integriert sind.

Agile Methoden vertreten die Philosophie, Details erst zur Implementation zu klären, d.h. möglichst spät. Zuvor sind nur grobgranulare Anforderungen bekannt. Diese Tatsache stellt eine Gefahr dar, die traditionellem RE so nicht droht. Durch die iterative, einem evolutionären Prototyping ähnelnde, Vorgehensweise agiler Methoden werden Anforderungen immer weiter verfeinert, bis schließlich ein Detail bekannt werden kann, welches mit dem bisherigen Ergebnis inkompatibel ist [18]. So können wichtige Ressourcen verpuffen und zum Scheitern eines Projektes beitragen. Traditionelles RE begegnet diesem Problem mit der ausführlichen und frühen Erhebung von Anforderungen, die im Detailgrad möglichst für das gesamte Projekt ausreicht, wenngleich dies in der Praxis nur schwer möglich ist [20].

Für dieses Problem gibt es nur zwei Lösungen. Die erste lautet das Problem zu ignorieren und sich darauf zu verlassen, dass die Kompetenz der Teammitglieder, auf die es bei agilen Methoden wesentlich mehr ankommt, als bei klassischen Vorgehen, dazu führt, dass solche Crashes möglichst früh erkannt und abgewendet werden können. Die andere und besonders bei unerfahrenen Teams sicherere Variante ist, umfassenderes RE zu betreiben und Gefahr zu laufen, zu viel, als zu wenig zu spezifizieren. Das würde aber gegen das Prinzip von „just enough“ agiler Methoden verstoßen. Entsprechend muss das Ziel sein, immer ein Mindestniveau an Erfahrung im Team zu halten, um mit der erstgenannten Lösung auszukommen und Ressourcen zu sparen. Dazu ist es wichtig, die kritischen Punkte in der Anforderungsspezifikation frühzeitig zu identifizieren und im Detailgrad ausreichend zu dokumentieren.

2.2.3.4 Verwaltung von Anforderungen

Agile Methoden haben Schwächen, übergreifende Anforderungen oder generell Abhängigkeiten zwischen Anforderungen zu beschreiben [18; 21]. Hier bieten spezialisierte RE-Vorgehen, wie das in [21] vorgestellte „aspect-oriented agile RE“ oder das in [14] vorgestellte „Goal-oriented RE“ Wege, die Defizite von agilen Vorgehen zu beseitigen. Im Sinne der Verfolgbarkeit und Dokumentation von Änderungen an Anforderungen ist ein aus klassischem RE bekanntes Requirements Management unabdingbar [17].

2.2.4 Zusammenfassung

Folgende Praktiken, die eine effektive Integration von RE und agilen Methoden erlauben, können zusammenfassend festgehalten werden:

Nummer	Praktik
<i>Kundeneinbindung</i>	
REAM 1	Stakeholder-Analyse durchführen.
REAM 2	„Product Owner“ „steering comitee“ zur Seite stellen (Scrum) oder „on-site customer“ durch restliche Interessensvertretende der Kundengruppen ergänzen (XP).
<i>Nicht-funktionale Anforderungen</i>	
REAM 3	Bewusst auch nicht-funktionale Anforderungen erheben.
REAM 4	System Stories für die Dokumentation und Verwaltung benutzen (Scrum & XP).
<i>Dokumentation</i>	
REAM 5	Über das Rahmenmodell von Scrum und XP hinaus ein Anforderungsdokument pflegen.
<i>Verifikation</i>	
REAM 6	Anforderungen auf Konsistenz prüfen.

REAM 7	Geeignete Werkzeuge einsetzen, die automatische Verifikation von Anforderungen unterstützen
<i>Granularität der Anforderungen</i>	
REAM 8	Kritische Anforderungen zu einem frühen Zeitpunkt im Detail ausarbeiten.
<i>Abhängigkeiten zwischen Anforderungen</i>	
REAM 9	Geeignete Werkzeuge einsetzen, die Abhängigkeiten zwischen Anforderungen modellieren und verfolgen können.

Tabelle 1: Praktiken zur Integration von RE und agilen Methoden

2.3 Gebrauchstauglichkeit

Nach der DIN EN ISO Norm 9241-11 [22] ist Gebrauchstauglichkeit (engl. Usability) „das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“. Dabei werden auch folgende Dialogprinzipien genannt, die Gebrauchstauglichkeit charakterisieren (nähere Beschreibung in DIN EN ISO Norm 9241-110 [23]):

- *Aufgabenangemessenheit*
Das Computerprogramm unterstützt die Benutzerinnen und Benutzer bei der Erledigung ihrer Aufgaben nicht als unnötiges Übel, sondern als Hilfe [24].
- *Selbstbeschreibungsfähigkeit*
Die Benutzerinnen und Benutzer wissen jederzeit, was der Computer gerade macht und welche Eingabe/Reaktion der Computer als nächstes erwartet [24].
- *Steuerbarkeit*
Die Benutzerinnen und Benutzer können die Abfolge der Arbeitsschritte weitgehend selbst bestimmen und, falls erforderlich, die Arbeit unterbrechen und später ohne Verluste fortführen [24].
- *Erwartungskonformität*
Das Softwaresystem ist konsistent und entspricht den Merkmalen der Benutzerinnen und Benutzer sowie allgemeinen Konventionen [24]. Das bedeutet vor allem, dass das Softwaresystem entsprechend der Erwartungen der Benutzerinnen und Benutzer gestaltet ist, im Einzelnen also beispielsweise Buttons an üblichen Stellen platziert werden etc.
- *Fehlertoleranz*
Das beabsichtigte Arbeitsergebnis kann trotz erkennbar fehlerhafter Eingaben mit keinem oder mit geringem Korrekturaufwand durch den Benutzer erreicht werden [24].
- *Individualisierbarkeit*
Das Dialogsystem lässt Anpassungen an die Erfordernisse der Arbeitsaufgabe, individuelle Vorlieben der Benutzerinnen und Benutzer und dessen Fähigkeiten zu [24].
- *Lernförderlichkeit*
Die Benutzerinnen und Benutzer werden beim Erlernen des Dialogsystems unterstützt und angeleitet [24].

Gebrauchstauglichkeit ist eine von mehreren Qualitätsfaktoren für Software. Lauesen nennt in [25] zusätzlich noch Korrektheit, Verfügbarkeit, Performanz, Sicherheit und Wartbarkeit.

Den Weg zu einem gebrauchstauglichen Softwaresystem beschreibt die DIN EN ISO 9241-210 Norm [26]. Sie gibt Anforderungen und Empfehlungen an einen Usability-Engineering Prozess zur Schaffung eines gebrauchstauglichen interaktiven Softwaresystems. Dabei basiert sie auf den Erfahrungen, die man etwas mehr als zehn Jahre lang mit der Vorgängernorm ISO 13407 gemacht hat und wurde im Vergleich zu dieser ergänzt und konkretisiert.

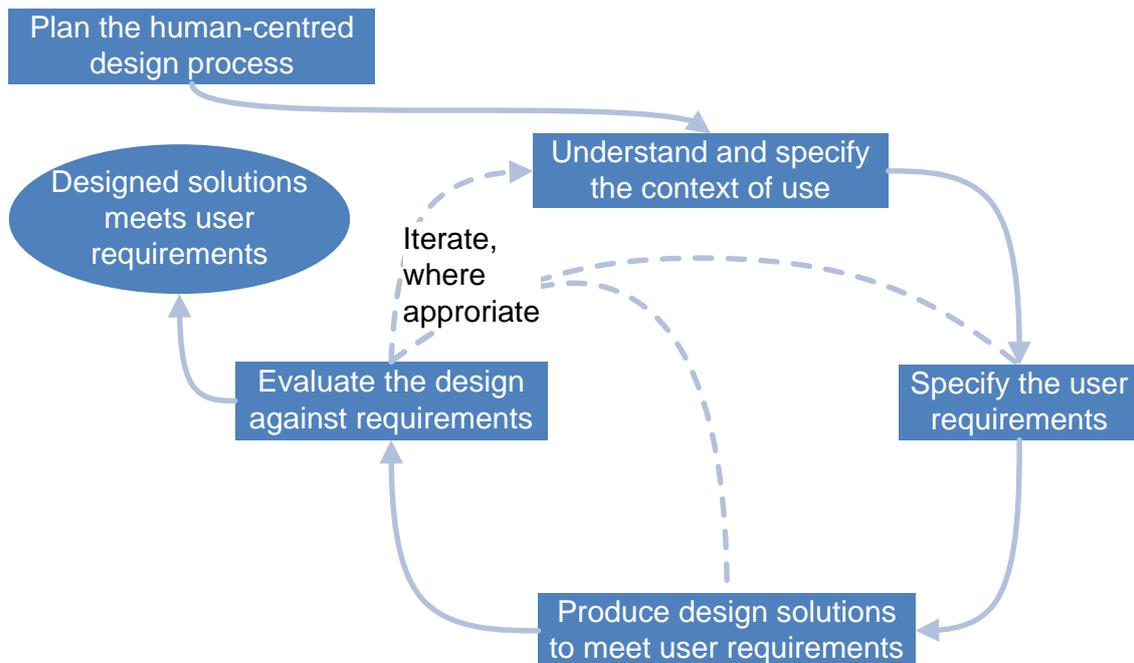


Abb. 4: „Interdependence of human-centred design activities“ aus DIN EN ISO 9241-210 [26]

Benutzerzentriertes Vorgehen beim Entwurf eines Softwaresystems bedarf der in Abb. 4 dargestellten Aktivitäten. Der Ablauf ist nur eine Orientierung und keinesfalls verbindlich. Auch seien diese Aktivitäten in jedes beliebige Vorgehensmodell integrierbar.

Die Aktivitäten zum Entwurf eines gebrauchstauglichen Softwaresystems bedürfen zunächst einer Planung. Hierbei sollen die für das jeweilige Unternehmen und Projekt benötigten Aktivitäten in das bestehende Vorgehensmodell eingegliedert werden. Bei der Identifikation der passenden Aktivitäten steht das Ziel der Reduktion von Risiken zwischen Benutzerinnen und Benutzern und Softwaresystem im Vordergrund.

Der iterative Zyklus von Aktivitäten aus Abb. 4 beginnt nach der Planung mit Verstehen und Spezifizieren des Nutzungskontexts. Hierbei sollen vor allem folgende Punkte berücksichtigt werden:

- die Benutzerinnen und Benutzer sowie andere Stakeholder-Gruppen
- die Charakteristiken der Benutzerinnen und Benutzer sowie deren Gruppierungen
- die Ziele und Aufgaben der Benutzerinnen und Benutzer
- das Umfeld des Softwaresystems

Es ist also wichtig zu verstehen, welche Stakeholder überhaupt am Softwaresystem beteiligt sind und welche davon die späteren Benutzer desselben darstellen. Welche Fähigkeiten, Erfahrungen, Ausbildung, physische Veranlagung, etc. diese mitbringen ist dabei ebenso wichtig, wie die Ziele und Aufgaben, die sie verfolgen und bei welchen das Softwaresystem sie unterstützen soll. Ebenso ist es notwendig zu verstehen, in welchem Umfeld das Softwaresystem angewendet werden soll.

Die gewonnenen Erkenntnisse müssen festgehalten werden, um einen nachvollziehbaren Zusammenhang zu den in der Aktivität „Specify the user requirements“ erhobenen Anforderungen sicher zu stellen. Dabei ist es wichtig, einen Trade-Off zu finden, um sowohl Erlernbarkeit als auch schnelles Arbeiten zu ermöglichen. Die Spezifikation der Anforderungen soll durch die relevanten Stakeholder verifiziert werden, konsistent und aktuell sein.

Auf diesen Anforderungen aufbauend soll ein Entwurf der Benutzungsschnittstelle erstellt und geprüft werden. Dabei kann zum einen auf Usability-Tests, zum anderen auf Inspektionen zurückgegriffen werden. Bei letzteren wird empfohlen, sie zusammen mit Usability-Tests anzuwenden, um vor einem Usability-Test schwerwiegende Fehler zu identifizieren und beheben zu können.

Im Folgenden werden die Methodiken von Lauesen (Kapitel 2.3.1) und Geis (Kapitel 2.3.2) beschrieben, welche jeweils ein Vorgehen zur Konzeption gebrauchstauglicher Produkte umfassen. Kapitel 2.3.3 greift abschließend die vorgestellten Methoden nochmals auf und vergleicht sie in Hinblick auf Gemeinsamkeiten und Unterschiede.

2.3.1 Lauesen

Soren Lauesen erläutert in [25; 27] seinen sogenannten „Virtual Windows“¹ Ansatz (siehe Abb. 5). Dieser Ansatz verfolgt das Ziel, ein möglichst gebrauchstaugliches

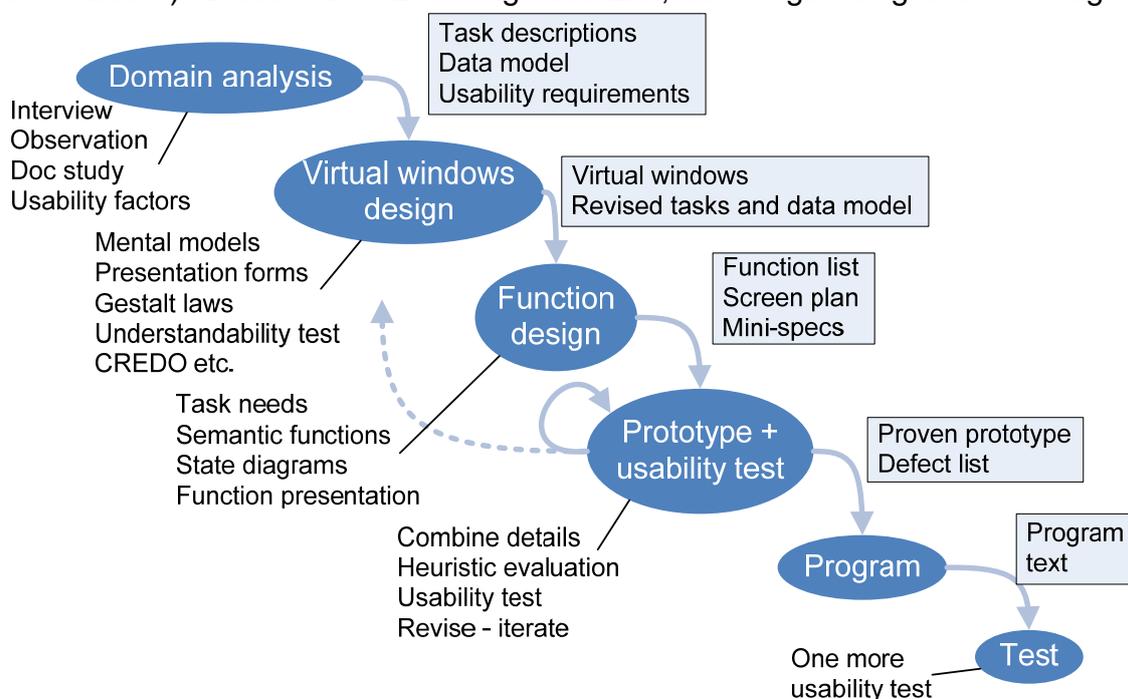


Abb. 5: „Virtual Windows“ Ansatz nach [25]

¹ Ein „Virtual Window“ ist ein Bild auf einem idealisierten Bildschirm, eine Benutzer-orientierte Darstellung von persistenten Daten. Virtual Windows sind auf Papier. Sie zeigen Daten, haben aber keine Buttons, Menüs oder andere Funktionen [25; 28].

Softwareprodukt zu schaffen. Er setzt sich aus zwei traditionellen Vorgehen zusammen [28]. Dies ist zum einen ein Daten-orientiertes Vorgehen, welches dazu neigt, Standard-Formulare zum Erstellen, Bearbeiten und Löschen von Daten bereitzustellen, welche ähnlich zu Microsoft Access alle Daten anzeigen. Zum anderen basiert der Ansatz von Lauesen auf einem Aufgaben-orientierten Vorgehen, welches den Nachteil des Daten-orientierten Vorgehens wettmacht, zu wenig Unterstützung für die effiziente Ausführung der Aufgaben zu geben.

Um oben genanntes Ziel der Schaffung eines möglichst gebrauchstauglichen Softwareprodukts zu erreichen, setzt Lauesen auf eine möglichst frühe Einbindung der Benutzerinnen und Benutzer. Der im Folgenden näher beschriebene Virtual Windows Ansatz vollführt diese frühe Einbindung der Benutzerinnen und Benutzer durch den Entwurf der Virtual Windows, welche als Basis für das gemeinsame Verständnis (der Benutzerinnen und Benutzer und der Entwicklerinnen und Entwickler) des Softwaresystems dienen.

Domänenanalyse

Im ersten Schritt, der Domänenanalyse, soll unter anderem durch Interviews und Beobachtungen erfasst werden, welche Nutzerrollen im Kontext des Softwaresystems existieren und welche Aufgaben diese durchführen. Weiterhin soll erfasst werden, wie die Durchführung der Aufgaben durch das geplante Softwaresystem unterstützt werden könnte, welches Datenmodell dazu nötig ist und welche grobgranularen Nutzungsanforderungen dabei anfallen. Für die zu unterstützenden Aufgaben wird hierbei zunächst nur eine grobe, nummerierte Aufgabenliste gepflegt. In einem ersten Verfeinerungsschritt wird diese Liste nach Arbeitsgebieten gruppiert, so dass ersichtlich wird, in welchem Gebiet welche Aufgaben erledigt werden müssen. Im nächsten Schritt wird zu jeder Aufgabe nach einer bestimmten Vorlage eine Beschreibung angelegt. Diese Beschreibung gibt so systematisch Auskunft über Start- und Endereignis, die Häufigkeit des Auftretens, mögliche Schwierigkeiten und die

T1.2: Check In Start: A guest arrives. End: The guest has got room(s) and key. Accounting started. Frequency: Total: Around 0.5 check-ins per room per night. Per User: 60... Difficult: A bus with 50 guests arrives	
<i>Subtasks:</i>	<i>Example Solution:</i>
1. Find room. Problem: Guest wants neighbour rooms; price bargain. 1a. Guest has booked in advance. 1b. No suitable room.	System shows free rooms on floor maps. System shows bargain prices, time and capacity dependent.
2. Record guest data. 2a. Guest recorded at booking. 2b. Regular guest.	(Simple data entry, see data model) (Search with many criteria, e.g. name, booking number, phone)
3. Record that guest is checked in.	Explicit actor
4. Deliver key. Problem: Guest forgets to return the key; guest wants two keys.	System prints electronic keys. New key for each customer.

Abb. 6: Beispiel eines „Task & Support approach“ [25]

Unteraufgaben, die zu erledigen sind. Auch Varianten einiger Unteraufgaben können hierbei festgehalten werden. Als finalen Detailierungsschritt wird die zuletzt genannte Beschreibung um eine „beispielhafte Lösung“ erweitert, die zu jeder Unteraufgabe exemplarisch erläutert, was genau z.B. durch das System getan wird. Diese Darstellung nennt Lauesen „Task & Support approach“, ein Beispiel gibt Abb. 6.

Neben den Aufgabenbeschreibungen werden auch die Arbeitsgebiete beschrieben, um festzuhalten, wie deren Umfeld aussehen, welche Arten von Benutzerinnen und Benutzern damit beschäftigt sind, etc. Darüber hinaus wird mittels eines Entity-Relationship-Diagramms das Datenmodell des geplanten Softwaresystems festgehalten. Szenarien und Use Case-Diagramme helfen weitere Details und Zusammenhänge des Softwaresystems zu erfassen.

Virtual windows design

Basierend auf den zuvor identifizierten Aufgaben und Daten werden in diesem Schritt des Virtual Windows Ansatzes die Virtual Windows selbst entworfen. Dazu wird ein grober Plan zur Aufteilung der Aufgaben und Daten auf Virtual Windows erstellt - Abb. 7 illustriert ein Beispiel. Es werden nacheinander die Aufgaben betrachtet und überlegt, welche Informationen (Daten) die Benutzerinnen und Benutzer benötigen, um die jeweilige Aufgabe durchführen zu können. Diese Informationen werden in Virtual Windows gekapselt. Es kann zusätzlich festgehalten werden, ob ein Virtual Window nur einmal zur selben Zeit geöffnet sein kann oder nicht. Im Plan wird dies mit einer Ansammlung an Virtual Windows angedeutet, das Virtual Window „Guest“ wäre ein solcher Fall. Es kann vorkommen, dass einige Aufgaben zusätzliche Informationen benötigen, diese aber nicht in einem eigenen Virtual Window, sondern zusammen mit anderen Informationen in einem bestehenden angezeigt werden sollten. Dies trifft in Abb. 7 auf die Aufgabe (Task) „Checkout“ zu.

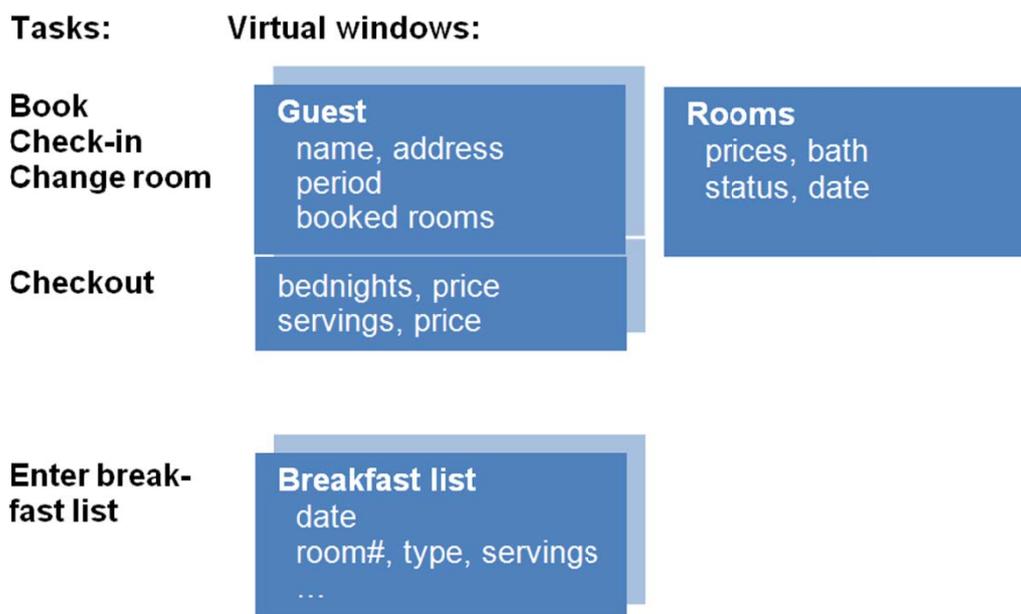


Abb. 7: Beispiel eines Virtual Windows Plans [28]

Um nun bereits eine erste Prüfung der Gebrauchstauglichkeit durchzuführen und den Plan zu validieren, müssen zunächst die Virtual Windows zu einer grafischen Repräsentation ausgearbeitet werden. Dazu wird vorgeschlagen, sich an Standardwege zu

halten, um Daten anzuzeigen. Dazu gehört die Gestalt laws² zu berücksichtigen, extreme aber zugleich realistische Daten beispielhaft zu zeigen und auch die Entwicklungswerkzeuge zu kennen, um keine schwer zu realisierenden Entwürfe zu erstellen. Bei den Entwürfen soll auf jedwede Darstellung von Funktionen verzichtet werden, also insbesondere keine Buttons oder Menüs angedeutet werden, da es nach Lauesen zu früh ist, um diese Komplexität mit einzubeziehen. Die eigentliche Gebrauchstauglichkeitsprüfung, Lauesen nennt dies in diesem Stadium „Understandability-Test“ [27], erfolgt schließlich zusammen mit einer bzw. einem der späteren Benutzerinnen und Benutzer. Hier werden die Papier-Entwürfe nacheinander der jeweiligen Person vorgelegt. Sie soll dann ohne weiteres Zutun der Testerin bzw. des Testers erklären, was ihrer Meinung nach die jeweilige Maske zeigt, ob etwas fehlt, etc. Die Testerin bzw. der Tester schreibt sich dabei alle entdeckten Probleme auf. Es wird empfohlen diesen Test sowohl mit normalen Benutzerinnen und Benutzern, aber auch mit sogenannten Experten-Benutzerinnen und -Benutzern³ durchzuführen, um möglichst alle Perspektiven auf das Softwaresystem einzufangen.

Neben dem Understandability-Test kann der sogenannte CREDO⁴-Check durchgeführt werden, um die Konsistenz des Gesamtentwurfs zu prüfen. Hierbei kann sehr leicht festgestellt werden, ob die geplanten Virtual Windows ausreichen, um alle Daten des erhobenen Datenmodells gemäß CREDO manipulieren zu können oder ob es Defizite gibt.

Zu diesem frühen Zeitpunkt in dem Entwicklungsprozess ist es noch problemlos möglich, teilweise auch erhebliche Änderungen am Entwurf vorzunehmen ohne große Kosten zu verursachen. Das ist nach Lauesen ein großer Vorteil seiner Methode, um Problemen vorzubeugen und so zu einem möglichst gebrauchstauglichen Softwareprodukt zu kommen.

Function design

Im dritten Schritt des Vorgehensmodells von Lauesen werden zunächst semantische Funktionen⁵, die im Rahmen der Aufgaben benötigt werden, auf die Virtual Windows verteilt. Dazu werden die Aufgaben wieder nacheinander durchgegangen und die Funktionen entsprechend zugeordnet. Zu jeder Funktion wird eine „Mini-spec“ erstellt, die kurz beschreibt was das Softwaresystem tut, wenn die Funktion ausgelöst wird. Hier können auch wichtige Randbedingungen für eine Funktion beschrieben werden, die auf keinen Fall bei der späteren Implementation vergessen werden dürfen.

Nach den semantischen Funktionen sollen die Navigationsfunktionen betrachtet werden. Hierzu wird ein Zustandsübergangdiagramm vorgeschlagen, dessen Knoten als Sichten und dessen Kanten als Funktionen anzusehen sind. Eine Sicht kann aus einem oder mehreren Virtual Windows bestehen - die Aufteilung derselben obliegt den Designerinnen und Designern. Die Funktionen umfassen sowohl semantische Funktionen wie auch Navigationsfunktionen. Zunächst jedoch wird nur von semanti-

² Gestalt laws beschreiben die intuitive Wahrnehmung von Formen oder Objekten. Ein Beispiel ist das Gesetz der Nähe, welches besagt, dass nahe beieinander liegende Objekte als zusammengehörig wahrgenommen werden [25].

³ Eine Experten-Benutzerin bzw. ein Experten-Benutzer ist jemand mit einem größeren Überblick über das Softwaresystem und Wissen über schwierige Situationen und gilt als direkte Anlaufstelle bei Problemen „normaler“ Benutzerinnen und Benutzer [25].

⁴ CREDO = Create, Read, Edit, Delete, Overview [25]

⁵ Semantische Funktionen sind solche, deren Aufruf „ernste“ Folgen hat. Beispiele dafür sind Änderungen persistenter Daten oder das Senden von Informationen an die Umgebung [25].

schen Funktionen aus der bisherigen Vorarbeit ausgegangen und schließlich die fehlenden Übergänge mit Navigationsfunktionen ergänzt.

Nachdem die Funktionen zu den Virtual Windows ergänzt wurden, wird die genaue Darstellung derselben festgelegt. Hierbei ist es wichtig, den genauen Nutzungskontext des Softwaresystems erkannt und verstanden zu haben, um die beste Wahl für die Benutzerinnen und Benutzer treffen zu können. Lauesen erwähnt dabei besonders verschiedenartige Typen von Benutzerinnen und Benutzern zu berücksichtigen, so z.B. für die Aktivierung von Funktionen Tastatur-Shortcuts und Klickbuttons anzubieten, wobei Klickbuttons für Einsteigerinnen und Einsteiger und Tastatur-Shortcuts für Expertinnen und Experten relevant sind. Die Entscheidung, wie die jeweilige Funktion repräsentiert werden soll, soll zusätzlich in der Mini-spec festgehalten werden, wenn möglich auch mit einer kurzen Begründung.

Prototype + usability test

Im nächsten Schritt werden die noch groben Entwürfe als Virtual Windows mittels Prototypen konkretisiert. Dazu werden alle Erkenntnisse zusammengefasst und nach Lauesen am besten als Paper-Mock-Up bereitgestellt. Sind alle Prototypen vorhanden, so dienen diese als Grundlage für einen Usability-Test. Hierbei werden einer späteren Benutzerin bzw. einem späteren Benutzer die Prototypen vorgelegt und die Aufgabe gestellt, die typischen Arbeitsabläufe mit diesen Prototypen zu vollführen. Dabei auftretende Probleme etc. werden festgehalten.

Lauesen hat dabei festgestellt, dass durch sein Vorgehen kritische Probleme bereits früh entdeckt werden und in der Prototyping-Phase meist nur noch kleine Mängel entdeckt werden.

Program & Test

Die Vorarbeit, die bis zu diesem Zeitpunkt geleistet wurde, ist nach Lauesen ideal geeignet, um reine Programmierinnen und Programmierer mit der Implementation zu betrauen. Dieser folgt ein abschließender Usability-Test mit der dann fertigen Anwendung.

Lauesen stellte in [27] fest, dass der Aufwand durch Durchführung seines Ansatzes der Virtual Windows sich sehr positiv auswirkte. So konnten in einem Pilotprojekt doppelt so viele Einheiten des Endprodukts zu dem doppelten Preis verkauft werden. Für das agile Umfeld schlägt Lauesen in [25] vor, besonders die Domänenanalyse (domain analysis) und die Usability-Tests zu verfolgen. Darüber hinaus sagt er, dass sein Vorgehen dynamisch mit agilen Methoden umgehen kann und auch mit der Projektgröße skaliert. So können bestimmte Teilschritte, wie etwa der Funktionsentwurf (function design), für kleinere Projekte radikal gekürzt werden. In agilen Methoden sei es problemlos möglich zwischen den Schritten zu iterieren.

2.3.2 Geis

Thomas Geis stellt in [29] einen Usability Engineering-Prozess vor, der die Schaffung eines gebrauchstauglichen interaktiven Softwaresystems zum Ziel hat (Abb. 8). Dieser Prozess besteht aus fünf Aktivitäten, Er beginnt mit der Ermittlung der tatsächlichen Nutzergruppen, also diejenigen Gruppen von Stakeholdern, die die spätere Software benutzen. Um diese Gruppen zu identifizieren, soll der Nutzungskontext analysiert werden. Das betrifft die Benutzerinnen und Benutzer, die Aufgaben, die



Abb. 8: Usability Engineering-Prozess nach Geis

Arbeitsmittel und die physische und soziale Umgebung, in der das Produkt eingesetzt wird.

Im zweiten Schritt wird der Kontext der Nutzergruppen beschrieben. Dies geschieht in Form sogenannter Kontextszenarien. Diese beschreiben zunächst, wessen Nutzungskontext betrachtet wird und in welchem Zusammenhang diese Person mit dem Softwaresystem steht. Weiter wird beschrieben, welche typischen Aufgaben die Benutzerinnen und Benutzer wie erledigen, welche Sonderfälle sie dabei beachten müssen und auf welche Probleme sie stoßen. Auch besondere Wünsche der Benutzerinnen und Benutzer werden festgehalten. Von solchen Szenarien sollen pro Nutzergruppe drei bis fünf erhoben werden. Um dies zu erreichen sollen zweistündige Interviews mit jeweils einer Person aus der jeweiligen Nutzergruppe gehalten werden. Die Interviews sollen von zwei Personen geführt werden, welche auch zusammen dafür Sorge tragen, dass das aus dem Interview entstehende Kontextszenario vollständig erfasstes Wissen beschreibt. Für die Interviews sollten Fragen vorbereitet sein. Bei unerfahrenen Interviewern sollte die Durchführung zuvor geprobt werden. Durch die Interviews und die davon abgeleiteten Kontextszenarien werden laut Geis 80% der sogenannten Nutzungsanforderungen⁶ an ein Softwaresystem identifiziert.

Die zuvor identifizierten Nutzungsanforderungen werden im dritten Schritt des Usability Engineering-Prozesses spezifiziert. Dabei werden zunächst die Kontextszenarien ausgewertet. Dies geschieht tabellarisch und ist, wie in Tabelle 2 gezeigt, aufgebaut. Die Tabelle wird schon im zweiten Schritt, der Beschreibung der Arbeitskontexte pro Nutzergruppe, in der ersten Spalte ausgefüllt. Im jetzigen, dritten Schritt wird dann für jedes Kontextszenario ein Erfordernis⁷ abgeleitet. Dieses Erfordernis wird einem Dialogprinzip (siehe 2.3) zugeordnet und schließlich daraus eine Nutzungsanforderung definiert. Eine Nutzungsanforderung sollte dabei nach dem Muster „Die Benutzerin bzw. der Benutzer muss am System ...“ formuliert sein. Zuletzt werden Folgerungen für die Organisation als „organisatorische Anforderung“ festgehalten.

Geis schlägt zudem eine Erweiterung der Tabelle um eine Systemspezifikation vor, die mögliche Realisierungsmöglichkeiten, wie auch finale Vorgaben festhalten kann.

⁶ Eine Nutzungsanforderung ist „eine Benutzeraktion, die am System ausführbar sein muss, um ein Erfordernis im Nutzungskontext effizient zu erfüllen“ [29].

⁷ Ein Erfordernis ist „eine notwendige Voraussetzung, die es ermöglicht, den in einem Sachverhalt (Nutzungskontext) enthaltenen Zweck effizient zu erfüllen“ [29].

<i>Kontextszenario</i>	<i>Erfordernis</i>	<i>Dialogprinzip</i>	<i>Nutzungsanforderung</i>	<i>Organisatorische Anforderung</i>
...
...
Umsatzerwartungen in seiner Abteilung ermittelt Abteilungsleiter Herr Nauheim aufgrund von Abschätzungen der Mitarbeiter. Herr Nauheim spricht mit Mitarbeitern typischerweise über die zu erwartenden Umsätze für den kommenden Monat und das laufende Quartal. In seiner Abteilung sind immer etwa 30 bis 50 Angebote bei Kunden und Interessenten offen, deren Wert sich auf mehrere 100.000 € beläuft.	Herr Nauheim muss wissen, welche Umsätze in seiner Abteilung zu erwarten sind im kommenden Monat und im laufenden Quartal, um Planungssicherheit zu haben.	Aufgabenangemessenheit	Die Benutzerin bzw. der Benutzer muss am CRM-System eine Auftragsvorschau für seine Abteilung einsehen können, die zu erwartende Umsätze anzeigt, aufgeschlüsselt nach Kunden und in Gesamtsumme für: - Nächsten Monat - Laufendes Quartal.	Bei der Angebotslegung müssen erwartete Startzeitpunkte, Endzeitpunkte und Auftragswahrscheinlichkeiten im CRM-System eingegeben werden. Identifizierte Veränderungen im Laufe der Angebotsphase müssen im System aktualisiert werden.

Tabelle 2: Auswertung der Kontextszenarien (mit Beispiel)

<i>(Teil)Aufgabe</i>	<i>Nutzungsanforderung</i>	<i>Prüfergebnis</i>	<i>Objektiver Nachweis</i>	<i>Verletztes Dialogprinzip / Normempfehlung</i>
...
...
A1: Rückwärts einparken	NA1: Der Autofahrer muss durch die Einparkhilfe zurückgemeldet bekommen, wie weit er vom nächsten Objekt entfernt ist, auf das er sich zu bewegt.	<input type="checkbox"/> Erfüllt <input checked="" type="checkbox"/> Nicht erfüllt	Es gibt lediglich akustische Signale. So ist für den Autofahrer nicht eindeutig feststellbar, wie viel Platz tatsächlich bis zum Hindernis verfügbar ist, auf das er sich zu bewegt.	Selbstbeschreibungsfähigkeit „Rückmeldungen oder Erläuterungen sollten genau auf die Situation bezogen sein, für die sie gebraucht werden, um ihren Wert für den Nutzer zu erhöhen.“ ISO 9241-10, Abschnitt 3.3

Tabelle 3: Inspektionsbericht (mit Beispiel)

Für die im vierten Schritt vorgesehene Inspektion⁸ wird eine Analyse des bestehenden Softwaresystems oder alternativ auch der Prototypen des SOLL-Systems mit den zuvor spezifizierten Nutzungsanforderungen durchgeführt. Hierzu wird erneut die Er-

⁸ Eine Inspektion ist ein „Vergleich der Nutzungsmöglichkeiten / Produktmerkmalen eines interaktiven Systems mit spezifizierten Nutzungsanforderungen durch einen Usability-Tester“ [29].

fassung der Daten in einer Tabelle, wie in Tabelle 3, vorgeschlagen. Entsprechend der (Teil)Aufgaben des interaktiven Softwaresystems werden die Nutzungsanforderungen auf Erfüllung geprüft. Dieses Prüfergebnis wird mit einem „objektiven Nachweis“ begründet und belegt. Zusätzlich wird in der letzten Spalte festgehalten, gegen welche der Dialogprinzipien verstoßen wurde, insofern das Prüfergebnis negativ ausfiel. Bei der Erstellung eines neuen Softwaresystems wird dieser Schritt, zusammen mit der Anpassung der Prototypen, für gewöhnlich mehrmals ausgeführt.

Neben der Inspektion sieht der vierte Prozessschritt teilnehmende Beobachtungen⁹ vor. Diese zielen darauf ab, die Effizienz des vorliegenden Softwaresystems (also Bestandssystem oder Prototyp) zu untersuchen. Geis schlägt dazu vor, Nutzungsszenarien festzuhalten. Dazu werden Benutzerinnen und Benutzer bei ihrer Arbeit mit dem interaktiven System beobachtet und notiert, welche Aktionen dabei durch den die Personen durchgeführt werden und wie das System reagiert. Auch weitere, dabei auftauchende, Nutzungsanforderungen sollen aufgeschrieben werden.

Neben diesen Nutzungsszenarien wird eine Liste kritischer Fehler geführt. Diese hält fest, auf welche Probleme die Benutzerinnen und Benutzer bei der Ausführung der Aktionen stoßen und gegen welches Dialogprinzip dies jeweils verstößt.

Im fünften Schritt werden die bisher erlangten Erkenntnisse zusammengefasst - nähere Informationen kann [29] dazu nicht geben, Allianz liegt darüber hinaus auch kein Material vor.

Der vorgestellte Prozess bietet sich an, um Defizite eines Bestandssystems aufzudecken und daraus Verbesserungsvorschläge für ein neues Softwaresystem ableiten zu können.

2.3.3 Gemeinsamkeiten und Unterschiede

Der von Geis vorgeschlagene Usability Engineering-Prozess ist analytisch und zielt vor allem auf die Identifikation von Mängeln bezüglich Gebrauchstauglichkeit bei bestehenden Softwaresystemen ab. Der Prozess ist aber auch für Neuentwicklungen anwendbar, wenngleich Lauesens Ansatz letzteren Fall besser zu unterstützen scheint. Dieser Schluss lässt sich dadurch begründen, dass die Quellen zu Geis den Prozess nur sehr vage beschreiben und praxisrelevante Techniken thematisieren, die zumeist auch, von in der Praxis vorherrschenden, Bestandssystemen ausgehen. Die Quellen zu Lauesen hingegen sind in jeder Hinsicht sehr ausführlich und greifbar, was seinen Ansatz als konkreter und daher als leichter einzuführen erscheinen lässt. Die grundlegenden Konzepte der beiden Ansätze sind sehr ähnlich, denn beide legen besonderen Wert darauf *alle* relevanten Stakeholder zu identifizieren, deren Umfeld und Charakteristiken zu verstehen und eng mit ihnen verbunden die Benutzungsschnittstelle zu erfassen. Dies entspricht den Maßgaben, die die DIN EN ISO 9241-210 Norm liefert. Besonders Lauesen verinnerlicht auch die restlichen Empfehlungen der Norm, indem er die Benutzerinnen und Benutzer besonders stark bei dem Entwurf der Benutzungsschnittstelle einbindet und die Benutzungsschnittstelle iterativ mit ihnen abstimmt.

⁹ Eine teilnehmende Beobachtung ist eine „Beobachtung von Benutzerinnen und Benutzern bei der Erledigung von Kernaufgaben zur Feststellung, ob vorhandene Nutzungsmöglichkeiten / Produktmerkmale hindernisfrei / effizient genutzt werden können.“

2.3.4 Zusammenfassung

Folgende Praktiken können zusammenfassend festgehalten werden. Sie stellen die maximalen Möglichkeiten dar, die Lauesen und Geis bieten, um einen Softwareentwicklungsprozess bezüglich der Berücksichtigung von Gebrauchstauglichkeit zu optimieren:

<i>Nummer</i>	<i>Praktik</i>
<i>Domänenanalyse</i>	
GEBR 1	Nutzerrollen ermitteln.
GEBR 2	Aufgaben und deren mögliche Unterstützung durch Software ermitteln.
GEBR 3	Nutzungsanforderungen ermitteln.
GEBR 4	Datenmodell ermitteln.
<i>Frühe Einbeziehung der Benutzer</i>	
GEBR 5	Virtual Windows verwenden.
GEBR 6	Understandability-Test mit späteren Benutzerinnen und Benutzern durchführen (normale, aber auch Experten-Benutzerinnen und -Benutzer).
<i>Prototyping</i>	
GEBR 7	Mittels Paper-Mock-Ups Usability-Tests mit späteren Benutzerinnen und Benutzern durchführen (normale, aber auch Experten-Benutzerinnen und -Benutzer).
GEBR 8	Iterativ den Entwurf optimieren.
<i>Abschließende Prüfung auf Usability</i>	
GEBR 9	Im Rahmen des Abnahme-Tests noch einen abschließenden Usability-Test mit den späteren Benutzern durchführen

Tabelle 4: Praktiken zur Optimierung eines Softwareentwicklungsprozesses bezüglich Gebrauchstauglichkeit

3 IST-Zustand bei Allianz

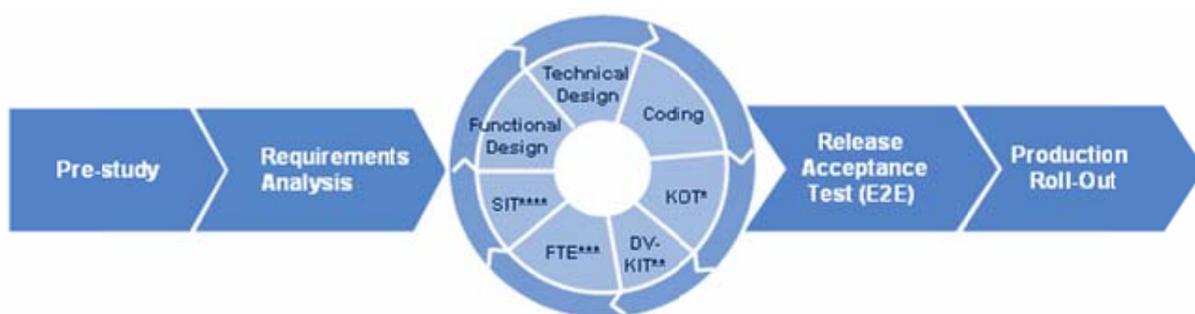
Dieses Kapitel beschäftigt sich in Abschnitt 3.1 zunächst mit dem Requirements Engineering-Prozess, wie er bisher bei Allianz vorherrscht. Kapitel 3.2 bewertet diesen IST-Zustand und leitet Schlussfolgerungen für eine Optimierung ab. In Kapitel 3.3 wird abschließend erläutert, welche Änderungen im Rahmen der agilen Bewegung bei Allianz auftraten und noch auftreten werden.

3.1 RE-Prozess

Um zu den Requirements Engineering-Prozess, so wie er gerade bei Allianz gelebt wird, zu erfassen, bedarf es zunächst einiger Vorarbeit. Als erstes ist wichtig das Umfeld, in dem sich der Prozess bewegt, zu verstehen. Dazu wird das ABS Vorgehensmodell in Unterkapitel 3.1.1 betrachtet. Hier kann der von RE betroffene Bereich identifiziert und extrahiert werden, um daraus einen RE-Prozess zu erheben. Um dies zu bewerkstelligen, wurden die Prozessdokumentationen und Schulungsunterlagen von Allianz betrachtet [30; 31; 32; 33; 34; 35] und zusätzlich mit einigen Beteiligten von Projekten, die das ABS Vorgehensmodell durchliefen, Interviews geführt (siehe F.2.5). Der daraus resultierende Prozess wird ab Abschnitt 3.1.2 erläutert.

3.1.1 ABS Vorgehensmodell

Um den Requirements Engineering-Prozess verstehen und analysieren zu können, ist es wichtig diesen im Kontext des gesamten Software Entwicklung-Prozesses zu sehen. Abb. 9 zeigt einen Überblick des sogenannten ABS Vorgehensmodells [34]. Hierin wird festgelegt, in welchen Phasen welche Ergebnistypen mit welchem Werkzeug von welchen Rollen nach welchen Richtlinien erstellt werden. Das Modell beschreibt das Vorgehen, wie ein neues Release im ABS durchgeführt werden soll. Dabei wird zwischen fünf größeren Phasen unterschieden, die im Folgenden genauer erläutert werden.



* KOT: Komponententest

** DV-KIT: Datenverarbeitungs-Komponentenintegrationstest

*** FTE: Fachlicher Test der Einzelfunktion

**** SIT: Planung, Vorbereitung und Durchführung durch das ABS Test Center

Abb. 9: Überblick über ABS Vorgehensmodell

3.1.1.1 Vorphase (Pre-study)

Die Vorphase beinhaltet die Ersterfassung grober, grundlegender fachlicher Anforderungen an ABS, sogenannter Basisanforderungen. Basierend auf diesen wird ein

Releaseplan abgestimmt, der die Planungen der Arbeiten am „Kern“, das ist alles, was die gesamte Allianz Group betrifft, und die Arbeiten bei der Allianz Deutschland AG (ADAG), das ist alles, was nur die deutsche Allianz Group betrifft, berücksichtigt. Ziel dieser Phase und gleichzeitig der Übergang zur folgenden Phase ist ein genehmigter Projektantrag.

3.1.1.2 Anforderungsanalyse (Requirements Analysis)

Die Anforderungsanalyse ist eng mit der Vorphase verknüpft. Hier sollen zusätzlich Anforderungen an das ABS identifiziert werden. Darüber hinaus werden die gefundenen Basisanforderungen priorisiert. Wird der Nutzen einiger Basisanforderungen gering eingestuft, so werden diese nicht weiter für die Umsetzung verfolgt, sondern ein fachlicher Workaround definiert, um das Ziel dieser Anforderung dennoch zu erreichen. Die restlichen Basisanforderungen werden zu sogenannten abgeleiteten Anforderungen verfeinert. Abgeleitete Anforderungen sind dabei so zu verstehen, dass sie notwendige Tätigkeiten zur Umsetzung der dazugehörigen Basisanforderung beschreiben.

Am Ende der Anforderungsanalyse steht ein besonders hervorzuhebender Status. Dieser symbolisiert, dass die Anforderungen „ready“ sind, d.h. soweit spezifiziert, dass das Team, welches im Folgenden damit weiterarbeiten muss, diese versteht.

3.1.1.3 Fachliches und technisches Design, Implementation und Integration

Im modellierten Zyklus des fachlichen und technischen Designs, der Implementation und Integration wird aus den abgeleiteten Anforderungen ein detaillierter Systementwurf erstellt, der als Grundlage für die Implementierung herangezogen wird. Vor dieser wird zunächst noch eine Prüfung des Entwurfs zur Umsetzung in Hinblick auf Usability und Ergonomie durchgeführt. Nach der eigentlichen Implementation folgen Komponenten- und Integrationstests. Hinzu kommen ein Systemtest und ein fachlicher Test der Einzelfunktionalitäten.

3.1.1.4 Releaseabnahmetest (Release Acceptance Test)

Der Releaseabnahmetest fungiert als klassischer Abnahmetest. Darüber hinaus werden aber auch das Zusammenspiel der vorhandenen Komponenten in ABS mit der neuen geprüft, wie auch auf Einhaltung der Richtlinien des Zielbetriebsmodells (ZBM)¹⁰ zusammen mit deren zu SOX¹¹, Datenschutz, Revision sowie Usability und Ergonomie (Styleguide).

¹⁰ Das Zielbetriebsmodell ersetzt die bisherige strikte Spartenrennung der Allianz durch die vernetzte und funktionale Struktur. Die Geschäftsprozesse von Sach, Leben und Kranken werden gebündelt in zentralen Anlaufstellen für Kundinnen und Kunden, die das Fach-Know-How aller Sparten umfassen. Erste Ansprechpartner sind die Mitarbeiterinnen und Mitarbeiter der Kundenbetreuung, die sich um die Bearbeitung der standardisierten und häufig vorkommenden Geschäftsprozesse kümmern (beispielsweise Adressänderungen), aber auch spartenspezifische Vorgänge (z.B. Vertrags- oder Schadenangelegenheiten) bearbeiten. Ziel ist es, mindestens 80% der Kunden- und Vertriebsanliegen beim Erstkontakt in der Kundenbetreuung abschließend zu bearbeiten. Technisch werden die Prozesse des ZBM durch das ABS unterstützt.

¹¹ Sarbanes-Oxley Act (SOX) ist ein US-Bundesgesetz, dessen Ziel es ist, das Vertrauen der Anleger in die Richtigkeit und Verlässlichkeit veröffentlichter Finanzdaten von Unternehmen zu sichern.

3.1.1.5 Produktivsetzung (Production Roll-Out)

Die letzte Phase ist die Produktivsetzung. Hier wird die voll spezifizierte, entwickelte und getestete Komponente in ABS in Betrieb genommen. Begleitend werden, falls nötig, z.B. Schulungen angeboten.

3.1.1.6 Einordnung des RE-Prozesses

Requirements Engineering ist in dem ABS Vorgehensmodell hauptsächlich in der Vorphase und der Anforderungsanalyse zu finden. Auch in dem fachlichen und technischen Design spielt RE eine Rolle. Im Rahmen der Tests auf Usability ist RE auch an einigen Testschritten beteiligt.

3.1.2 Rollen

Im Folgenden werden die im IST-RE-Prozess erwähnten Rollen kurz erläutert. Sollte bei den Namen der Rollen nur das ursprüngliche grammatische Geschlecht verwendet worden sein, so ist dies durch die Vorgaben von Allianz in Adonis begründet. Es wird hier ausdrücklich darauf hingewiesen, dass damit auch jeweils das andere Geschlecht angesprochen ist.

3.1.2.1 Anforderndes bzw. umsetzendes Teilprojekt

Ein Teilprojekt ist ein Versicherungsbereich, der noch nicht nach ABS migriert wurde, bzw. welcher in ABS für die speziellen Belange des deutschen Umfeldes nicht ausreichend unterstützt wird. ABS ist noch neu und soll viele heterogene Bestandssysteme zu einem gemeinsamen System bündeln. Beispiele sind „Vertrag“, „Schaden“ oder „In-/Exkasso“. Gemäß den exekutiven Strukturen kann eine Anforderung, d.h. die Notwendigkeit einer Änderung, aus einem Teilprojekt entstammen (→ anforderndes Teilprojekt), aber von einem anderen Teilprojekt umgesetzt werden (→ umsetzendes Teilprojekt). Im Regelfall (80% der Fälle) sind das anfordernde und das umsetzende Teilprojekt jedoch identisch. Der Grund für diese Unterscheidung ist, dass die Gesamtprojektstruktur aus verschiedenartigen Teilprojekten besteht.

Ein Teilprojekt setzt sich aus BO Mitarbeiterinnen und Mitarbeitern (= fachliche Expertinnen und Experten) und IT Mitarbeiterinnen und Mitarbeitern zusammen. Es besteht aus bis zu 50 Personen.

3.1.2.2 Anwender

Die Anwenderin oder der Anwender (gleichbedeutend mit Benutzerin oder Benutzer) ist eine Sachbearbeiterin bzw. Sachbearbeiter eines Dienstleistungsgebiets bei Allianz (z.B. „Sach“). Bezüglich dieser Sparte hat diese Person „Expertenwissen“.

3.1.2.3 BO PO / BO PL bzw. IT PO / IT PL

Die Projektleiterin oder der Projektleiter (PL) – im agilen Sinne ist dies bei Allianz der Product Owner (PO) - seitens der BO ist im Rahmen des RE-Prozesses für die größten Teile der Erhebung und Ausarbeitung der Anforderungen verantwortlich.

Die Rolle wird für gewöhnlich von einer erfahrenen Mitarbeiterin oder einem erfahrenen Mitarbeiter eingenommen. Diese Person kennt die IST-Prozesse und weiß auch, welche Möglichkeiten ABS bietet.

Im Rahmen der IT ist die Projektleiterin bzw. der Projektleiter für die Umsetzung verantwortlich.

3.1.2.4 BO- bzw. IT-Usability-Verantwortlicher

Die oder der Usability-Verantwortliche ist Ansprechpartner für Belange der Gebrauchstauglichkeit. Die Rolle kann direkt aus BO oder IT besetzt sein, aber auch aus dem Kompetenzzentrum für Ergonomie und Usability der ADAG. Letzteres kann von Usability-Verantwortlichen in jedem Fall unterstützend herangezogen werden. Im Rahmen der BO liegt der Fokus dieser Rolle eher auf dem ABS-Styleguide und dessen Richtlinien, im Rahmen der IT eher auf gängigen Normen, wie DIN EN ISO 9241 und dem Leitfaden BGI 650 [36].

3.1.2.5 Facharchitektur

Die Facharchitektur ist bei Produktgestaltung, Geschäftsgrundlagen, Datenmodell, Fachdesign, fachlicher Architektur und Anforderungsmanagement involviert. Sie besteht aus ca. acht Personen.

Im RE-Prozess ist sie als Verantwortlicher für die Aktivitäten im Unterprozess „Anforderungen überprüfen“ verantwortlich.

3.1.2.6 Fachlicher Tester

Die fachliche Testerin oder der fachliche Tester ist eine vom Work Team unabhängige Person, die am Ende des RE-Prozesses den Releaseabnahmetest durchführt. Dazu muss sie die in AMT dokumentierten Anforderungen entsprechend der festgehaltenen Testabnahmekriterien prüfen.

3.1.2.7 Programmleitung

Die Programmleitung übernimmt die Steuerung des Gesamtprojekts. Sie besteht aus zehn Personen. Im RE-Prozess ist die Programmleitung für die Priorisierung der Basisanforderungen verantwortlich.

3.1.2.8 Projektmanager (PM)

Die Projektmanagerin oder der Projektmanager ist verantwortlich für alle einem Gebiet (z.B. „Sach“) betreffende Teilprojekte.

3.1.2.9 Work Team

Die Bezeichnung des Work Teams entstammt dem neueren, agilen Vorgehensmodell. Um Verwechslungen mit anderen „Teams“ zu vermeiden, wurde das aus Scrum bekannte „Team“, bei Allianz als „Work Team“ eingeführt. Das Work Team ist eine Mischung aus Entwicklerinnen und Entwicklern aus der IT und Mitarbeiterinnen und Mitarbeitern aus der BO, die fachliche Kenntnisse besitzen. Es besteht aus fünf bis neun Personen.

3.1.3 Werkzeuge

In diesem Unterkapitel werden die im IST-RE-Prozess erwähnten Werkzeuge kurz erläutert.

hauptsächlich die nativ gegebenen Features, wie Testplanung und -auswertung. Genauere Informationen inklusive Downloadmöglichkeit einer Demonstrationsversion gibt [38].

3.1.3.4 ObjectiF

ObjectiF ist ein von microTOOL entwickeltes und vertriebenes Werkzeug, um UML zu modellieren und darauf basierend Software zu entwickeln. Genau auch in diesem Bereich kommt ObjectiF bei Allianz zum Einsatz. Allianz modelliert so UML Diagramme im Rahmen des Systementwurfs und der Spezifikation von Systemen. Dies umfasst hauptsächlich UML Use Case-Diagramme.

In den jeweiligen Modellen werden die Abhängigkeiten zu den Anforderungen aus AMT durch das Festhalten der entsprechenden IDs erfasst.

Noch mehr Informationen über die Möglichkeiten, die ObjectiF bietet, gibt [39]. Details über den Einsatz von ObjectiF bei Allianz gibt [35].

3.1.3.5 Planview

Planview ist ein durch den gleichnamigen Hersteller vertriebenes Werkzeug zum Projektmanagement. Bei Allianz und im Kontext des RE-Prozesses wird es zur Erstellung des Projektantrags genutzt. Es bietet keine Verknüpfungen zu den anderen Werkzeugen, die bei Allianz eingesetzt werden. Es ist also insbesondere nicht möglich Anforderungen aus AMT zu referenzieren.

3.1.4 Dokumente

In diesem Unterkapitel werden die wichtigsten, im IST-RE-Prozess erwähnten, Dokumente kurz erläutert. Wenn im Modell ein Dokumentenname im Singular formuliert ist, so kann entsprechend des Kontexts auch die Pluralform gemeint sein.

3.1.4.1 Abgeleitete Anforderung

Eine abgeleitete Anforderung beschreibt die notwendigen Tätigkeiten (z.B. „Anpassung GUI“, „neuer Domänenwert“, etc.), die für die Umsetzung der zugehörigen Basisanforderung durchgeführt werden müssen [31]. Sie entstehen in der Anforderungsanalyse-Phase durch die BO Projektleiterin bzw. den BO Projektleiter als Verfeinerung der Basisanforderungen. Abgeleitete Anforderungen existieren nicht in physischer Form, sondern werden elektronisch in AMT verwaltet. Sie werden dadurch gekennzeichnet, dass sie bei dem Attribut „Anf-Typ“ den Wert „2-Abgeleitete Anford.“ erhalten und bei dem Attribut „Referenz-Anf-ID“ die eindeutige Identifikationsnummer der zugehörigen Basisanforderung. Dies symbolisiert einen einseitigen Link.

Ein typisches Beispiel für eine abgeleitete Anforderung ist „Implementierung von neuen Tabellen für VVG für die VVG Workflowsteuerung“. Die dazugehörige Basisanforderung lautet „Erweiterung des VVG-Nebenabreden-Prozesses“.

3.1.4.2 ABS-Styleguide

Der ABS-Styleguide [40] ist eine speziell für das ABS abgestimmte Sammlung von Richtlinien zur Gestaltung der GUI. Dabei wird auf den Grundaufbau von Fenstern, die Elemente in GUIs, die Tastatursteuerung und die visuelle Gestaltung von Sichten eingegangen. Für ein Teilsystem (z.B. „Kraft“) im Rahmen des ABS ist es möglich,

einen eigenen Styleguide zu definieren, der die Richtlinien des ABS-Styleguides erweitert und noch weiter einschränkt. Die gegebenen Richtlinien können aber nicht aufgehoben oder gelockert werden. Der Styleguide nimmt zudem gesetzliche Grundlagen mit auf und wurde mit dem Betriebsrat bei Allianz abgestimmt. Er wird im Vorgehensmodell zur Prüfung der Anforderungsspezifikation, des GUI-Entwurfs und abschließend zur Prüfung der Anwendung herangezogen.

3.1.4.3 Basisanforderung

Eine Basisanforderung beinhaltet die grundlegende, fachliche Beschreibung oder gesetzliche Notwendigkeit einer Anforderung mit entsprechender Begründung des Vorhabens [31]. Im Kontext von ABS beschreibt sie eine Erweiterung der Use Cases des ABS Kern (vorliegend in ObjectiF) für den Einsatz im deutschen Umfeld.

Basisanforderungen werden, wie auch abgeleitete Anforderungen, nicht physisch, sondern elektronisch mittels AMT verwaltet. Um eine Anforderung als Basisanforderung zu kennzeichnen, wird dem Attribut „Anf-Typ“ der Wert „1-Basis Anford.“ und dem Attribut „Referenz-Anf-ID“ die eigene eindeutige Identifikationsnummer zugeordnet. Basisanforderungen entstehen unmittelbar aus den vorherrschenden Geschäftsprozessen und den Use Cases des ABS Kern. Sie beschreiben daher meistens sehr grob-granular was zu tun ist, um Defizite des ABS Kern, welche die Verwendung einer bestimmten Komponente (z.B. „Kraft“) im deutschen Umfeld der Allianz nicht möglich macht, zu beheben. Die dafür vorgesehenen Use Cases des ABS Kern werden auf diese Weise für Allianz Deutschland erweitert.

Ein typisches Beispiel für eine Basisanforderung ist „Erweiterung des VVG-Nebenabreden-Prozesses“.

3.1.4.4 Projektantrag

Der Projektantrag ist ein in Planview verwaltetes Dokument. Es beschreibt, ausgehend von einer Idee, systematisch die Notwendigkeit eines Projekts. Dabei werden Kriterien der Machbarkeit, Finanzierbarkeit, Wirtschaftlichkeit und des Marktes berücksichtigt. Er dient als Entscheidungsgrundlage für das Management. Die Grundlage für den Projektantrag ist die Erfassung von entsprechend, zur vermittelten Idee passenden, Basisanforderungen durch die BO Projektleiterin bzw. den BO Projektleiter. Eine direkte Referenzierung dieser Basisanforderungen erfolgt aus technischen Gründen jedoch nicht. Der Projektantrag selbst wird durch die Projektmanagerin bzw. den Projektmanager gestellt.

3.1.4.5 Releaseplan

Der Releaseplan ist ein Dokument, welches beschreibt, welche Basisanforderungen in welchem Release von ABS implementiert werden sollen. Der Releaseplan wird in Microsoft Powerpoint gepflegt.

3.1.5 Erläuterung des Prozesses

Im Folgenden wird der im Anhang unter Abschnitt F.2, nach der Vorgabe von Allianz in Adonis, modellierte IST-RE-Prozess vorgestellt. Eine Legende zu Adonis ist unter F.1 zu finden. Zum besseren Verständnis orientiert sich die Erläuterung des Prozesses an den in Kapitel 2.2.1 skizzierten Phasen des RE. Abb. 11 zeigt eine grobe Übersicht des IST-RE-Prozesses, gruppiert nach den Phasen des ABS Vorgehensmodells. Der eigentliche RE-Prozess ist in die ersten beiden Phasen, das sind „Vor

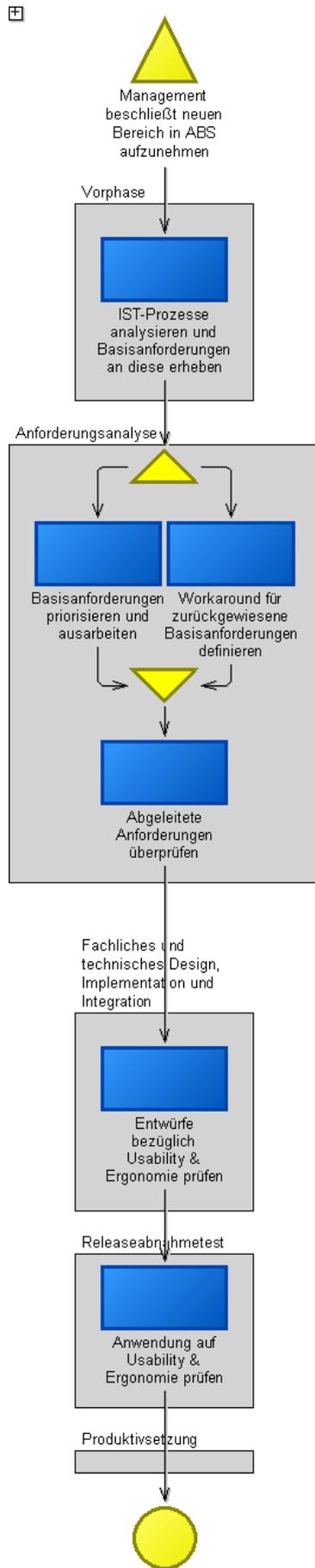


Abb. 11: Übersicht über IST-RE-Prozesses

phase“ und „Anforderungsanalyse“ einzuordnen. Weitere Aktivitäten, die zum RE-Prozess gezählt werden können, entfallen auf die anderen Phasen und sind entsprechend auch in der Übersicht zu sehen.

Die Aktivitäten des Prozesses sind zwei Rollen zugeordnet. Das ist zum einen die des anfordernden Teilprojekts, zum anderen die des umsetzenden Teilprojekts.

Erhebung von Anforderungen

Das initiale Ereignis, das den Prozess startet, ist ein Beschluss des Managements, einen neuen Bereich in ABS aufzunehmen. Das bedeutet, dass die durch den ABS Kern bereitgestellte Logik in diesem Bereich für die deutschen Verhältnisse nicht ausreicht und daher erweitert werden muss. Von dieser Tatsache ausgehend ist die BO in Form der Projektleiterin bzw. des Projektleiters im Unterprozess „IST-Prozesse analysieren und Basisanforderungen an diese erheben“ zuständig für die Analyse der bisherigen IST-Prozesse. Die Rolle des BO Projektleiters ist meist durch eine erfahrene Person besetzt, so dass diese die Geschäftsprozesse (= Prozesse des IST) schon sehr gut kennt. Sie kann daher leicht erkennen, welche Teile der Geschäftsprozesse das neue Teilprojekt in ABS abbilden will. Zudem fließen bei der Analyse die Vorgaben aus dem Zielbetriebsmodell ein, das eine Anpassung der Geschäftsstrukturen der Allianz Group vorsieht. Die Projektleiterin bzw. der Projektleiter hat die Möglichkeit, die in Adonis vorliegenden Geschäftsprozesse des IST zu konsultieren. Bei der folgenden Delta-Analyse des IST zu dem ABS-IST sind die Use Case-Diagramme aus ObjectiF, die die Verwendung des ABS Kern beschreiben, nützlich, wenngleich auch hier die Zuständigen auf ihr Wissen aufbauen. Die Projektleitung, die auch diese Tätigkeit ausführt, kennt sich sowohl mit den Bestandssystemen, also den Systemen, die noch nicht zu ABS migriert wurden, oder vor ABS existierten, und dem aktuellen ABS aus. Bei der Delta-Analyse werden, wie bei der Analyse der IST-Prozesse zuvor auch, keine späteren Benutzerinnen und Benutzer hinzugezogen. Durch die Delta-Analyse werden Basisanforderungen extrahiert, die an ABS gestellt werden müssen, damit das Teilprojekt integriert werden kann. Formal beschreiben die Basisanforderungen eine Erweiterung zu entsprechenden Use Cases aus dem ABS Kern, die speziell für Deutschland relevant sind. Wurden keine Basisanforderungen identifiziert, so wird der Unterprozess und auch der gesamte RE-Prozess an dieser Stelle abgebrochen, da keine Änderungen an ABS notwendig sind.

Im Rahmen der Erhebung von Anforderungen ist zu erwähnen, dass das bisherige Vorgehensmodell keine expliziten Aktivitäten kennt, die verschiedene Stakeholder zur Erhebung mit einbeziehen, wie z.B. Interviews, Beobachtungen oder Focus Groups. Die Notwendigkeit ein neues Teilprojekt zu initiieren, stammt entweder direkt vom Management oder kommt (das ist der Normalfall) von den Dienstleistungsgebieten. Die Angestellten der Dienstleistungsgebiete sind die späteren Benutzerinnen und Benutzer. Die BO, als vernetzende und organisatorische Einheit, vertritt die Dienstleistungsgebiete während des gesamten Prozesses als Kunde. Eine direkte Beteiligung der letztlichen Benutzerinnen und Benutzer gibt es nicht.

Dokumentation von Anforderungen

Mit der ersten Aktivität welche AMT mit einbezieht – das ist hier „Basisanforderungen erfassen“ –, beginnt die RE-Phase „Dokumentation von Anforderungen“. Die von der BO Projektleitung in der Delta-Analyse identifizierten Basisanforderungen werden nun durch selbige systematisch in AMT festgehalten. Im Anschluss stellt die Projektmanagerin bzw. der Projektmanager einen Projektantrag mittels Planview. Im Rahmen des Ausfüllens des Projektantrags kreuzt die Projektmanagerin bzw. der Projektmanager an, ob das geplante Vorhaben Usability-relevant ist oder nicht, d.h. ob Änderungen oder Ergänzungen an der GUI vorgenommen werden müssen. Diese Entscheidung ist für den weiteren Fortlauf des Prozesses wichtig, da dadurch einige Aktivitäten hinzu kommen bzw. ausgelassen werden können und diese auch im Zeitplan des Teilprojekts berücksichtigt werden müssen. Mit dem Stellen des Projektantrags und dessen Genehmigung ist die Vorphase des ABS Vorgehensmodells von oben beendet.

Der nächste Schritt im Prozess ist die Priorisierung der Basisanforderungen. Dies geschieht in der Rolle des umsetzenden Teilprojekts. Die Aktivitäten zuvor sind alle dem anfordernden Teilprojekt verschrieben. Die Priorisierung erfolgt in AMT und kann drei Werte annehmen: „1“ für hoch, „2“ für mittel und „3“ für niedrig. Dabei definiert sich eine Anforderung mit Priorität „1“ als „für das Projekt essentiell wichtig und nicht verhandelbar“ [31], Anforderungen mit Priorität „2“ als „nicht essentiell erfolgskritisch“, wenngleich sie dennoch eine „hohe Relevanz“ haben und umgesetzt werden sollten [31]. Anforderungen mit Priorität „3“ haben „geringe Relevanz“ und sollen erst implementiert werden, wenn keine wichtigeren Anforderungen übrig und noch Kapazitäten vorhanden sind [31]. Wie genau der Wert vergeben wird, orientiert sich meist an der Dringlichkeit der Notwendigkeiten, die durch die jeweiligen Dienstleistungsgebiete formuliert werden. Sollte von dieser Quelle keine Information diesbezüglich mitgeliefert worden sein, so kann dort nachgefragt werden oder die BO Projektleiterin bzw. der BO Projektleiter priorisiert nach ihrem bzw. seinem eigenen besten Wissen. Verantwortlich ist aber die sogenannte Programmleitung. In jedem Fall muss der Releaseplan für das umsetzende Teilprojekt hinzugezogen werden, weshalb die Durchführung der Aktivität auch dem umsetzenden Teilprojekt zufällt. Neben den in AMT vorgegebenen Werten zur Priorisierung, kann es passieren, dass Anforderungen zurückgewiesen werden. Dies ist für die Folgeaktionen, beschrieben im Rahmen der nächsten RE-Phase, wichtig.

Prüfung und Abstimmung von Anforderungen

Nach der Priorisierung werden parallel zwei Wege beschritten. Dabei wird zwischen akzeptierten und zurückgewiesenen Anforderungen unterschieden. Dies bezieht sich darauf, ob die Anforderungen in AMT mit einem Wert aus „1“, „2“ und „3“ bewertet wurden (→ akzeptiert), oder nicht und stattdessen dem Typ „3-Ablaufänderung“ zu-

gewiesen wurden (→ zurückgewiesen). Für letztere definiert die BO Projektleiterin bzw. der BO Projektleiter des anfordernden oder des umsetzenden Teilprojekts, je nach höherem Fachwissen, für jede dieser Basisanforderungen einen fachlichen Workaround. Dies wird auch in AMT festgehalten. Die betroffenen Basisanforderungen werden im weiteren Prozessablauf nicht weiter betrachtet.

Für Anforderungen mit hohem Nutzen wird die Aktivität „Basisanforderungen ausarbeiten“ im Unterprozess „Basisanforderungen ausarbeiten“ ausgeführt. Dabei werden, basierend auf den Basisanforderungen, durch die BO Projektleiterin bzw. den BO Projektleiter nach eigenem Ermessen abgeleitete Anforderungen erhoben und dokumentiert, die die Tätigkeiten beschreiben, die notwendig sind, um die jeweilige Basisanforderung umzusetzen. Es ist besonders hervorzuheben, dass diese abgeleiteten Anforderungen nicht priorisiert werden, sondern in der Priorisierung direkt an die Basisanforderung gekoppelt sind. Ist die Basisanforderung also z.B. von hoher Priorität (in AMT mit dem Wert „1“), so müssen auch alle von ihr abgeleiteten Anforderungen unbedingt implementiert werden. Im Anschluss an die Ausarbeitung der Basisanforderungen werden alle abgeleiteten Anforderungen im Unterprozess „Anforderungen überprüfen“ überprüft. Dabei werden unter der Verantwortung der sogenannten Facharchitektur parallel vier Tätigkeiten ausgeführt. Im Sinne der Qualitätssicherung der Anforderung wird die Spezifikation entsprechend für Usability- und Ergonomiekonformität geprüft. Dabei ist es wichtig, dass die Anforderungen nicht gegen die firmeninternen Richtlinien, verkörpert durch den ABS-Styleguide, verstoßen. Die Prüfung wird durch eine zuvor benannte Usability-Verantwortliche bzw. einen Usability-Verantwortlichen der BO durchgeführt. Neben dieser Prüfung werden die Auswirkungen auf ein mit ABS verwandten System namens Agentur Management- und Informationssystem (kurz AMIS) geprüft. AMIS ist das Clientprogramm, das Sachbearbeiterinnen und Sachbearbeitern zur Verfügung gestellt wird, die im Außendienst tätig sind. AMIS greift dabei auf dieselben Datenbestände wie ABS zu. Diese gemeinsame Schnittstelle muss in dieser Aktivität berücksichtigt werden.

Die beiden verbleibenden Aktivitäten beschreiben den Abstimmungsprozess zwischen dem ABS Kern und der ADAG und die letztliche Planung der Implementation. Auch das ist jedoch für die Spezifikation und damit für die Qualität der Anforderungen wichtig, da letztlich die Dokumentation in AMT entscheidend für das weitere Vorgehen ist. Um die Überprüfung abzuschließen, werden Anforderungen, die fachliche oder technische Änderungen an der Architektur beschreiben, im Architekturstab vorgestellt, der für die Organisation von Modulen in ABS verantwortlich ist. Erfüllen Anforderungen nicht die Maßgaben bzw. fallen bei einer der Prüfungen durch, so müssen sie überarbeitet werden. Erst wenn alle Anforderungen die Prüfung überstehen, ist die Anforderungsanalyse im Sinne des ABS Vorgehensmodells beendet. Auch die RE-Phase der Prüfung und Abstimmung von Anforderungen ist somit beendet.

Weiteres Vorgehen

Bevor noch kurz auf die übergreifende RE-Aktivität „Verwaltung von Anforderungen“ eingegangen wird, wird zunächst noch der modellierte Prozess bis zum Ende erläutert. Der eigentliche RE-Prozess endet zwar hier, doch wurde sich in Absprache mit Allianz dafür entschieden den Rest des ABS Vorgehensmodells noch grob zu skizzieren, um noch für den RE-Prozess relevante Tätigkeiten mit aufzunehmen. Dies ist auch besonders für den Entwurf des SOLL-RE-Prozesses in Kapitel 4 wichtig, um den Zusammenhang aller Tätigkeiten zu dem jetzigen Stand bei Allianz transparent machen zu können und auch Tätigkeiten zu erfassen, die für die Betrachtungen zur Gebrauchstauglichkeit relevant sind.

Sind die Anforderungen alle aufgenommen, ausgearbeitet und geprüft, so werden im Rahmen des umsetzenden Teilprojekts Anforderungen betreffend dem ABS Kern in Österreich durch die dort zuständigen Teams entworfen, implementiert und getestet. Der ABS Kern betrifft alle Teile der Allianz Group und muss daher zentral verwaltet werden. Im Modell wurde absichtlich kein Verantwortlicher genannt, da dies aus Sicht der ADAG nicht transparent ist und zudem auch keine Rolle spielt. Für Teile, die nur die ADAG betreffen, wird das sogenannte Work Team zunächst die Anforderungen studieren und schließlich akzeptieren, um zu signalisieren, dass alle Anforderungen verstanden wurden, sie also „ready“ sind (siehe 3.1.1.2). Der Entwicklungsprozess soll so möglichst nicht durch Verständnisprobleme aufgehalten werden. Als nächstes folgt der Entwurf der Anforderungen unter Zuhilfenahme von ObjectiF. Insofern das Vorhaben am Anfang des Prozesses durch die Projektmanagerin bzw. den Projektmanager als Usability-relevant eingestuft wurde, folgt auf den Entwurf eine Qualitätssicherung des GUI-Designs, um die Maßgaben des ABS-Styleguides, der gesetzlichen Rahmenbedingungen und im Zweifel auch des projektspezifischen Styleguide auf Einhaltung zu überprüfen. Sollten dabei durch die Usability-Verantwortliche bzw. den Usability-Verantwortlichen der IT Probleme im Entwurf festgestellt werden, so muss der Entwurf überarbeitet werden. Sind keine Probleme aufgefallen, so wird mit der Implementation und gleichzeitigen Spezifikation der Testfälle fortgefahren. Beide Aktivitäten werden durch dieselben IT Entwicklerinnen und Entwickler des Work Teams durchgeführt. Die Spezifikation der Testfälle erfolgt nach den in AMT spezifizierten Abnahmekriterien zu jeder abgeleiteten Anforderung. Dazu wird HP Quality Center benutzt. Nach der Umsetzung und der Testspezifikation erfolgt ein abschließender Komponenten- und Integrationstest, der, bei Entdeckung von Fehlern, wieder in der Umsetzung und Spezifikation von Testfällen mündet. Andernfalls ist der Unterprozess beendet.

Bevor das anfordernde Teilprojekt eine fachliche Prüfung der Funktionalität durchführt, müssen alle Entwicklungsaktivitäten in der ADAG und in Österreich beendet worden sein und zur Verfügung stehen. Dann wird der Test durch die fachlichen Expertinnen und Experten aus dem Work Team durchgeführt. Sollten dabei Probleme festgestellt werden, so muss an geeigneter Stelle eine Korrektur erfolgen. Dazu wurde im Modell ein Pfad zurück bis zum Unterprozess „Basisanforderungen ausarbeiten“ gezogen, der letztlich den Worst-Case darstellt. Sollte die Ursache nicht direkt bei diesem Unterprozess liegen, so können Aktivitäten geeignet übersprungen werden, um zur Ursache zu gelangen.

Nach dem fachlichen Test führt eine fachlicher Testerin oder ein fachlicher Tester, diesmal unabhängig vom Work Team, einen „Releaseabnahmetest“ (das ist ein gewöhnlicher Abnahmetest) durch. Dazu werden alle in AMT spezifizierten Abnahmekriterien herangezogen. Sollte das Teilprojekt Usability-relevant gewesen sein, so wird nachfolgend durch die IT-Usability-Verantwortliche bzw. den IT-Usability-Verantwortlichen noch eine abschließende Prüfung der Ergonomie und Usability durchgeführt. Hierbei kann die Prüfung jedoch unterschiedlicher Natur sein. Von Fall zu Fall werden, wie in den analogen Aktivitäten zuvor, nur ABS-Styleguide etc. auf Einhaltung geprüft. Es kann aber auch sein, dass ein literaturgemäßer Usability-Test durchgeführt wird, der die Einbeziehung von späteren Benutzerinnen und Benutzern vorsieht. Wie genau entschieden wird, welche Teilaktivitäten unter dieser abschließenden Prüfung subsumiert werden, unterliegt keiner Systematik. Es wird von Fall zu Fall entschieden, ob ein Usability-Test einen „Mehrwert“ nach Meinung der Usability-Verantwortlichen erbringt. Im Zweifel ist dieser Schritt aber der einzige, bei dem die späteren Benutzerinnen und Benutzer der entwickelten Komponente für ABS einbezogen werden.

Verwaltung von Anforderungen

Die RE-Aktivität „Verwaltung von Anforderungen“ ist im gesamten RE-Prozess gegenwärtig. Die systematische Dokumentation in AMT ist ein Aspekt davon. AMT bietet zahlreiche Attribute, die für Anforderungen ausgefüllt werden können. Auch läuft die Priorisierung von Anforderungen über AMT.

3.2 Bewertung

Als Vorarbeit zur Bewertung des IST-RE-Prozesses werden in Unterkapitel 3.2.1 nach dem GQM-Ansatz die von Allianz gesetzten Ziele mit Fragen konkretisiert und

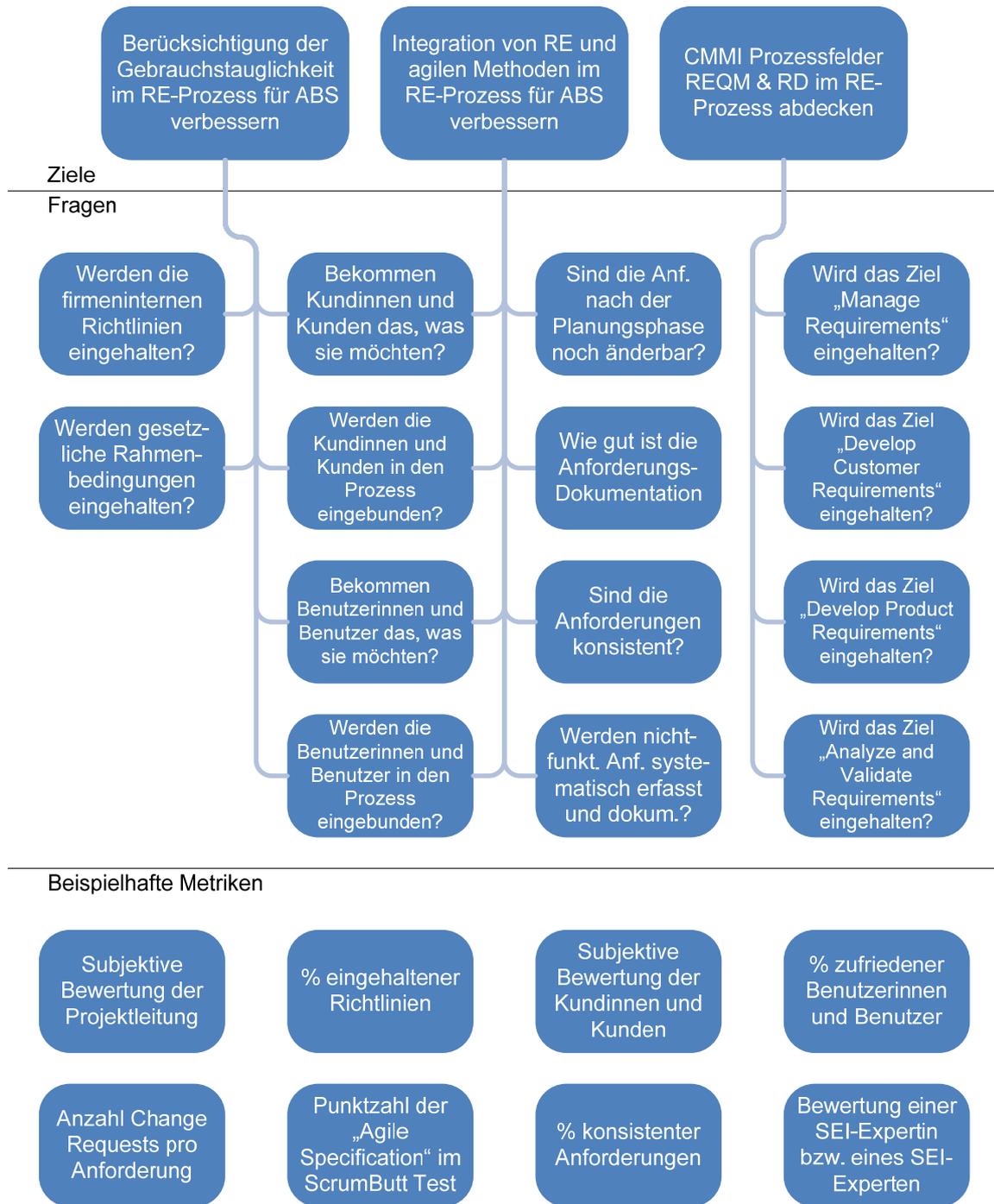


Abb. 12: GQM-Plan zu SPI

einige Vorschläge für Metriken zur Messung gegeben. In Kapitel 3.2.2 wird der Kriterienkatalog erläutert, der basierend auf den beispielhaften Metriken ausgearbeitet wurde. Dieser Kriterienkatalog dient als Grundlage für eine grobe Messung des RE-Prozesses, deren Ergebnisse in Abschnitt 3.2.3 vorgestellt werden. Die Diskussion dieser Ergebnisse folgt schließlich in Abschnitt 3.2.4.

3.2.1 GQM-Plan

Aus dem zuvor vorgestellten IST-RE-Prozess und mittels der vorgestellten SPI-Vorgehensweise GQM konnte der in Abb. 12 gezeigte Plan von Zielen und Fragen abgeleitet werden, der mit Allianz abgestimmt wurde. Um eine Messung des RE-Prozesses durchführen zu können, werden beispielhaft einige Metriken genannt. Die genauen Metriken sind dem Kriterienkatalog im Anhang unter F.5 zu entnehmen und werden in Kapitel 3.2.2 genauer beschrieben.

Die beiden aufgeführten Ziele „Berücksichtigung der Gebrauchstauglichkeit im RE-Prozess für ABS verbessern“ und „Integration von RE und agilen Methoden im RE-Prozess für ABS verbessern“ konnten unmittelbar aus den Zielen dieser Arbeit übernommen werden - die Bearbeitung dieser Teilgebiete wird in Kapitel 4.2 bzw. 4.1 erläutert. Hinzu kommt das Ziel der Abdeckung der Prozessfelder Requirements Management (REQM) und Requirements Development (RD) aus dem CMMI Modell für Entwicklung (siehe [41]). Dieses letzte Ziel entstammt der Verwendung von CMMI als Referenz bei der Qualitätssicherung der Softwareentwicklungsprozesse bei Allianz.

Für das Ziel der Gebrauchstauglichkeitsverbesserung wurden sechs Fragen identifiziert (erkennbar an den Verbindungslinien), die das Erreichen des Ziels genauer messbar machen sollen. Darunter sind zum einen Fragen, die auf den bisherigen Stand bei Allianz verweisen, z.B. Einhaltung des Styleguides, aber auch typische, aus der Literatur bekannte Fragen nach der Zufriedenheit der Benutzerinnen und Benutzer. Interessant dabei sind vor allem die Belange der Kundinnen und Kunden im Sinne der Auftraggeber und die der späteren Benutzerinnen und Benutzer zu vereinen und so beide Stakeholder-Gruppen zufrieden zu stellen. Alle sechs Fragen spiegeln direkt wider, wie Allianz Gebrauchstauglichkeit misst, stimmen aber gleichermaßen in den Kernaspekten mit den Zielsetzungen von Gebrauchstauglichkeit in der Literatur überein, nämlich die Benutzerinnen und Benutzer zufriedenzustellen.

Die Frage nach der Zufriedenheit der Kundinnen und Kunden ist auch ein Bestandteil des Ziels der Integration von RE und agilen Methoden, denn das Ergebnis eines Entwicklungsprozesses hängt unmittelbar von der Art und Weise des RE ab. Insbesondere die Frage nach der Änderbarkeit von Anforderungen nach deren Erhebung ist eng mit der Zufriedenheit der Kundinnen und Kunden verknüpft, genauso wie die Frage nach der Einbindung der Kundinnen und Kunden in den Prozess an sich. Diese genannten Aspekte stammen von Allianz und beschreiben daher primär deren Vorstellung der Aspekte einer Integration von RE und agilen Methoden. Ein Merkmal für die Qualität des agilen Vorgehens und daher auch als Fragen formuliert, sind Aussagen über die Anforderungsdokumentation (inkl. Konsistenz) und die Beteiligung von nicht-funktionalen Anforderungen. Diese Punkte konnten aus der angestellten Literaturrecherche zur Integration von RE und agilen Methoden, ausgeführt unter Kapitel 2.2.3, extrahiert werden und symbolisieren die dort genannten Verbesserungspotentiale.

Die Abdeckung der Prozessfelder REQM und RD aus CMMI wird nach [41] durch die dort definierten Ziele konkretisiert.

3.2.2 Kriterien

Aus den formulierten Fragen des GQM-Plans wurden Bewertungskriterien abgeleitet und in einem Katalog zusammen gefasst (siehe F.5 im Anhang). Dieser ordnet, ähnlich dem GQM-Plan, jedem Kriterium ein Ziel zu und bietet zusätzlich konkret eine oder mehrere Metriken zur Messung an. So kann für jede Metrik ein „Score“ errechnet werden, der in einer Skala zwischen 0% und 100% liegt. 0% entspricht dabei der schlechtesten Bewertung, z.B. wenn eine geforderte Aktivität im Prozess nicht vorhanden ist, und 100% entspricht der besten. Aus der Summierung der Scores ergibt sich eine Gesamtbewertung, die maximal 28 Punkten entsprechen kann.

Das Ziel der Gebrauchstauglichkeit wird mit Kriterien zur Einhaltung von Richtlinien und Kriterien zur Zufriedenheit der Kundin bzw. des Kunden sowie der Benutzerinnen und Benutzer abgedeckt. Zur Bewertung der Kriterien bezüglich Richtlinien soll geprüft werden, ob im RE-Prozess eine Aktivität zur Prüfung auf die jeweilige Richtlinie vorgesehen ist. Um die Zufriedenheit der Kundin bzw. des Kunden und der Benutzerinnen und Benutzer zu bewerten, soll eine subjektive Bewertung der Kundin bzw. des Kunden und der Benutzerinnen und Benutzer eingeholt werden. Dies soll mittels eines Fragebogens geschehen, der verschiedene Teilbereiche hinsichtlich Zufriedenheit abhandelt. Als Grundlage kann [42] dienen, muss aber noch um den Bereich der RE-Prozess-Einbindung erweitert werden. Dies soll explizit aufzeigen, ob sich Kundin bzw. Kunde und die späteren Benutzerinnen und Benutzer ausreichend eingebunden fühlen und ob dies in Kontrast zu der Zufriedenheit mit dem fertigen Produkt steht. Um hier einen „Score“ zu berechnen, soll eine Formel entwickelt werden, die die einzelnen Teilbereiche anteilig berücksichtigt und zu einem Wert zwischen 0% und 100% für die „Gesamtzufriedenheit“ umwandelt.

Bei den Kriterien zu RE und agilen Methoden wird auch die Zufriedenheit der Kundin bzw. des Kunden und der Benutzerinnen und Benutzer herangezogen, aber auch die Einbindung des Product Owners und durch diesen auch der Benutzerinnen und Benutzer. Hierzu soll „Question 4 - Product Owner“ des ScrumButt Test [43], mit dessen Ergebnis dividiert durch 10, und die Prozentzahl der eingebundenen Stakeholder, in Bezug auf alle relevanten Stakeholder, dienen. Dazu soll jeder relevante Stakeholder eine subjektive Bewertung abgeben. Die Identifikation der relevanten Stakeholder soll eine RE-Expertin oder ein RE-Experte vornehmen. Darüber hinaus wurden die regelmäßige Anpassbarkeit der Anforderungen, die Qualität der Anforderungsdokumentation, die Konsistenz der Anforderungen, die Berücksichtigung von Abhängigkeiten zwischen Anforderungen und die systematische Erfassung nicht-funktionaler Anforderungen als Kriterien festgehalten. Zur Messung kommen unter anderem der ScrumButt Test, wie auch die Prüfung, ob eine bestimmte Aktivität im Prozess vorgesehen ist, zum Einsatz.

Das letzte Ziel der Abdeckung der Prozessbereiche REQM und RD aus CMMI wird durch die jeweiligen definierten CMMI-Ziele konkretisiert. Deren Bewertung erfolgt durch die Prüfung auf die Ausführung der dazu definierten „Practices“. Diese Bewertung soll durch eine CMMI-zertifizierte Expertin oder einen CMMI-zertifizierten Experten erfolgen.

3.2.3 Messung

Basierend auf dem Kriterienkatalog und den darin festgehaltenen Metriken wurde zusammen mit dem Betreuer bei Allianz eine grobe Bewertung durchgeführt - eine detaillierte und genaue Bewertung ist im Rahmen dieser Arbeit nicht möglich -, um den zuvor vorgestellten IST-RE-Prozess einordnen zu können und zugleich Schwachpunkte zu identifizieren. Dabei wurde der IST-RE-Prozess einmal bezüglich eines traditionellen Vorgehens bewertet [44] und einmal bezüglich eines agilen Vorgehens [45]. Eine Übersicht über die beiden genannten Bewertungen gibt Tabelle 5. Der IST-RE-Prozess erreicht demnach 18,3 bzw. 21,25 der 28 Punkte. Im Folgenden werden die hervorzuhebenden Ergebnisse kurz zusammengefasst.

Die Zufriedenheit der Kundinnen und Kunden wurde auf ca. 75% geschätzt, die der späteren Benutzerinnen und Benutzer auf 25%. Die Einbindung des Product Owners (bzw. bei Allianz der Projektleiterin oder des Projektleiters im traditionellen Vorgehen) wurde mit 10% bewertet, da so nach ScrumButt Test nur die Existenz einer solchen Rolle zählt, wohingegen sich der Product Owner im agilen Vorgehen für ein klares Product Backlog, deren Anforderungen bezüglich Aufwand durch das Team geschätzt wurde, verantwortlich zeigt (50%). Der Product Owner vernachlässigt in beiden Vorgehen die Benutzerinnen und Benutzer als relevanten Stakeholder, weshalb dort eine Bewertung von 90% geschätzt wurde, da sonst alle relevanten Stakeholder einbezogen werden. Die Qualität der Anforderungsdokumentation wird im Sinne von Scrum mit nur 10% bewertet („big requirements document“) – im agilen Vorgehen bei Allianz mit 50% („good requirements“). Es fehlt an *ausreichendem* Detaillierungsgrad zum Zeitpunkt der Implementation. Bei Allianz kommt es durchaus vor, dass aufgrund von Zeitdruck Anforderungen nur spärlich (und auch im agilen Vorgehen, nicht mit User Stories) beschrieben sind. Die Anpassbarkeit der Anforderungen ist nur im agilen Vorgehen gegeben. Die Konsistenz der Anforderungen kann bei Allianz nicht durch ein Werkzeug (zur Syntaxprüfung) überprüft werden; es ist jedoch eine Aktivität zum Review der Anforderungen vorgesehen, die schätzungsweise in 75% der Fälle (unter anderem) auch zur Konsistenzprüfung dient. Abhängigkeiten zwischen Anforderungen werden zu 75% im Projektrahmen berücksichtigt.

Bei CMMI werden alle genannten Practices bis auf „Validate Requirements“ (Prüfung auf geplante Nutzbarkeit im Benutzerkontext) ausgeführt, wenngleich einige nicht mit 100% bewertet wurde, da die Benutzerinnen und Benutzer nicht einbezogen werden. Die Practice „Establish a Definition of Required Functionality“ wurde aufgrund von Kapazitätsproblemen in der Durchführung nur mit 75% bewertet. „Analyze Requirements“ ist zu restriktiv und verhindert die Umsetzung durchaus relevanter und gewinnbringender Anforderungen, auch aufgrund der restriktiven Umsetzung des ABS Kerns und wurde daher mit 50% bewertet. „Obtain Commitment to Requirements“ ist durch die engen Rahmenbedingungen des Managements gehemmt und wurde daher ebenfalls nur mit 50% bewertet. Hier spielen vor allem zu knappe Zeitvorgaben eine Rolle, so dass sich ändernde Anforderungen auf einen starren Projektplan stoßen.

	Kriterium	Bewertet durch ...	Score	
			traditionell	agil
Gebrauchstauglichkeit	Einhaltung der firmeninternen Richtlinien	Aktivität zur Prüfung im RE-Prozess vorgesehen?	1	1
	Einhaltung der gesetzlichen Rahmenbedingungen	Aktivität zur Prüfung im RE-Prozess vorgesehen?	1	1
	Zufriedenheit der Kundin bzw. des Kunden <i>(betrifft auch RE+agile Methoden)</i>	Subjektive Bewertung der Kundin bzw. des Kunden mittels Fragebogen*	0,75	0,75
	Zufriedenheit der Benutzerinnen und Benutzer <i>(betrifft auch RE+agile Methoden)</i>	Subjektive Bewertung der Benutzerinnen und Benutzer mittels Fragebogen*	0,25	0,25
Requirements Engineering + agile Methoden	Einbindung des Product Owners & der Benutzerinnen und Benutzer	ScrumButt Test (Question 4 – Product Owner)	0,2	0,5
		% aller relevanten Stakeholder eingebunden; subjektive Bewertung der jeweiligen Stakeholder*	0,9	0,9
	Anpassbarkeit der Anforderungen	Aktivität zur Anpassung von Anforderungen vorgesehen?	0	1
	Qualität der Anforderungsdokumentation	ScrumButt Test (Question 3 – Agile Specification)	0,1	0,5
	Konsistenz der Anforderungen	Automatische Syntaxprüfung durch verwendete Tools möglich und in RE-Prozess integriert?	0	0
		Aktivität zum Review der Anforderungen durch Product Owner im RE-Prozess vorgesehen?*	0,75	0,75
	Berücksichtigung von Abhängigkeiten zwischen Anforderungen	Aktivität zur Prüfung von Abhängigkeiten zwischen Anforderungen für gesamtes Projekt vorhanden?	0,75	0,75
		... jeden Sprint vorhanden?	0	1
	Systematische Erfassung nicht-funktionaler Anforderungen	Werden nicht-funktionale Anforderungen systematisch erfasst und dokumentiert?	1	1
	Abdeckung des CMMI-Ziels „Develop Customer Requirements“	Wird Practice „Elicit Needs“ ausgeführt?	0,75	0,75

CMMI		Wird Practice „Develop the Customer Requirements“ ausgeführt?	1	1	
	Abdeckung des CMMI-Ziels „Develop Product Requirements“	Wird Practice „Establish Product and Product Component Requirements“ ausgeführt?	1	1	
		Wird Practice „Allocate Product Component Requirements“ ausgeführt?	1	1	
		Wird Practice „Identify Interface Requirements“ ausgeführt?	1	1	
	Abdeckung des CMMI-Ziels „Analyze and Validate Requirements“	Wird Practice „Establish Operational Concepts and Scenarios“ ausgeführt?	1	1	
		Wird Practice „Establish a Definition of Required Functionality“ ausgeführt?	0,75	0,75	
		Wird Practice „Analyze Requirements“ ausgeführt?	0,5	0,5	
		Wird Practice „Analyze Requirements to Achieve Balance“ ausgeführt?	0,8	0,8	
		Wird Practice „Validate Requirements“ ausgeführt?	0	0	
	Abdeckung des CMMI-Ziels „Manage Requirements“	Wird Practice „Obtain an Understanding of Requirements“ ausgeführt?	0,8	0,8	
		Wird Practice „Obtain Commitment to Requirements“ ausgeführt?	0,5	0,75	
		Wird Practice „Manage Requirements Changes“ ausgeführt?	1	1	
		Wird Practice „Maintain Bidirectional Traceability of Requirements“ ausgeführt?	0,5	0,5	
		Wird Practice „Identify Inconsistencies Between Project Work and Requirements“ ausgeführt?	1	1	
	* Bewertung wurde explizit geschätzt				
	Gesamtscore (maximal 28 Punkte):			18,3	21,25

Tabelle 5: Bewertung des IST-RE-Prozesses bezüglich traditionellem und agilen Vorgehen

3.2.4 Schlussfolgerungen

In Hinblick auf das Erfüllen der genannten Ziele und im Sinne einer Analyse des IST-RE-Prozesses basierend auf den Erkenntnissen aus der Messung werden nachfolgend Probleme genannt, welche durch eine Lösung zur Verbesserung des Prozesses beitragen können:

Benutzerinnen und Benutzer werden nur wenig bis gar nicht einbezogen.

Die Benutzerinnen und Benutzer bekommen nicht das, was sie gerne möchten. Sie sind unzufrieden und das liegt daran, dass sie, wenn überhaupt, erst ganz am Ende des Entwicklungsprozesses mit einbezogen werden, wenn ein Usability-Test durchgeführt wird. Doch zu diesem Zeitpunkt sind die Anforderungen nicht nur erhoben, sondern auch umgesetzt. Es wäre nur noch möglich ein neues Projekt zur Verbesserung der Usability zu initiieren, denn das eigentliche Projekt ist in dieser Hinsicht abgeschlossen.

Bezüglich des Kriterienkatalogs zur Bewertung des RE-Prozesses kann die Verbesserung dieser Problematik eine Steigerung von maximal 2,55 Punkten zur Folge haben.

Praktiken der Dokumentation und Analyse werden vernachlässigt.

Wie schon in Kapitel 2.2.3 erwähnt, besteht bei agilen Vorgehen die Gefahr, dass vor allem die Kundeneinbindung, die Dokumentation, die Analyse & Granularität von Anforderungen und nicht-funktionale wie auch übergreifende Anforderungen nicht genügend behandelt werden. Doch auch im traditionellen Vorgehen kann das als Orientierung dienen und tatsächlich konnten Defizite diesbezüglich festgestellt werden, wie die Messung belegt. Demnach bestehen in den Bereichen der Dokumentation und der Analyse Probleme, die darauf zurückzuführen sind, dass die eingesetzten Werkzeuge keine ausreichende Unterstützung liefern. Beispielsweise werden Abhängigkeiten zwischen Anforderungen nur schlecht dargestellt, Links sind oft einseitig und nicht visualisiert, sondern lediglich durch Daten erfasst, es ist also schwer ihnen zu folgen. Auch der RE-Prozess kennt keine explizite Tätigkeit z.B. zur Verifikation von Anforderungen.

Durch eine Verbesserung kann eine Bewertungssteigerung von maximal 2,0 Punkten im Sinne des Kriterienkatalogs erreicht werden.

Neben den prozessbezogenen Problemen gibt es auch einige strukturelle und organisatorische Probleme, die aber nicht durch eine Optimierung des RE-Prozesses behoben werden können.

Das Management räumt den Teilprojekten „unrealistische“ Zeitrahmen ein, so dass oft wichtige Schritte im RE-Prozess, aber auch im gesamten Vorgehensmodell, ausgelassen werden müssen. Darunter fällt z.B. die Ausarbeitung der Anforderungen, so dass sie *nicht* vollständig durch die Entwicklerinnen und Entwickler im Work Team verstanden werden. Aufgrund dieser teilweise sehr starren Rahmenbedingungen ist es zudem nicht immer möglich, auf sich ändernde Anforderungen zu reagieren. Die Kapazitäten reichen oft nicht aus.

Der zentral bereitgestellte ABS Kern kann, weil er vielen einzelnen Länderbedürfnissen genügen muss, nicht die gleichen technischen Voraussetzungen, wie einige Bestandssysteme, bieten, so dass einige Detailfeatures, die bei Bestandssystemen zur Effektivität beitragen, nicht in ABS übernommen werden können. Dies kann z.B. eine datengestützte Berechnung sein, die im Bestandssystem durch das Softwaresystem

durchgeführt wurde und in ABS nun manuell durch die Benutzerinnen und Benutzer durchgeführt werden muss, weil es technisch nicht mehr möglich ist.

3.3 Agile Methoden

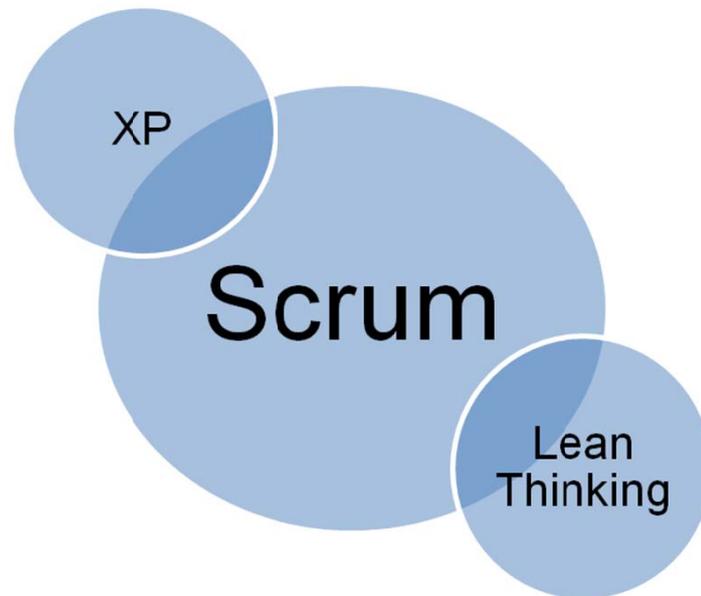


Abb. 13: „Agile“ bei Allianz

Der Begriff „Agile“ ist bei Allianz eine Symbiose von unterschiedlichen Paradigmen. Den Kern bildet Scrum, um das Projektmanagement zu organisieren. Dazu wurden alle Praktiken von Scrum übernommen. Der Fokus liegt darüber hinaus auf Hochleistungs-Teams, das sind kleine Teams bestehend aus qualifizierten Personen, und der stetigen, direkte Zusammenarbeit mit dem Kunden. Bei Allianz bedeutet dies: die Zusammenarbeit mit der Betriebsorganisation (BO). Die Praktiken bei Scrum (hier besonders das Daily Scrum) sorgen dafür, dass die Teams von jeglichen Arbeitshindernissen befreit werden und so ihr volles Potential entfalten können.

Für die Entwicklung wird eXtreme Programming (XP) angewandt. Dazu zählen vor allem die Verwendung von User Stories und des Planning Games. Zusätzlich bedeutet „Agile“ für Allianz, der Philosophie von „Lean“ zu folgen, d.h. sich kontinuierlich zu verbessern und Aktivitäten ohne direkten Nutzen (Verschwendung von Ressourcen) zu minimieren [46].

Allianz motiviert die Einführung von agilen Methoden mit einer Verbesserung des Return of Investment, einer schnelleren Reaktion auf geänderte Anforderungen und einer verbesserten Kundenzufriedenheit durch kontinuierliche Interaktion und Lieferung. Darüber hinaus durch frühes Identifizieren und Ausräumen von Risiken, einem nachhaltigen Arbeitstempo ohne eine Zermürbung der Mitarbeiterinnen und Mitarbeiter und einer Verbesserung der Softwarequalität mit dem Ergebnis von verringerten Lebenszyklus-Kosten [47]. Im Zuge der Umstellung auf „Agile“, wurden und werden die Prozesse bei Allianz angepasst. Allianz ist zum Zeitpunkt dieser Arbeit dabei, umzustellen. Das folgende Kapitel erklärt die Anpassungen an den Prozessen bei Allianz.

3.3.1 Anpassungen der Prozesse durch Agile

3.3.1.1 ABS Vorgehensmodell

Das ABS Vorgehensmodell ist der Hauptangriffspunkt für die Anpassungen. Basierend auf der Übersicht zum Modell (siehe Abb. 9 auf Seite 26) wird der dargestellte Zyklus des fachlichen und technischen Designs, der Implementation und Integration im Rahmen von Sprints aus Scrum durchgeführt. Dazu ist es nicht mehr notwendig, dass alle Anforderungen aus der Anforderungsanalyse auf einmal „ready“ sein müssen, sondern nur der für den Sprint relevante Teil. Auch die Ausarbeitung der Anforderungen wird auf den Sprint verlagert, so dass das „just enough“-Prinzip aus Scrum zum Tragen kommt.

In den anderen Phasen des ABS Vorgehensmodells wird kein agiles Vorgehen angedacht. Das Ziel dieser Arbeit ist es unter anderem, agile Elemente, wenn möglich, auch in der Vorphase und der Anforderungsanalyse einzusetzen, so dass weite Teile des gesamten ABS Vorgehensmodells agil werden.

3.3.1.2 RE-Prozess

Der RE-Prozess ist bislang nicht zwangsläufig agil. Einige, von Scrum begeisterte, Teilprojektteams führen ihn allerdings bereits agil aus. In Sprints werden Anforderungen erfasst, priorisiert und zumindest teilweise ausgearbeitet.

Dadurch, dass der RE-Prozess im Rahmen des gesamten ABS Vorgehensmodell zu sehen ist und dieser zum Teil mit agilen Methoden ausgestattet wird, treten daher auch im RE-Prozess einige neue Artefakte auf. So kommt zusammen mit dem Projektantrag eine „Product Vision“ auf [48]. Diese wird durch den Product Owner (= Projektleiterin bzw. Projektleiter) entwickelt und an die Beteiligten vermittelt. Sie verkörpert die Idee, die auch dem Projektantrag zugrunde liegt. Aus diesem Grund hängen beide Artefakte bei Allianz eng zusammen.

Die Product Vision dient als Grundlage für das Product Backlog, welches eine Sammlung aller Anforderungen und Arbeitsergebnisse ist, die zur Erreichung des Projektziels umgesetzt oder erbracht werden müssen [11]. Im Gegensatz zur Product Vision kann das Product Backlog während des gesamten Prozesses geändert und erweitert werden, wohingegen die Product Vision nach Definition (zum Zeitpunkt der Projektantragsstellung) stabil bleibt.

4 Vorschlag für SOLL-RE-Prozess

Basierend auf den in Kapitel 3.2.4 gezogenen Schlussfolgerungen aus dem vorgestellten IST-RE-Prozess und den zusätzlich erfassten Zielen der Optimierung, befasst sich dieser Abschnitt der Arbeit mit der Optimierung der Mängel, um einen Entwurf für einen neuen, verbesserten RE-Prozess machen zu können. Dazu wird im Unterkapitel 4.1 eine Integration zwischen klassischem Requirements Engineering und agilen Methoden, unter Berücksichtigung von „Agile“ bei Allianz, unternommen. In Kapitel 4.2 werden die bei Allianz vorherrschenden Philosophien zu Usability von Geis mit denen von Lauesen integriert. Schließlich wird aus dieser Vorarbeit in Abschnitt 4.3 ein Entwurf eines SOLL-RE-Prozesses vorgestellt.

4.1 Integration von RE und agilen Methoden

In der Analyse des IST-RE-Prozesses wurden Defizite in der Dokumentation und der Analyse von Anforderungen, wie auch der Kundeneinbindung festgestellt. Um diese Defizite zu beseitigen können die in Kapitel 2.2.4 formulierten Praktiken angewandt werden. Tabelle 6 zeigt auf, welche der Praktiken bei Allianz bereits in einer ähnlichen Form praktiziert werden und welche überhaupt anwendbar sind. Diese Erkenntnisse werden im Folgenden diskutiert.

Bei Betrachtung des jetzigen RE-Prozesses wird deutlich, dass bislang keine systematische Stakeholder-Analyse (REAM 1) durchgeführt wird. Dadurch, dass die Stakeholder für jede Erweiterung des ABS allerdings sehr ähnlich, wenn nicht identisch sind (beispielsweise sind die Benutzerinnen und Benutzer immer Sachbearbeiterinnen und Sachbearbeiter) muss eine Stakeholder-Analyse nur einmalig durchgeführt werden. Ich gehe davon aus, dass dies bei der Erstellung des ABS getan wurde. Daher muss diese Praktik nicht in den RE-Prozess aufgenommen werden. Ich empfehle aber die Stakeholder-Analyse einmalig zu wiederholen oder regelmäßig, z.B. jedes Jahr, unabhängig von Teilprojekten durchzuführen.

Die Projektleiterin bzw. der Projektleiter (oder Product Owner im agilen Sinne) bei Allianz berücksichtigt die Bedürfnisse der späteren Benutzerinnen und Benutzer nicht in ausreichendem Maß (REAM 2). Um dies zu ändern, muss der Prozess jedoch nicht angepasst werden. Bei den Entscheidungen des Product Owners ist es lediglich notwendig, dass er die Belange der Benutzerinnen und Benutzer berücksichtigt. Dazu kann es sinnvoll sein Vertreter der Benutzerinnen und Benutzer zu bestimmen und explizit mit einzubinden. Meiner Meinung nach ist es wenig gewinnbringend den Prozess stark abzuändern, indem man die Rolle des Product Owners durch ein „steering comitee“ oder Ähnliches erweitert und dadurch mehr Komplexität und Aufwand generiert.

<i>Praktik</i>	<i>Bereits praktiziert?</i>	<i>Anwendbar?</i>
REAM 1	Einmalig	Nein
REAM 2	Nein	Teilweise
REAM 3	Ja	-
REAM 4	Nein	Nein
REAM 5	Ja	-
REAM 6	Implizit	Ja
REAM 7	Nein	Nein
REAM 8	Ja	-
REAM 9	Nein	Ja

Tabelle 6: Anwendbarkeit der REAM-Praktiken bei Allianz

Nicht-funktionale Anforderungen werden bei Allianz bislang, genau wie funktionale Anforderungen, systematisch erhoben und (in AMT) dokumentiert. Entsprechend ist die Praktik REAM 3 hinfällig. Auch die Dokumentation durch System Stories (REAM 4) scheint nicht notwendig, da Allianz durch die Verwendung von AMT bereits ein Anforderungsdokument pflegt, welches über die Vorgaben von Scrum und XP hinaus geht. Die Organisation nicht-funktionaler Anforderungen mittels System Stories ist dadurch insofern überflüssig, als dass die nicht-funktionalen Anforderungen in AMT ausreichend dokumentiert sind und bei User Stories referenziert werden können. Entsprechend der Dokumentation in AMT entfällt auch die Anwendung von REAM 5, der Pflege eines Anforderungsdokuments über das Rahmenmodell von Scrum und XP hinaus.

Die Verifikation von Anforderungen, speziell die Prüfung auf Konsistenz von Anforderungen (REAM 6), wird bei Allianz bislang nicht explizit durchgeführt und ist auch eine der entdeckten Schwachstellen des jetzigen RE-Prozesses. Diese Prüfung sollte im Unterprozess „Anforderungen überprüfen“ durch den Product Owner, der für das Product Backlog verantwortlich ist, durchgeführt werden. Dabei sollten zudem unterstützend Werkzeuge eingesetzt werden, welche eine automatische Verifikation, beispielsweise in Form einer Syntaxprüfung von Anforderungen zur Prävention von Inkonsistenzen unterstützen (REAM 7). Dies ist bei Allianz allerdings nur insoweit möglich, wie dies die bestehende Landschaft an Werkzeugen unterstützt. Die Einführung neuer Werkzeuge, allein zu diesem Zweck, würde zu hohe Kosten verursachen.

Die Anforderungen werden bei Allianz bereits sehr früh ausgearbeitet. Somit ist auch die Granularität bereits sehr früh ausreichend (REAM 8), um so später nicht von wichtigen Details überrascht zu werden. Auch sind die Teams meist mit bereits erfahrenen Entwicklerinnen und Entwicklern besetzt, so dass diese frühzeitig kritische Punkte in Anforderungen entdecken. Hierbei besteht also kein Handlungsbedarf. Bei der Modellierung und Darstellung von Abhängigkeiten zwischen Anforderungen (REAM 9) jedoch besteht Handlungsbedarf, denn Abhängigkeiten werden bislang nur rein in Form von Daten in AMT erfasst, aber nicht geeignet visualisiert, um eine zufriedenstellende Verfolgbarkeit zu garantieren.

Zusätzlich zu den Betrachtungen der REAM-Praktiken, muss der Aufbau des RE-Prozesses zu einem agilen Prozess umgestellt werden. Er muss eine iterative Durchführung in Form von Sprints explizit unterstützen. Dazu schlage ich für die Vorphase vor den Unterprozess „IST-Prozesse analysieren und Anforderungen an diese erheben“ im Rahmen von Sprints mehrmals durchführen zu können. Bei der Anforderungsanalyse schlage ich vor die Unterprozesse „Basisanforderungen ausarbeiten“ (bzw. im Kontext des SOLL-RE-Prozesses „Basisanforderungen ausarbeiten und Nutzungsanforderungen erfassen“ benannt) und „Anforderungen implementieren“ ebenfalls iterativ durchführbar zu machen. Diese beiden Unterprozesse und die fachliche Prüfung der Einzelfunktionalität sollten parallel ausgeführt werden, um den RE-Prozess so möglichst effizient zu gestalten. So dienen die Anforderungen, die in einer Iteration des Unterprozesses „Basisanforderungen ausarbeiten“ ausgearbeitet wurden, als Eingabe für die Implementation von Anforderungen, währenddessen können aber auch schon weitere Anforderungen ausgearbeitet werden. Ähnliches gilt für den fachlichen Test der Einzelfunktion.

Zusammenfassung

Die Kernpunkte der Integration von Requirements Engineering und agilen Methoden im Sinne der Praktiken REAM können wie folgt zusammengefasst werden:

- Berücksichtigung der Interessen der Benutzerinnen und Benutzer durch den Product Owner (bestenfalls durch Bestimmung von Vertreterinnen und Vertretern, die den Product Owner beraten)
- Prüfung auf Konsistenz der Anforderungen durch den Product Owner in Unterprozess „Anforderungen überprüfen“ aufnehmen
- Einsatz geeigneter Werkzeuge zur Darstellung von Abhängigkeiten zwischen Anforderungen
- Einfügen einer Iterationsmöglichkeit für den Unterprozess „IST-Prozesse analysieren und Anforderungen an diese Erheben“
- Einfügen einer Iterationsmöglichkeit für den Unterprozess „Basisanforderungen ausarbeiten“ (bzw. „Basisanforderungen ausarbeiten und Nutzungsanforderungen erfassen“)
- Einfügen einer Iterationsmöglichkeit für den Unterprozess „Anforderungen implementieren“
- Parallelisierung des Unterprozesses „Basisanforderungen ausarbeiten“, der Implementation von Anforderungen und der fachlichen Prüfung der Einzelfunktionalität

Durch diese Vorschläge kann der Prozess zunächst stärker auf agile Ausführung angepasst werden. Darüber hinaus werden auch die erkannten Mängel in der Dokumentation und der Analyse von Anforderungen behoben.

4.2 Integration der Methoden von Lauesen und Geis

Bei der Analyse des IST-RE-Prozesses wurde festgestellt, dass die Einbindung und die Zufriedenheit der Benutzerinnen und Benutzer stark verbesserungswürdig sind. Um dies zu optimieren, können die in Kapitel 2.3.4 formulierten Praktiken angewandt werden. Welche dieser Praktiken bei Allianz noch nicht praktiziert werden, d.h. in einer ähnlichen Form, und welche davon überhaupt anwendbar sind, zeigt Tabelle 7 und wird im Anschluss diskutiert.

Nicht explizit durch die Messung, aber durch Studium des RE-Prozesses, wird klar, dass Allianz die Nutzerrollen (GEBR 1) im Rahmen des ABS nicht systematisch erfasst. Dadurch, dass die Benutzerinnen und Benutzer allerdings bekannt sind und bei jedem Projekt ein ähnliches Profil haben (immer Sachbearbeiterinnen oder Sachbearbeiter mit Expertenwissen in einem bestimmten Fachgebiet), kann diese Praktik jedoch ausgelassen werden. Ich gehe davon aus, dass die Nutzerrollen bei der Erstellung des ABS betrachtet wurden. Dennoch empfehle ich die Nutzerrollen einmalig erneut zu erfassen oder regelmäßig zu wiederholen, beispielsweise einmal pro Jahr. Es ist also nicht notwendig eine Tätigkeit zur systematischen Ermittlung der Nutzerrollen in den RE-Prozess aufzunehmen.

<i>Praktik</i>	<i>Bereits praktiziert?</i>	<i>Anwendbar?</i>
GEBR 1	Einmalig	Nein
GEBR 2	Ja	-
GEBR 3	Nein	Ja
GEBR 4	Ja	-
GEBR 5	Nein	Ja
GEBR 6	Nein	Ja
GEBR 7	Nein	Ja
GEBR 8	Nein	Ja
GEBR 9	Ja	-

Tabelle 7: Anwendbarkeit der GEBR-Praktiken bei Allianz

Die Ermittlung der Aufgaben und deren Unterstützung der Erweiterung des ABS (GEBR 2) wird bisher im Rahmen der Erhebung der Anforderungen mit dem Ergebnis der Basisanforderungen durchgeführt. Die Berücksichtigung von Nutzungsanforderungen (GEBR 3) fehlt allerdings bislang im RE-Prozess. Zwar werden abgeleitete Anforderungen formuliert, die notwendige Tätigkeiten zur Umsetzung der Basisanforderungen beschreiben, dort fehlt allerdings die Berücksichtigung der Erfordernisse der Benutzerinnen und Benutzer. Es bietet sich an, die von Geis vorgeschlagene Aktivität der Spezifikation von Nutzungsanforderungen in den RE-Prozess mit aufzunehmen, da Lauesen hierzu keine expliziten Praktiken aufführt. Um diese Anforderungen zu erfassen, ist es notwendig Interviews mit typischen Benutzerinnen und Benutzern durchzuführen. Diese Interviews sollten bereits möglichst früh im Projekt durchgeführt werden, allerdings erst nachdem der Antrag genehmigt wurde, da hierdurch keine neuen Basisanforderungen entstehen werden und darüber hinaus damit auch ein nicht unerheblicher Aufwand verbunden ist, der nicht ohne direkten Nutzen aufgebracht werden sollte. Die beiden Aktivitäten, also die Durchführung der Interviews und die darauf basierende Spezifikation von Nutzungsanforderungen, schlage ich daher vor parallel zur Ausarbeitung der Basisanforderungen durch Usability-Expertinnen und Usability-Experten (BO und IT) durchzuführen.

Die Ermittlung eines Datenmodells (GEBR 4) wird durch Ausarbeitung der Basisanforderungen zu abgeleiteten Anforderungen erledigt. Durch die zuvor angesprochenen Interviews und Spezifikation der Nutzungsanforderungen ist zu erwarten, dass noch weitere Ergänzungen zu dem Datenmodell werden. Es sind, meiner Meinung nach, darüber hinaus keine weiteren Tätigkeiten notwendig.

Wie genau Aufgaben und Daten auf Arbeitsbereiche und Sichten verteilt werden, liegt bislang im Ermessen der Entwicklerinnen und Entwickler im Work Team und wird nicht systematisch geplant, wie es GEBR 5 vorsähe. Entsprechend dieser Beobachtung ist es ratsam eine solche Tätigkeit in den RE-Prozess aufzunehmen, welche von dem Work Team durchgeführt werden sollte. Lauesen nennt in diesem Zuge auch den Understandability-Test (GEBR 6), welcher unverzichtbar zur Prüfung des Entwurfs der Virtual Windows ist. Auch hier findet sich kein Äquivalent im derzeitigen RE-Prozess. Mein Vorschlag ist daher, vor der Aktivität zum Entwurf der Lösung zunächst die Virtual Windows zu erstellen und diese im Anschluss durch einen Understandability-Test mit späteren Benutzerinnen und Benutzern zu prüfen. Der Understandability-Test sollte vom Work Team, unterstützt durch die BO-Usability-Verantwortliche bzw. den BO-Usability-Verantwortlichen, durchgeführt werden. Gegebenenfalls müssen die beiden Schritte der Erstellung der Virtual Windows und des Understandability-Tests solange wiederholt werden, bis der Entwurf zufriedenstellend ist.

Der eigentliche Entwurf der Sichten sollte basierend auf den Virtual Windows ausgearbeitet werden und mittels Paper-Mock-Ups iterativ entwickelt und durch Usability-Tests geprüft werden (GEBR 7 und 8). Auch hier gibt es keine entsprechenden Tätigkeiten, die diesem Vorgehen im jetzigen RE-Prozess entsprechen würden. Die Tätigkeiten sollten direkt im Anschluss zu dem erfolgreichen Understandability-Test durch das Work Team, in Zusammenarbeit mit einer oder einem Usability-Verantwortlichen, durchgeführt werden.

Die späteren Benutzerinnen und Benutzer können am Ende des Entwicklungsprozesses mittels eines Usability-Tests eingebunden werden. Dies entspricht der Praktik GEBR 9, der Durchführung eines abschließenden Usability-Tests. Es sollte lediglich sichergestellt werden, dass die Benutzerinnen und Benutzer *immer* hinzugezogen werden, insofern das Vorhaben Usability-relevant ist und dass nicht nur z.B. auf Styleguide-Konformität geprüft wird.

Zusammenfassung

Die Kernpunkte der Optimierung des RE-Prozesses in Bezug auf Gebrauchstauglichkeit nach den Vorgaben von Lauesen und Geis in Form der Praktiken GEBR umfassen zusammenfassend folgende Punkte. Diese greifen nur, insofern das Vorhaben in der Vorphase als Usability-relevant eingestuft wurde.

- Erhebung und Spezifikation von Nutzungsanforderungen durch Interviews nach Geis, durchgeführt von Usability-Verantwortlichen der BO und IT (parallel zur Ausarbeitung der Anforderungen).
- Erstellung von Virtual Windows nach Lauesen vor Entwurf der Lösung durch Work Team.
- Direkt im Anschluss Understandability-Test nach Lauesen durch Work Team, unterstützt durch eine Usability-Verantwortliche oder einen Usability-Verantwortlichen. Sollten Probleme festgestellt werden, dann Virtual Windows überarbeiten und Understandability-Test erneut durchführen.
- Erstellung von Paper-Mock-Ups nach Lauesen durch Work Team nach erfolgreichem Understandability-Test.
- Usability-Test nach Lauesen im Anschluss durch Work Team, unterstützt durch eine Usability-Verantwortliche oder einen Usability-Verantwortlichen. Sollten Probleme festgestellt werden, dann Paper-Mock-Ups überarbeiten und Usability-Test erneut durchführen.
- Hinzuziehen der Benutzerinnen und Benutzer in der abschließenden Prüfung auf Usability und Ergonomie.

Durch diese Vorschläge werden die Benutzerinnen und Benutzer viel stärker (und auch systematischer) eingebunden und der Entwurf der Anforderungen wird zu einem frühen Zeitpunkt, zu dem die Änderbarkeit noch problemlos gegeben ist, durch die Benutzerinnen und Benutzer geprüft. Diese Änderungsvorschläge entsprechen zudem der in Kapitel 2.3 thematisierten Norm DIN EN ISO 9241.

Es ist möglich, je nach Umfang des Teilprojekts, das Ausmaß der von mir vorgeschlagenen Aktivitäten anzupassen. So muss beispielsweise nicht bei jeder Iteration ein Understandability-Test durchgeführt werden, wenngleich es ratsam ist dies zu tun, um die Gebrauchstauglichkeit des Softwareprodukts zu maximieren.

4.3 RE-Prozess

Bevor in Kapitel 4.3.4 der eigentliche Prozess erläutert wird, werden in den Abschnitten 4.3.1 bis 4.3.3 zunächst Rollen, Werkzeuge und Dokumente beschrieben, die sich im Vergleich zum IST-RE-Prozess verändert haben oder neu hinzugefügt worden sind. Die restlichen Erläuterungen sind in den Kapiteln 3.1.2 bis 3.1.4 zu finden.

4.3.1 Rollen

Alle Rollen, die im IST-RE-Prozess vorhanden waren, bleiben im SOLL-RE-Prozess bestehen. Die Kompetenzen der Rollen Anwender, BO PO / BO PL, BO bzw. IT-Usability-Verantwortlicher und Work Team erweitert.

4.3.1.1 Anwender

Die Rolle des Anwenders wird im SOLL-RE-Prozess deutlich stärker eingebunden. Bislang war der Anwender, wenn überhaupt, nur bei der abschließenden Usability-

und Ergonomieprüfung eingebunden. Nun wird der Anwender bereits in der Phase der Anforderungsanalyse eingebunden, da er bezüglich seines Nutzungskontexts interviewt wird. Auch in der späteren Entwurfs- und Implementationsphase wird der Anwender im Rahmen von Understandability- und Usability-Tests eingebunden.

4.3.1.2 BO PO / BO PL

Die wesentliche Erweiterung der Rolle des BO PO bzw. BO PL besteht darin, dass die Rolle die Belange der späteren Benutzerinnen und Benutzer berücksichtigen soll. Dies betrifft vor allem Fragen der Priorisierung von Anforderungen im Rahmen der Sprintplanung zur Implementierung.

Des Weiteren ist der Rolle nun im Kontext der Überprüfung von Anforderungen eine explizite Aktivität zur Prüfung der Anforderungen auf Konsistenz zugewiesen, was im IST-RE-Prozess bislang nur implizit enthalten war.

4.3.1.3 BO- bzw. IT-Usability-Verantwortlicher

Die Verantwortlichen für Usability aus BO und IT erhalten, gegenüber dem IST-RE-Prozess, die Verantwortung für neue Aktivitäten zur Erhebung von Nutzungsanforderungen. Sie sollen bei der Durchführung neuer Understandability- und Usability-Tests unterstützen.

4.3.1.4 Work Team

Das Work Team ist im SOLL-RE-Prozess für neue Aktivitäten zur Erstellung von Virtual Windows und Paper-Mock-Ups, sowie zur Durchführung von Understandability- und Usability-Tests verantwortlich.

4.3.2 Werkzeuge

Im Gegensatz zum IST-RE-Prozess fallen die bisherigen Systeme AMT und HP Quality Center weg. Sie werden durch HP Demand bzw. Tosca ersetzt. Dies beruht allerdings nicht auf Erkenntnissen dieser Arbeit, sondern wurde von Allianz selbst so geplant.

4.3.2.1 HP Demand

HP Demand (im Folgenden nur noch „Demand“ genannt) ersetzt AMT als Werkzeug zum Anforderungsmanagement. Dazu werden die Funktionalitäten von AMT in Demand nachgestellt, aber auch erweitert. Beispielsweise ist es in Demand auch möglich, Abhängigkeiten zwischen Anforderungen visuell darzustellen.

4.3.2.2 Tosca

Tosca ist ein von der TRICENTIS Technology & Consulting GmbH entwickeltes und vertriebenes Werkzeug für Testmanagement und Testautomation. Es ersetzt HP Quality Center als Werkzeug zum Testmanagement. Diese Änderung gegenüber dem IST-RE-Prozess begründet sich auf eine Angleichung an internationale Vorgaben der Allianz Group.

Über die von Allianz benutzte Funktionalität von HP Quality Center (Testmanagement) hinaus, soll bei Tosca auch die Testautomation genutzt werden [49]. Tosca bietet dazu die Möglichkeit Systemtestfälle automatisiert auszuführen.

4.3.3 Dokumente

Die verwendeten Dokumente aus dem IST-RE-Prozess werden alle beibehalten, aber um solche zur Unterstützung von Gebrauchstauglichkeit verstärkt.

4.3.3.1 Kontextszenario

Ein Kontextszenario (nach Geis) beschreibt den Nutzungskontext von Benutzerinnen und Benutzern zu einem Softwaresystem. Dazu wird zunächst in einer Einleitung festgehalten, wessen Nutzungskontext betrachtet wird und in welchem Zusammenhang diese Person mit dem Softwaresystem steht. Weiterhin wird beschrieben, welche typischen Aufgaben die Benutzerinnen und Benutzer wie erledigen, welche Sonderfälle sie dabei beachten müssen und auf welche Probleme sie stoßen. Auch besondere Wünsche der Benutzerinnen und Benutzer werden festgehalten.

Kontextszenarien werden mittels Interviews mit Benutzerinnen und Benutzern erhoben. Sie stellen die Grundlage zur späteren Erfassung von Nutzungsanforderungen dar.

4.3.3.2 Nutzungsanforderung

Durch die Auswertung von Kontextszenarien erstehen Nutzungsanforderungen. Diese sind Benutzeraktionen, die am System ausführbar sein müssen, um ein Erfordernis im Nutzungskontext effizient zu erfüllen [29].

4.3.3.3 Paper-Mock-Up

Ein Paper-Mock-Up ist ein Papier-basierter Prototyp der Benutzungsschnittstelle. Solche Prototypen werden im RE-Prozess aus den Virtual Windows heraus erstellt. Sie können mit Stift und Papier, aber auch mit Werkzeugen erstellt werden. Sie sollen die späteren Masken des Softwaresystems widerspiegeln, um bereits früh im Entwicklungsprozess Usability-Tests durchführen zu können. Sie dienen als Grundlage für die spätere Implementation der Benutzungsschnittstelle.

4.3.3.4 Virtual Window

Ein Virtual Window ist ein Bild auf einem idealisierten Bildschirm, eine Benutzerorientierte Darstellung von persistenten Daten. Virtual Windows sind auf Papier. Sie zeigen Daten, haben aber keine Buttons, Menüs oder andere Funktionen [25; 28].

Im RE-Prozess bilden sie basierend auf den Nutzungsanforderungen die Grundlage für die Implementation eines gebrauchstauglichen Softwaresystems. Dazu werden die Virtual Windows im Vorfeld der Implementation mit den späteren Benutzerinnen und Benutzern abgestimmt, um Fehler möglichst früh zu entdecken und zu beheben.

4.3.4 Erläuterungen des Prozesses

Im Folgenden wird nun der im Anhang unter Kapitel F.2.5 modellierte SOLL-RE-Prozess vorgestellt. Zum besseren Verständnis orientiert sich die Erläuterung, ähnlich dem IST-RE-Prozess, an den RE-Phasen aus Kapitel 2.2.1.

Die Aktivitäten des SOLL-RE-Prozesses sind zwei Rollen zugeordnet. Zum einen die des anfordernden Teilprojekts und zum anderen die des umsetzenden Teilprojekts.

Erhebung von Anforderungen

Der Prozess beginnt mit dem Beschluss des Managements einen neuen Bereich in ABS aufzunehmen, also die vorhandene Logik des ABS Kerns für deutsche Verhältnisse zu erweitern. Der Product Owner der BO ist im Rahmen der ersten Tätigkeit – siehe dazu der Unterprozess „IST-Prozesse analysieren und Anforderungen an diese erheben“ – dafür verantwortlich, zunächst die IST-Prozesse zu analysieren und dann darauf basierend eine Delta-Analyse durchzuführen, um notwendige Erweiterungen des ABS Kerns in Form von Basisanforderungen zu identifizieren.

Die Aktivitäten wurden unverändert aus dem IST-RE-Prozess übernommen. Die Ausführung selbiger ist allerdings nicht mehr einmalig, sondern kann nun iterativ vorgenommen werden, um der Agilität gerecht zu werden.

Dokumentation von Anforderungen

In einer der oben angesprochenen Iterationen werden, insofern durch die Delta-Analyse Notwendigkeiten zur Änderung erkannt wurden, die Basisanforderungen, die diese Notwendigkeiten beschreiben, in Demand festgehalten. Nun werden in einem nächsten Sprint erneut die IST-Prozesse analysiert und eine Delta-Analyse durchgeführt und wenn nötig auch Basisanforderungen erfasst. Die Iteration um diesen Unterprozess endet damit, dass alle notwendigen Änderungen erkannt wurden und somit keine weiteren Basisanforderungen mehr erfasst werden müssen.

Im Hauptprozess folgt nun zunächst die Prüfung, ob im Unterprozess zuvor überhaupt Basisanforderungen erfasst wurden, also Änderungen notwendig sind. Ist dem nicht so, so ist der gesamte Prozess zu diesem Zeitpunkt beendet. Andernfalls folgt die Stellung des Projektantrags, wobei aus Requirements Engineering-Sicht die dabei enthaltene Prüfung des Vorhabens auf Usability-Relevanz besonders hervorzuheben ist. Hierbei kreuzt die Projektmanagerin bzw. der Projektmanager an, ob das geplante Vorhaben Usability-relevant ist oder nicht, also ob Änderungen oder Ergänzungen an der GUI vorgenommen werden müssen. Mit dem genehmigten Projektantrag endet die Vorphase des ABS Vorgehensmodells. Der nächste Schritt ist die Priorisierung der Basisanforderungen durch die Programmleitung, im Gegensatz zu den bisherigen Aktivitäten, im umsetzenden Teilprojekt.

Prüfung und Abstimmung von Anforderungen

Nach der Priorisierung werden parallel zwei Wege beschritten. Zum einen werden für alle zurückgewiesenen Anforderungen (das sind solche, die mit dem Typ „3-Ablaufänderung“ ausgezeichnet wurden) durch den BO Product Owner (je nach höherem Fachwissen aus dem anfordernden oder umsetzenden Teilprojekt) ein Workaround definiert und in Demand festgehalten. Für alle akzeptierten Basisanforderungen werden, wenn das Vorhaben bei Stellen des Projektantrags als Usability-relevant eingestuft wurde, – und das ist der nächste Unterschied zum IST-RE-Prozess – Interviews mit Benutzerinnen und Benutzern durchgeführt, um Kontextszenarien zu erheben (GEBR 3). Die Interviews sollen nach Geis (siehe S. 37 ff. in [29]) von zwei Personen durchgeführt werden, wobei eine der beiden Personen die Gesprächsführung übernehmen und die andere die Erfassung aller relevanten Daten sicherstellen soll. Diese Personen sollen die beiden BO- und IT-Usability-Verantwortlichen sein, da sich diese am besten mit dem Thema Gebrauchstauglichkeit auskennen. Interviewt werden sollen insgesamt drei bis fünf Benutzerinnen und Benutzer aus einer Nutzergruppe, um möglichst alle verschiedenen Typen von Benutzerinnen und Benutzern zu befragen. Ein Interview dauert insgesamt ca. zwei Stunden und soll mit einer Per-

son durchgeführt werden (also keine Gruppeninterviews). Die Fragen im Interview sollen sich an den Leitfragen aus dem DATech Leitfaden Usability [50] ausrichten und darauf abzielen die Tätigkeit der Mitarbeiterin oder des Mitarbeiters mit all ihren Facetten, wie es sich täglich darstellt, zu verstehen. Direkt im Anschluss an das Interview soll die bzw. der BO-Usability-Verantwortliche dann das Kontextszenario festhalten und im Anschluss zusammen mit der bzw. dem IT-Usability-Verantwortlichen validieren, so dass die Ansichten beider Usability-Verantwortlichen übereinstimmend beschrieben sind.

In der Folge werden im Modell erneut parallele Pfade beschriftet. Zunächst soll dabei allerdings erst der linke Pfad zu dem Unterprozess „Basisanforderungen ausarbeiten und Nutzungsanforderungen erfassen“ verfolgt werden, da die Ergebnisse der dortigen ersten Iteration den Input für den rechten Pfad, der Implementation innerhalb der ADAG, darstellen; allgemeingültige Basisanforderungen können bereits an die Verantwortlichen aus Österreich weitergegeben werden. Fertig entworfene, implementierte und getestete Lösungen zu den abgeleiteten Anforderungen und Nutzungsanforderungen können parallel im mittleren Pfad, dem fachlichen Test der Einzelfunktion durch ein Mitglied des Work Teams geprüft werden. Diese Prüfung folgte bislang im IST-RE-Prozess im Anschluss an die fertiggestellte Implementation *aller* abgeleiteten Anforderungen.

Im Unterprozess „Basisanforderungen ausarbeiten und Nutzungsanforderungen erfassen“, im IST-RE-Prozess sequentiell vor der Implementation, werden parallel durch den BO Product Owner die Basisanforderungen zu abgeleiteten Anforderungen ausgearbeitet und, falls das Vorhaben Usability-relevant ist, die Kontextszenarien ausgewertet und die daraus entstehenden Nutzungsanforderungen in Demand erfasst (GEBR 3). Erstgenannte Aktivität ist so aus dem IST-RE-Prozess bereits bekannt, die letztgenannte wurde hinzugefügt. Um Nutzungsanforderungen aus den Kontextszenarien zu extrahieren, ist es zunächst notwendig direkte Erfordernisse der Benutzerinnen und Benutzer zu identifizieren. Ein Erfordernis ist eine notwendige Voraussetzung, die es ermöglicht, den in einem Nutzungskontext enthaltenen Zweck effizient zu erfüllen. Leitfragen nach [29] S. 49 zur Extraktion von Erfordernissen aus Kontextszenarien sind z.B. „Was soll in dieser Situation erreicht werden?“, „Warum wird das in dieser Situation so gemacht?“, „Was ist notwendig, damit man in dieser Situation zum Ergebnis kommt?“ und „Wodurch kann diese Situation verbessert werden?“. Aus Erfordernissen können dann direkt die Nutzungsanforderungen abgeleitet werden. Sie sind Interpretationen der Erfordernisse, die beschreiben, wie selbige erfüllt werden können, ohne dabei eine konkrete Lösung vorzugeben.

Im Anschluss werden die abgeleiteten Anforderungen und Nutzungsanforderungen im Unterprozess „Anforderungen überprüfen“, wie auch bereits im IST-RE-Prozess, geprüft. Bei dieser Prüfung wurde lediglich noch eine weitere Tätigkeit unter der Verantwortung der BO Product Owners – diese Person hat den besten Überblick über die Anforderungen – hinzugefügt, die die Konsistenzprüfung der Anforderungen vorsieht (REAM 6). Hierbei soll der Product Owner sich die bisherigen abgeleiteten Anforderungen und Nutzungsanforderungen ansehen und prüfen, ob diese konsistent sind, also insbesondere keine Widersprüche oder logische Duplikate enthalten.

Sollten Mängel festgestellt werden, so müssen die entsprechenden Anforderungen überarbeitet werden. Auch zu dem Unterprozess „Basisanforderungen ausarbeiten und Nutzungsanforderungen erfassen“ besteht wieder die Möglichkeit zur Iteration im agilen Sinne, d.h. Sprint für Sprint die Basisanforderungen ausarbeiten und die Nutzungsanforderungen erfassen und schließlich alle überprüfen. Anforderungen sollen nicht mehr alle zur selben Zeit ausgearbeitet werden, sondern nur noch stückweise. Dabei ist ein enges Zusammenspiel mit der parallel laufenden Implementation wich-

tig. Die Ausarbeitung der Anforderungen sollte nur soweit vollzogen werden, dass das Verständnis der Anforderung für das Work Team zu Beginn eines Sprints zur Implementation gegeben ist, nicht jedoch in einem überflüssigen Ausmaß. Hier das richtige Niveau zu finden, ist vor allem eine Frage der Abstimmung der einzelnen Teams im Teilprojekt und kann schwer vorgegeben werden.

Weiteres Vorgehen

Bevor noch kurz auf die übergreifende RE-Aktivität „Verwaltung von Anforderungen“ eingegangen wird, wird zunächst noch der modellierte Prozess bis zum Ende erläutert.

Im rechten Pfad der oben angesprochenen parallelen Ausführung steht der Unterprozess „Anforderungen implementieren“. Dieser benötigt als Input die aus dem Unterprozess „Basisanforderungen ausarbeiten und Nutzungsanforderungen erfassen“ Stück für Stück festgehaltenen, abgeleiteten Anforderungen und Nutzungsanforderungen. Am Anfang des Unterprozesses steht der bereits bekannte Schritt des Akzeptierens der Anforderungsspezifikation, der sicherstellt, dass das Work Team, welches die Anforderungen im Rahmen des umsetzenden Projekts nun entwerfen, implementieren und testen soll, die Anforderungen auch verstanden hat. Im Folgenden sollen, falls das Vorgehen im Projektantrag als Usability-relevant eingestuft wurde, weitere, im bisherigen IST-RE-Prozess bislang nicht vorhandene Aktivitäten ausgeführt werden. Zunächst sollen Virtual Windows nach Lauesen (siehe auch Kapitel 2.3.1 – „Virtual windows design“) durch das Work Team erstellt werden (GEBR 5). Dazu werden vor allem die Nutzungsanforderungen, aber auch abgeleitete Anforderungen benutzt, um die Aufgaben, die die Benutzerinnen und Benutzer mit dem System durchführen müssen, auf Virtual Windows aufzuteilen. Es müssen die mit einer Aufgabe verbundenen Daten auf Virtual Windows verteilt werden, unter der Maßgabe möglichst wenige Virtual Windows zu kreieren, die gleichartige Daten gruppieren. Während der Erstellung ist es vor allem auch wichtig, vorhandene Virtual Windows (die bereits während diesem oder des vorhergehenden Sprints erstellt wurden) wieder zu benutzen oder zu erweitern, statt neue zu entwerfen. Um zu prüfen, ob die Darstellung der Daten ausreicht, um alle notwendigen Manipulationen (nach Bedarf: erstellen, lesen, bearbeiten, löschen) durchführen zu können, kann der CREDO¹²-Check angewandt werden. Funktionen sollen aber noch in keinsten Weise auf diesen Papier-basierten sehr frühen Prototypen angedeutet werden. Dafür können Beispieldaten schon einmal exemplarisch eingefügt werden, damit das Verständnis für andere erleichtert wird. Letzteres wird in der nachfolgenden Aktivität „Understandability-Test durchführen“ geprüft (GEBR 6). Hierbei soll eine Vertreterin oder ein Vertreter des Work Teams einer Benutzerin oder einem Benutzer während des etwa 30 bis 60 Minuten dauernden Tests nacheinander die Virtual Windows vorlegen und schließlich erläutern lassen, was nach der Meinung der Benutzerin bzw. des Benutzers auf der jeweiligen Maske zu sehen ist und was noch fehlt etc. Ein weiteres Mitglied des Work Teams soll während des Tests Protokoll führen. Dieser Test sollte nicht nur mit einer Person aus dem Kreis der Benutzerinnen und Benutzer durchgeführt werden, sondern, ähnlich wie bei den Interviews zur Bestimmung von Kontextszenarien, mit mindestens einer Person aus jeder Nutzergruppe. Dazu gehört vor allem auch eine Experten-Benutzerin oder einen Experten-Benutzer mit einzubeziehen, was insbesondere hier wichtig ist. Die BO-Usability-Verantwortliche oder der BO-Usability-

¹² CREDO = Create, Read, Edit, Delete, Overview [25]

Verantwortliche soll das Work Team bei der Durchführung des Understandability-Tests mit Know-How unterstützen, dies muss bei erfahrenen Work Teams allerdings nicht mehr zwingend so sein. Sollten in dem Test wesentliche Probleme festgestellt werden, fehlen z.B. wichtige Daten auf den Masken, so müssen die Virtual Windows überarbeitet und der Understandability-Test bei Bedarf wiederholt werden.

Konnten keine schwerwiegenden Probleme mit den Virtual Windows festgestellt werden oder wurden alle derartigen Probleme behoben, so werden die Virtual Windows im nächsten Prozessschritt weiter zu fertigen Paper-Mock-Ups ausgearbeitet. Dazu werden die zur Erstellung der Virtual Windows bereits herangezogenen Nutzungsanforderungen benutzt, um Funktionen zu extrahieren, die für die Ausführung der Aufgaben notwendig sind. Diese Funktionen werden dann auf die Virtual Windows aufgeteilt. Dabei müssen auch solche Funktionen berücksichtigt werden, die zur Navigation zwischen den Masken notwendig werden.

Die fertigen Paper-Mock-Ups werden im Anschluss durch einen Usability-Test geprüft (GEBR 7). Dazu werden, ähnlich wie beim Understandability-Test, jeweils einer Benutzerin oder einem Benutzer einer Nutzergruppe durch ein Mitglied des Work Teams die Masken vorgelegt, diesmal allerdings nicht nacheinander, sondern in der Reihenfolge, in denen sie bei der späteren GUI erscheinen würden. Die Testperson erhält eine Aufgabe, die sie mit Hilfe des Papier-Systems erledigen muss und verfährt dabei, wie bei einem echten Softwaresystem. Die Interaktion wird durch das Mitglied des Work Teams hergestellt, der die Masken entsprechend der Eingaben der Testperson auswechselt. Probleme bei der Handhabung der Ausführung der Aufgabe (z.B. Missverständnisse o.ä.) oder der Masken sollen durch ein weiteres Mitglied des Work Teams protokolliert werden. Es kann der Testperson dabei auch erlaubt sein, die Gedanken laut zu äußern, um so für das Work Team leichter erkennlich zu machen, wo Probleme auftauchen. Der Test dauert für gewöhnlich ca. eine Stunde und auch hier sollte die BO-Usability-Verantwortliche oder der BO-Usability-Verantwortliche unterstützend tätig sein.

Wurden bei dem Test kritische Probleme festgestellt, so müssen die Paper-Mock-Ups angepasst und gegebenenfalls erneut ein Usability-Test durchgeführt werden (GEBR 8). Wurden keine derartigen Probleme festgestellt, so werden die späteren ABS-Masken – dies ist wieder aus dem IST-RE-Prozess bekannt – basierend auf den Paper-Mock-Ups entworfen, nochmals gemäß der Richtlinien zur Usability und Ergonomie geprüft, implementiert und getestet. Bei den Aktivitäten zum Testen kommt, im Gegensatz zum IST-RE-Prozess, wegen einer Angleichung an internationale Vorgaben der Allianz Group, nicht mehr HP Quality Center, sondern Tosca zum Einsatz. Tosca unterstützt allerdings auch die Funktionalität, die Allianz bisher von HP Quality Center benötigte. Am Ende der genannten Kette von Aktivitäten und nach Beendigung des Unterprozesses besteht die Möglichkeit in eine weitere Iteration zu gehen und den Unterprozess erneut zu beginnen, um inzwischen neue, durch den parallel laufenden Unterprozess der Ausarbeitung von Basisanforderungen und Erfassung von Nutzeranforderungen dokumentierten, Anforderungen zu bearbeiten.

Wenn alle Basisanforderungen ausgearbeitet, Nutzungsanforderungen erfasst, implementiert und getestet wurden, geht der Prozess in den Releaseabnahmetest über, der bereits aus dem IST-RE-Prozess bekannt ist. Abschließend folgt bei Usability-relevanten Vorhaben ein Test auf Usability und Ergonomie. Bei letzterem Test sollen allerdings ohne Ausnahme nochmals die Benutzerinnen und Benutzer hinzugezogen werden.

Verwaltung von Anforderungen

Die RE-Aktivität Verwaltung von Anforderungen ist, wie im IST-RE-Prozess, überall gegenwärtig. Allerdings stützt sich dies nicht mehr auf AMT, sondern auf Demand. Dies stellt einen wichtigen Vorteil gegenüber dem IST-RE-Prozess dar, da Demand über die Funktionalität von AMT hinaus auch die visuelle Darstellung von Abhängigkeiten zwischen Anforderungen beherrscht (REAM 9).

5 Fallstudie

In diesem Kapitel wird der in Kapitel 4 vorgestellte SOLL-RE-Prozess mit einer Fallstudie illustriert. Dazu wird ein Teilprojekt, angelaufen im Februar 2009, herangezogen, was sich mit der Automatisierung der Briefschreibung durch das ABS beschäftigt. Die abgeleiteten Anforderungen sind zu einem Großteil bereits implementiert und getestet, die Ergebnisse werden im nächsten Release von ABS übernommen.

In Kapitel 5.1 werden die elementaren Dokumente im IST-RE-Prozess anhand des Beispiels genauer erläutert. Die Reihenfolge richtet sich nach der Chronologie des Prozesses. Kapitel 5.2 zeigt auf, wie die im SOLL-RE-Prozess neu hinzugekommenen Dokumente für das gegebene Beispiel aussehen könnten und wie sich das Zusammenspiel derer mit dem bisher Geläufigen darstellt, um zu zeigen, wie sich die Änderungen des SOLL-RE-Prozesses gegenüber dem IST-RE-Prozess auswirken.

5.1 Bisherige Dokumente

Das Teilprojekt, das genauer betrachtet wird, basiert auf der Basisanforderung „Automatisierte Briefschreibung“. Diese beschreibt, dass das ABS Briefe nach entsprechenden Aktionen im Softwaresystem automatisiert erstellen soll und nicht, wie bisher, nur die Möglichkeit bietet, Briefe manuell durch ein Menü zu erstellen. Dies würde den Benutzerinnen und Benutzern Arbeitszeit sparen. Diese Tatsache erkannte auch die Programmleitung, die die genannte Basisanforderung mit Priorität 1 („unverzichtbar“) einordnete. Es sollten zunächst (in 2009) – und das ist auch der derzeitige Stand – die Briefe zur Ablehnung, Weglegung und EKA (Erschwerung und Klauseln Änderungsvorschlag) eines Antrags einer Lebensversicherung automatisiert durch das ABS verfasst werden. Ablehnung bedeutet, dass Allianz das Risiko die

Anforderungsbezeichnung
Automatisierte Briefbeschreibung – Weglegung und Ablehnung
Domäne für Begründung der Weglegung
Domäne für Begründung der Ablehnung
DB-Erweiterung für automatisierte Briefschreibung – Weglegung und Ablehnung
Konsolidierung der GUI-Erweiterung für Weglegung und Ablehnung
Invitatioverfahren Pilot Leben
Zuordnung Domäne j/n zu DE1_KZAblehnung
Business-Logik für Weglegung und Ablehnung (Prüfung, Begründung)

Tabelle 8: Abgeleitete Anforderungen zur Basisanforderung „Automatisierte Briefschreibung“

Kundin bzw. den Kunden zu versichern, als zu hoch einschätzt. Weggelegt heißt, dass der Antrag zu einem späteren Zeitpunkt wieder betrachtet wird, da die Kundin bzw. der Kunde derzeit z.B. ein Gebrechen vorweist, das Grund eines hohen Beitrags sein kann und dessen Heilung auf Wunsch der Kundin bzw. des Kunden abgewartet werden soll. Eine Weglegung kann auch folgen, wenn die Kundin bzw. der Kunde eine Auskunft über den Gesundheitszustand verweigert. EKA bedeutet, dass die Kundin bzw. der Kunde aufgrund der Risikoprüfung ein geändertes, neues Angebot erhält.

Aus der Basisanforderung wurden acht abgeleitete Anforderungen ausgearbeitet (Tabelle 8, aus AMT entnommen), die im Beschreibungsfeld genauer festlegen, was an ABS geändert werden muss, um die Basisanforderung umzusetzen. Dazu gehören formulierte Änderungen an der Datenstruktur wie auch an der Datenbank. Für den Fall einer

Invitatio (eine Sachbearbeiterin oder ein Sachbearbeiter macht einer Kundin oder einem Kunden ein Angebot) soll es nicht mehr möglich sein den Antrag abzulehnen (Abgeleitete Anforderung „Invitatioverfahren Pilot Leben“). Das Hauptaugenmerk dieser Fallstudie liegt auf der abgeleiteten Anforderung „Konsolidierung GUI-Erweiterung für Weglegung und Ablehnung“, da diese Anforderung die Merkmale des RE-Prozesses am besten herausstellt. Durch ein der genannten Anforderung angehängtes Worddokument, erstellt von Mitarbeiterinnen und Mitarbeitern der BO (aus dem Work Team) im Rahmen der Aktivität „Lösung entwerfen“ aus dem Unterprozess „Anforderungen implementieren“, wird sehr genau beschrieben, wie die GUI angepasst werden soll: Es muss möglich sein die Gründe der Ablehnung/Weglegung eingeben zu können, welche dann in den Brief übernommen werden. Dies wurde bisher direkt bei der manuellen Erstellung des Briefs eingegeben. Des Weiteren soll eine Checkbox hinzugefügt werden, um die Ablehnung zu einer gegebenenfalls zusätzlich beantragten Berufsunfähigkeitsversicherung ausdrücken zu können. Das schon vorhandene Feld zur Eingabe der Rückstelldauer bei Weglegung war bislang deaktiviert und soll aktiviert werden. Die Auswahlbox „Nicht-Policierung“ mit den Werten „Ablehnung“ und „Weglegung“ soll zu „Antragsstatus“ umbenannt werden und zusätzlich um den Wert „EKA“ ergänzt werden. Zuletzt soll das Kommentarfeld verkleinert, weiter unten platziert und alle zusammengehörigen Elemente in einer Box gruppiert werden.

All diese spezifizierten Änderungen veranschaulicht Abb. 14 (entnommen aus dem oben angesprochenen Anhang). Der Entwurf gründet sich auf den Erfahrungen, die mit dem ABS bisher gemacht wurden, und auf subjektives Empfinden. Dem Entwurf lag die in Abb. 15 dargestellte Maske des ABS zugrunde. Aus dem Entwurf entstand durch die Implementation schließlich die in Abb. 16 dargestellte Maske, welche ab Release 5.0 des ABS verfügbar sein wird.

The screenshot shows a software interface with the following elements and annotations:

- Navigation Bar:** VE | Produkt | Vergütung | Inkasso | Rechte | Fragen | VN | Zustellad. | Vertrag | S/L | Auftrag
- Header:** [User/ID] / [User/ID] / [User/ID] / HVP
- Tariff:** RisikoLebensversicherung LD(DL)
- Zuschläge in % (monatlicher Satz):**

Klasse	Autom.	Indiv.	Begründung	Gilt von	Gilt bis
Erschwerung	0,00	0,00		01.01.0001	01.01.0001
Sonderisiken	0,00	0,00		01.01.0001	01.01.0001
Beruf	0,00	0,00		01.01.0001	01.01.0001
Temporär		0,00		01.01.0001	01.01.0001
- Allgemeine Daten zur Risikoprüfung:**
 - Antragsstatus:** Dropdown menu with options: Ablehnung, Weglegung, EKA. Annotation: **umbenennen** (points to the dropdown), **neu** (points to EKA).
 - Begründung:** Dropdown menu.
 - Rückstellung (MM):** Input field. Annotation: **Editierbarkeit** (points to the field).
 - Ablehnung BU:** Checkbox. Annotation: **neu** (points to the checkbox).
 - Risikoverschlechterung:** Checkbox.
 - Erhöhungssperre:** Checkbox.
 - Untersuchung am:** Date field (01.01.0001).
- Kommentar:** Text area. Annotation: **Kommentarfeld verkleinern** (points to the text area).
- Anomalien:** Table with columns: Nummer, Bezeichnung.

Abb. 14: Mock-Up der Änderungen in der entsprechenden Maske aus ABS

VP Produkt Vergütung Inkasso Rechte Fragen VN Zustelladr. Vertrag S/L Auftrag

Duck Daisy / weiblich / 15.05.1980 / HVP P

Tarif RisikoLebensversicherung L0(DL) Autom. Indiv. Liste
 Berufsgruppe -- -- Detail

Zuschläge in ‰ (monatlicher Satz)

Klasse 0 gpol.Ent. --

	Autom.	Indiv.	Begründung	Gilt von	Gilt bis
Erschwerung	0,00	0,00		01.01.0001	01.01.0001
Sonderisiken	0,00	0,00		01.01.0001	01.01.0001
Beruf	0,00	0,00		01.01.0001	01.01.0001
Temporär		0,00		01.01.0001	01.01.0001

Kommentar

Nicht-Policierung Begründung Rückstellung (MM)

Ausschlussklausel vorhanden
 Risikoverschlechterung Erhöhungssperre Untersuchung am 01.01.0001

Anomalien

Nummer	Bezeichnung

Abb. 15: Bisherige Maske aus ABS

VP Produkt Vergütung Inkasso Rechte Fragen VN Zustelladr. Vertrag S/L Auftrag

/ / / HVP P

Tarif RisikoLebensversicherung L0(DL) Autom. Indiv. Liste
 Berufsgruppe -- -- Detail

Zuschläge in ‰ (monatlicher Satz)

Klasse 0 gpol.Ent. --

	Autom.	Indiv.	Begründung	Gilt von	Gilt bis
Erschwerung	0,00	0,00		01.01.0001	01.01.0001
Sonderisiken	0,00	0,00		01.01.0001	01.01.0001
Beruf	0,00	0,00		01.01.0001	01.01.0001
Temporär		0,00		01.01.0001	01.01.0001

Allgemeine Daten zur Risikoprüfung

Antragsstatus Begründung Rückstellung (MM)

Ausschlussklausel vorhanden Ablehnung BU
 Risikoverschlechterung Erhöhungssperre Untersuchung am 01.01.0001

Kommentar

Anomalien

Nummer	Bezeichnung

Abb. 16: Neue Maske, verfügbar ABS Release 5.0

5.2 Neuerungen

Im Folgenden werden die Neuerungen durch den SOLL-RE-Prozess anhand von Beispieldokumenten näher erläutert. Diese Beispiele wurden von mir erstellt. Sie basieren auf einer Vorführung der relevanten Arbeitsabläufe. Jegliche personenbezogenen Daten sind frei erfunden und beruhen nicht auf wahren Begebenheiten.

Die wichtigste Erkenntnis des SOLL-RE-Prozesses und zugleich der wichtigste Unterschied hinsichtlich Gebrauchstauglichkeit ist, dass der Entwurf der Maske (Abb. 14) nicht mehr nur auf der Erfahrung und der Empfindung der Mitarbeiterinnen und Mitarbeiter des Work Teams aufbauen wird. Als fundierte Grundlage, um eine möglichst gebrauchstaugliche Benutzungsschnittstelle zu schaffen, führen die beiden Usability-Verantwortlichen von BO und IT im Rahmen der Phase Anforderungsanalyse des ABS Vorgehensmodells zunächst Interviews mit Benutzerinnen und Benutzern des ABS, um Kontextszenarien zu erheben. Ein solches Kontextszenario könnte wie folgt lauten:

Kontextszenario „Antragsbearbeitung Risikolebensversicherung“

Frau Petra Müller arbeitet seit drei Jahren als Sachbearbeiterin bei Allianz Deutschland in einer Niederlassung in Berlin. Ihr Fachgebiet sind Risikolebensversicherungen. Als Sachbearbeiterin hat sie direkten Kundenkontakt. Dabei muss sie vor allem in den Kernzeiten zwischen 9 Uhr und 12 Uhr sowie zwischen 13 Uhr und 16 Uhr für ihre Kundinnen und Kunden erreichbar sein. Frau Müller hat im Jahr 2000 eine Ausbildung zur Versicherungskauffrau absolviert, sie ist nun 28 Jahre alt.

Typischerweise rufen Frau Müllers Kundinnen und Kunden bei ihr an, falls sie etwas an ihrem Versichertenstatus ändern möchten. Frau Müller versucht dann einen Termin für ein persönliches Kundengespräch auszumachen, um in direkten Kontakt zu treten. Sollte dies nicht möglich sein, so sendet Frau Müller notwendige Unterlagen per Post an die Kundinnen und Kunden. [...] Trifft ein Antrag auf eine Risikolebensversicherung bei Frau Müller ein, so öffnet sie die Akte der versicherten Person (kurz VP) in ABS. Sollte die Person neu bei Allianz sein, [...]. Frau Müller pflegt die Daten aus dem ihr vorliegenden Antrag in ABS ein. Bei der Risikobewertung muss sie dazu eine Vielzahl von gesundheitlichen Informationen über die VP abgeben. Dazu gehören [...]. Hierbei kann es öfters vorkommen, dass wichtige Informationen (das sind [...]) fehlen. Sollte dies der Fall sein, so versucht Frau Müller telefonisch nachzufragen. Will die VP keine Angabe machen, so bricht Frau Müller die Eingabe ab und wählt im Feld „Nicht-Policierung“ „Weglegung“ aus. Sie schreibt sich gesondert auf, bis wann etwa der Antrag weggelegt wird, um dann wieder auf die VP zugehen zu können, wenn die diese Zeit abgelaufen ist. Daraufhin muss sie im Menü zur Briefschreibung manuell einen Brief erstellen, indem sie [...] die Begründung für die Weglegung eingibt. Sollten bei der Eingabe keine Probleme auftreten, so schließt sie diese ab und wartet auf das Ergebnis der Berechnung. Sollte das Risiko laut Berechnung zu hoch sein, so wählt Frau Müller bei „Nicht-Policierung“ den Wert „Ablehnung“ aus. Auch hier muss sie im Anschluss über das Menü die Erstellung des Briefs für die VP vornehmen. Frau Müller ist es dabei schon oft passiert, dass sie bei der Erstellung des Briefs einen kleinen Fehler begangen hat und diesen dann vor dem Druck nochmals berichtigen musste. Sie sagt, dass man sich durch die vielen Buttons auf der Oberfläche „ja mal schnell verklicken“ könne. Durch das monotone Eintippen am PC, so sagt sie weiter, könne auch hin und wieder die Kon-

zentration etwas schwinden. Daher ist sie von der Notwendigkeit, die Briefe vor dem Druck und der anschließenden Zustellung an die VP nochmals durchzugehen, fest überzeugt. [...]

Aus diesem und anderen Kontextszenarien leitet die bzw. der BO-Usability-Verantwortliche in der Aktivität „Nutzungsanforderungen erfassen“ zunächst Erfordernisse ab und verarbeitet diese dann zu Nutzungsanforderungen. Tabelle 9 zeigt exemplarisch, wie dies aussehen könnte.

Nr.	Kontextszenario (Ausschnitt)	Erfordernis	Nutzungsanforderung
1	[...], so bricht Frau Müller die Eingabe ab und wählt im Feld „Nicht-Policierung“ „Weglegung“ aus. Sie schreibt sich gesondert auf, bis wann etwa der Antrag weggelegt wird, [...]	Frau Müller muss, um zeitgerecht wieder auf Ihre Kundinnen und Kunden zugehen zu können, die Dauer der Weglegung festhalten.	Die Benutzerin bzw. der Benutzer muss am System Übersicht über die derzeitigen Weglegungszeiträume erhalten.
2			Die Benutzerin bzw. der Benutzer muss durch das System benachrichtigt werden, wenn die Dauer eines Weglegungszeitraums abgelaufen ist.
3	Daher ist sie von der Notwendigkeit, die Briefe vor dem Druck und der anschließenden Zustellung an die VP nochmals durchzugehen, fest überzeugt.	Frau Müller muss einen Brief, bevor er gedruckt wird, ansehen und bearbeiten können, um gegebenenfalls auftretende Fehler zu korrigieren.	Die Benutzerin bzw. der Benutzer muss vor dem Druck eines Briefs eine Ansicht desselben erhalten und Korrekturen vornehmen können.

Tabelle 9: Erfordernisse und Nutzungsanforderungen

Die Erfassung von Nutzungsanforderungen, genau wie die parallel laufende Ausarbeitung der Basisanforderungen, kann in mehreren Sprints erfolgen. Eine sinnvolle Auftrennung in Sprints könnte z.B. anhand von Basisanforderungen sein, so dass pro Sprint eine Basisanforderung ausgearbeitet und Nutzungsanforderungen diesbezüglich erfasst werden. Allgemein sollten wichtige Anforderungen dabei zuerst betrachtet werden, so dass diese auch zuerst implementiert werden. Auf diese Weise lassen sich auch enge Zeitrahmen bewältigen, da die wichtigen Anforderungen zuerst erfasst und sodann implementiert werden. In späteren Sprints können weitere Anforderungen, die nicht sehr erfolgskritisch sind („nice to have“), erfasst werden.

Die erfassten Anforderungen (sowohl abgeleitete Anforderungen, als auch Nutzungsanforderungen) werden unter der Verantwortung der Facharchitektur überprüft. Der BO Product Owner prüft dabei die in diesem Sprint neu erfassten Anforderungen auf Konsistenz zu den bisherigen.

Aus den in einem Sprint erfassten Nutzungsanforderungen und den abgeleiteten Anforderungen erstellt das Work Team im Unterprozess „Anforderungen implementieren“ zunächst Virtual Windows. Diese bilden die benötigten Daten auf möglichst wenige Masken ab. Die im vorherigen Kapitel erwähnte neue Maske aus ABS Release 5.0 (Abb. 16) repräsentiert ohne die seitlichen Registerleisten ein gut ausgearbeitete-

tes Beispiel eines solchen Virtual Windows. Es zeigt ausschließlich Daten und keine Funktionen. Darüber hinaus könnte es im Beispiel durch eine weitere Maske, welche den Brief zeigt und Veränderungen an selbigem zulässt, ergänzt werden.

Der nächste Schritt im SOLL-RE-Prozess sieht vor, dass das Work Team Benutzerinnen und Benutzern im Rahmen eines Understandability-Tests die Virtual Windows vorlegen. Die Benutzerin bzw. der Benutzer soll daran erläutern, was sie bzw. er denkt was auf der Maske zu sehen sei, um zu prüfen, ob sich das intuitive Verständnis von Benutzerinnen und Benutzern mit dem der Entwicklerinnen und Entwickler deckt. Wenn beispielsweise das genannte Virtual Window ähnlich zu Abb. 16 einer Experten-Benutzerin bzw. einem Experten-Benutzer vorgelegt wird, so könnte diese bzw. dieser anmerken, dass die verwendeten Abkürzungen nicht sofort darauf schließen lassen, was darunter zu verstehen ist, weil sie bzw. er oft von Kolleginnen und Kollegen nach der Bedeutung von Abkürzungen gefragt wird. Daraufhin könnte das Work Team das Virtual Window anpassen, indem es z.B. die Abkürzungen ausschreibt oder visuell andeutet, dass die Benutzerinnen und Benutzer bei Berührung einer Abkürzung mit der Maus eine Erklärung angezeigt bekommen.

Die Virtual Windows werden im nächsten Schritt des SOLL-RE-Prozesses durch das Work Team zu fertigen Prototypen ausgearbeitet. Das heißt, dass Funktionen aufgenommen werden, wie z.B. auch die Registerleisten am oberen und rechten Rand, die man in Abb. 16 sieht. Darüber hinaus könnte die Gruppe „Allgemeine Daten zur Risikoprüfung“ um eine Funktion zum Anstoß der automatischen Briefschreibung ergänzt werden. Dies wäre durch die Nutzungsanforderung Nr. 3 aus Tabelle 9 zu begründen, die der Basisanforderung insofern widerspricht, als dass die Briefschreibung zwar nicht vollautomatisch, wie es nun von Allianz implementiert wurde, dafür jedoch transparent und nachvollziehbar für die Benutzerinnen und Benutzer, erfolgt. Die modifizierte Maske wird in Abb. 17 gezeigt. Bei Klick auf das Brieficon (siehe rote Markierung) wird eine Ansicht des generierten Briefs geöffnet, die Änderungen von

The screenshot shows a software interface with a menu bar at the top containing: VP, Produkt, Vergütung, Inkasso, Rechte, Fragen, VN, Zustelladr., Vertrag, S/L, Auftrag. Below the menu bar, there are several input fields and dropdown menus, including 'Tarif' (set to 'RisikoLebensversicherung LD(DL)'), 'Autom.', 'Indiv.', and 'Berufsgruppe'. A table titled 'Zuschläge in % (monatlicher Satz)' is visible, with columns for 'Klasse', 'Autom.', 'Indiv.', 'Begründung', 'Gilt von', and 'Gilt bis'. The table contains rows for 'Erschwerung', 'Sonderrisiken', 'Beruf', and 'Temporär'. Below the table is a section titled 'Allgemeine Daten zur Risikoprüfung' with fields for 'Antragsstatus', 'Begründung', and 'Rückstellung (MM)'. There are also checkboxes for 'Ausschlussklausel vorhanden', 'Ablehnung BU', 'Risikoverschlechterung', 'Erhöhungssperre', and 'Untersuchung am'. A red circle highlights an envelope icon next to the 'Rückstellung (MM)' field. At the bottom, there is a 'Kommentar' field and an 'Anomalien' table with columns 'Nummer' and 'Bezeichnung'.

Abb. 17: Paper-Mock-Up der neuen Maske; Änderungen gegenüber Abb. 16 rot markiert

selbigem zulässt. Bei Bestätigung durch die Benutzerinnen und Benutzer erfolgt die endgültige Erstellung.

Der in Abb. 17 gezeigte Paper-Mock-Up und die zusätzlich angesprochene Maske zur Ansicht und Korrektur des Briefs legt das Work Team im Rahmen des auf die Erstellung der Paper-Mock-Ups folgenden Usability-Tests Benutzerinnen und Benutzern vor und prüft die Masken so auf Gebrauchstauglichkeit. Sollten dort keine Probleme festgestellt werden, können die Masken entsprechend der Paper-Mock-Ups implementiert werden. Abschließend folgt noch der Test der implementierten Anforderungen. Sollten zwischenzeitlich durch den parallel laufenden Unterprozess „Basisanforderungen ausarbeiten und Nutzungsanforderungen erfassen“ neue Anforderungen dokumentiert worden sein, so wird der Unterprozess zur Implementation mittels eines weiteren Sprints wiederholt. Dabei werden die bisherigen Dokumente, wie beispielsweise die Virtual Windows, erweitert. Im Beispiel ist allerdings kein zweiter Sprint notwendig.

Gemäß dem SOLL-RE-Prozess sollten in größeren Teilprojekten die Möglichkeit zur iterativen, agilen Vorgehensweise wahrgenommen werden. Bei dem gezeigten Beispiel ist der Umfang so gering, dass keine Veränderung in Hinsicht der Agilität sichtbar wird.

6 Zusammenfassung & Ausblick

Als Vorarbeit zum Entwurf eines RE-Prozesses mit Fokus auf Gebrauchstauglichkeit für das ABS wurde zunächst die IST-Situation bei Allianz untersucht. Die Untersuchung ergab, dass die Benutzerinnen und Benutzer des ABS nicht nur im RE-Prozess, sondern auch im gesamten Entwicklungsprozess unzureichend eingebunden werden. Nur am Ende der Entwicklung oder noch später, bei laufenden Produkktivsystemen, werden Benutzerinnen und Benutzer im Rahmen von Usability-Tests mit einbezogen. In den Grundlagen wurde deutlich gemacht, dass Gebrauchstauglichkeit vor allem dann erreicht wird, wenn die Benutzerinnen und Benutzer möglichst früh in den Entwicklungsprozess eingebunden werden. Dieser Maßgabe folgend wurden im entworfenen SOLL-RE-Prozess Aktivitäten zur Bestimmung von Kontextszenarien, zur Erfassung von Nutzungsanforderungen und zur Erstellung von Virtual Windows als sehr frühen Prototyp der Benutzungsschnittstelle eingeführt. Darüber hinaus wurde der RE-Prozess um Aktivitäten zur Prüfung der Virtual Windows auf Verständnis durch die Benutzerinnen und Benutzer, zur Erstellung auf Virtual Windows basierender Paper-Mock-Ups und zuletzt zur Prüfung dieser Paper-Mock-Ups erweitert. Diese Auswahl an Aktivitäten wurde speziell für Allianz abgestimmt und in den RE-Prozess integriert. Entsprechend des Umfangs eines Teilprojekts ist es möglich Aktivitäten auszusparen oder in deren Umfang zu reduzieren. Diese Möglichkeit stützt sich auch auf die Überarbeitung des RE-Prozesses hinsichtlich Agilität. Durch die zahlreichen Iterationsmöglichkeiten kann mit einer stärkeren Priorisierung gearbeitet und so erst sehr wichtige bis hin zu unwichtigeren Anforderungen betrachtet werden. Unter den Gesichtspunkten von Agilität ist die IST-Situation des RE-Prozesses bei Allianz bereits gut. Bei der Untersuchung konnten nur kleinere Defizite, die größtenteils die Dokumentation von Anforderungen betreffen, entdeckt werden. Abhängigkeiten zwischen Anforderungen werden nicht visuell dargestellt und Anforderungen werden nicht systematisch auf Konsistenz geprüft. Die Prüfung auf Konsistenz wurde entsprechend in den SOLL-RE-Prozess mit aufgenommen, das Problem der Darstellung der Abhängigkeiten zwischen Anforderungen wird durch den Einsatz von HP Demand behoben.

Der SOLL-RE-Prozess wurde gemäß der Zielsetzung gegenüber dem IST-RE-Prozess in seinen Aktivitäten stärker parallelisiert und agilisiert, so dass Ausarbeitung, Umsetzung und Prüfung der Umsetzung von Anforderungen Hand in Hand iterativ durchgeführt werden können. Auch die Analyse der IST-Prozesse (Geschäftsprozesse) zu Beginn eines Teilprojekts kann im SOLL-RE-Prozess iterativ durchgeführt werden, um die Vorteile von Agilität auszunutzen. Dennoch verbietet der neue RE-Prozess die bisherige Weise der Durchführung (strikt sequentiell) nicht und erleichtert damit den Übergang zu der agilen Variante wie auch die Durchführung kleinerer Teilprojekte, die keiner übermäßigen agilen Durchführung bedürfen.

Die Vorteile des neuen RE-Prozesses sind eine konsequent agile Vorgehensweise – nicht nur während der Implementation (außerhalb des RE-Prozesses), sondern auch in der Vorphase und der Anforderungsanalyse im ABS Vorgehensmodell – und die damit verbundenen positiven Effekte, wie Verbesserung von Return of Investment, Verbesserung der Mitarbeitermoral und Verbesserung der Kundenzufriedenheit. Darüber hinaus verspricht der deutliche Fokus des SOLL-RE-Prozesses auf Gebrauchstauglichkeit eine höhere Zufriedenheit der Benutzerinnen und Benutzer des ABS. Das hat nicht nur den Effekt, dass die Benutzerinnen und Benutzer gerne mit dem ABS arbeiten, sondern auch, dass kostspielige Teilprojekte zur nachträglichen Verbesserung der Gebrauchstauglichkeit auf lange Sicht nicht mehr notwendig sein wer-

den. Der neue RE-Prozess behebt Defizite bei der Umsetzung der CMMI-Prozessfelder Requirements Management und Requirements Development und nähert sich gleichermaßen der Erfüllung aktueller Normen, wie DIN EN ISO 9241-210 an. Der Mehraufwand, den der SOLL-RE-Prozess verlangt, muss also nicht als „verloren“ angesehen werden, sondern viel eher als „gut investiert“.

Im Rahmen dieser Arbeit war es leider nicht möglich den SOLL-RE-Prozess ausführlich zu prüfen und abzustimmen. Ich empfehle Allianz daher, den von mir vorgestellten SOLL-RE-Prozess in einem geeigneten Projekt zu pilotieren und Feinheiten im RE-Prozess entsprechend der dort gewonnenen Erfahrungen abzustimmen. Ein solches Projekt sollte mehrere Basisanforderungen umfassen, so dass die agilen Elemente des RE-Prozesses zum Tragen kommen. Ein gesondertes Komitee, besetzt aus mit dem RE-Prozess vertrauten Personen, bestenfalls aus BO, IT und Kompetenzzentrum für Ergonomie und Usability, sollte die Durchführung des Projekts begleiten und regelmäßiges Feedback der Beteiligten einholen, sowie unterstützend eingreifen, da davon ausgegangen werden muss, dass die erstmalige Durchführung des RE-Prozesses *ohne* Anleitung nicht erfolgreich sein wird. Durch das regelmäßige Feedback werden Schwachstellen und Probleme im RE-Prozess identifiziert. Am Ende des Projekts ist es zunächst notwendig das Prozessmodell zu konsolidieren. Es gilt festzuhalten, welche Aktivitäten so nicht durchführbar sind bzw. wo Änderungen notwendig werden. Diese Änderungen sollten durchgeführt und je nach Ausmaß nochmals eine Evaluation durchgeführt werden. Fallen keine Änderungen mehr an, so sollten Schulungen für beteiligte Mitarbeiterinnen und Mitarbeiter abgehalten werden, damit der Prozess auch gelebt und bei der Ausführung nicht vom Modell abgewichen wird.

A Glossar

ABS

ABS (Allianz Business System) ist ein spartenübergreifendes IT-System, welches vor allem Sach-, Leben- und Krankenversicherungsvorgänge unterstützt. Es bietet eine ganzheitliche Kundensicht und ersetzt damit die verteilten Bestandssysteme.

Auftraggeberin bzw. Auftraggeber

Auftraggeberin bzw. Auftraggeber ist im Kontext dieser Arbeit synonym zu Kundin bzw. Kunde zu verstehen. Diese Rolle ist im Kontext von ABS durch die Betriebsorganisation (BO) besetzt.

Benutzerin bzw. Benutzer

Die Benutzerinnen und Benutzer sind im Kontext dieser Arbeit Sachbearbeiterinnen und Sachbearbeiter in ihren jeweiligen Dienstleistungsgebieten, z.B. Leben.

Benutzungsschnittstelle

Siehe „GUI“.

Entity-Relationship-Diagramm

Ein Entity-Relationship Diagramm ist ein Diagramm, welches Entitäten mittels Beziehungen in Verbindung setzt. Es wird oft im Rahmen der Datenbank-Modellierung eingesetzt.

Experten-Benutzerin bzw. Experten-Benutzerin

Eine Experten-Benutzerin bzw. ein Experten-Benutzer ist jemand mit einem größeren Überblick über ein Softwaresystem und Wissen über schwierige Situationen und gilt als direkte Anlaufstelle bei Problemen „normaler“ Benutzerinnen und Benutzer [25].

Gestalt law

Gestalt laws beschreiben die intuitive Wahrnehmung von Formen oder Objekten. Ein Beispiel ist das Gesetz der Nähe, welches besagt, dass nahe beieinander liegende Objekte als zusammengehörig wahrgenommen werden [25].

GUI

GUI (Graphical User Interface, deutsche Übersetzung: grafische Benutzungsschnittstelle) ist der Teil eines Softwaresystems, der es den Benutzerinnen und Benutzern erlaubt über eine grafische Oberfläche mit dem Softwaresystem zu interagieren.

Interaktives Softwaresystem

Siehe auch „Softwaresystem“.

Über die Definition eines Softwaresystems hinaus, wird der Zusatz „interaktiv“ im Kontext dieser Arbeit dann benutzt, wenn herausgestellt werden soll, dass das Softwaresystem sich besonders durch eine *gute* Benutzungsschnittstelle auszeichnet. Gut ist dabei im Sinne von Gebrauchstauglichkeit zu verstehen.

Kundin bzw. Kunde

Siehe „Auftraggeberin bzw. Auftraggeber“.

Nutzungsanforderung

Eine Nutzungsanforderung ist „eine Benutzeraktion, die am System ausführbar sein muss, um ein Erfordernis im Nutzungskontext effizient zu erfüllen“ [18].

Ein Erfordernis ist „eine notwendige Voraussetzung, die es ermöglicht, den in einem Sachverhalt (Nutzungskontext) enthaltenen Zweck effizient zu erfüllen“ [18].

Paper-Mock-Up

Ein Paper-Mock-Up ist ein Papier-basierter Prototyp einer Benutzungsschnittstelle. Er kann mit Stift und Papier, aber auch mit Werkzeugen erstellt werden. Er soll die späteren Masken des Softwaresystems widerspiegeln, um bereits früh im Entwicklungsprozess Usability-Tests durchführen zu können und so der kostspieligen, späten Behebung von Fehlern vorzubeugen.

Product Owner

Die Rolle des Product Owners entstammt aus dem agilen Framework Scrum (siehe dazu auch Kapitel 2.2.2.1). Sie ist im Kontext dieser Arbeit mit der Rolle des Teilprojektleiters jeweils auf Seiten der Betriebsorganisation und IT gleichzusetzen. Sie verkörpert die Interessen der Stakeholder, insbesondere der Kundinnen und Kunden.

Release

Ein Release bezeichnet eine fertige und veröffentlichte Version eines Softwaresystems.

Semantische Funktion

Semantische Funktionen sind solche, deren Aufruf „ernste“ Folgen hat. Beispiele dafür sind Änderungen persistenter Daten oder das Senden von Informationen an die Umgebung [25].

Softwaresystem

Ein Softwaresystem im Kontext dieser Arbeit ist ein auf Middleware und Basissoftware aufsetzende Anwendungssoftware, das eine direkt nutzbare Funktionalität bereit stellt.

Stakeholder

Ein Stakeholder eines Systems ist eine Person oder Organisation, welche (direkt oder indirekt) Einfluss auf die Anforderungen des betrachteten Systems hat [51].

Use Case-Diagramm

Ein Use Case-Diagramm ist ein Diagramm, welches die Aktoren eines Softwaresystems mit dessen Anwendungsfällen in Beziehung setzt.

V-Modell XT

Das V-Modell XT ist ein Vorgehensmodell zur Planung und Durchführung von Softwareentwicklungsprojekten.

Virtual Window

Ein „Virtual Window“ ist ein Bild auf einem idealisierten Bildschirm, eine Benutzer-orientierte Darstellung von persistenten Daten. Virtual Windows sind auf Papier. Sie zeigen Daten, haben aber keine Buttons, Menüs oder andere Funktionen [25; 28].

B Abbildungsverzeichnis

Abb. 1: Konzeptueller Prozess für SPI	3
Abb. 2: Scrum Prozess nach [10]	8
Abb. 3: Sicht auf Softwaresystem nach [18]	12
Abb. 4: „Interdependence of human-centred design activities“ aus DIN EN ISO 9241-210 [25].....	16
Abb. 5: „Virtual Windows“ Ansatz nach [17].....	17
Abb. 6: Beispiel eines „Task & Support approach“ [25]	18
Abb. 7: Beispiel eines Virtual Windows Plans [28].....	19
Abb. 8: Usability Engineering-Prozess nach Geis	22
Abb. 9: Überblick über ABS Vorgehensmodell	26
Abb. 10: Kommentierter Screenshot von AMT aus [28].....	30
Abb. 11: Übersicht über IST-RE-Prozesses	33
Abb. 12: GQM-Plan zu SPI.....	37
Abb. 13: „Agile“ bei Allianz.....	44
Abb. 14: Mock-Up der Änderungen in der entsprechenden Maske aus ABS	59
Abb. 15: Bisherige Maske aus ABS.....	60
Abb. 16: Neue Maske, verfügbar ABS Release 5.0.....	60
Abb. 17: Paper-Mock-Up der neuen Maske; Änderungen gegenüber Abb. 16 rot markiert.....	63

C Tabellenverzeichnis

Tabelle 1: Praktiken zur Integration von RE und agilen Methoden	15
Tabelle 2: Auswertung der Kontextszenarien (mit Beispiel)	23
Tabelle 3: Inspektionsbericht (mit Beispiel)	23
Tabelle 4: Praktiken zur Optimierung eines Softwareentwicklungsprozesses bezüglich Gebrauchstauglichkeit	25
Tabelle 5: Bewertung des IST-RE-Prozesses bezüglich traditionellem und agilen Vorgehen.....	42
Tabelle 6: Anwendbarkeit der REAM-Praktiken bei Allianz	46
Tabelle 7: Anwendbarkeit der GEBR-Praktiken bei Allianz.....	48
Tabelle 8: Abgeleitete Anforderungen zur Basisanforderung „Automatisierte Briefschreibung“	58
Tabelle 9: Erfordernisse und Nutzungsanforderungen	62

D Abkürzungsverzeichnis

ABS *Allianz Business System*
ADAG *Allianz Deutschland AG*
AMIS *Agentur Management- und Informationssystem*
AMT *Anforderungsmanagement-Tool*
BO *Betriebsorganisation*
CMM *Capability Maturity Model*
CMMI *Capability Maturity Model Integration*
DB *Datenbank*
GQM *Goal-Question-Metric*
GUI *Graphical User Interface*
IT *Informationstechnik*
LD *Lean Development*
PL *Projektleiter*
PO *Product Owner, Product Owner*
RD *Requirements Development*
RE *Requirements Engineering*
REQM *Requirements Management*
SEI *Software Engineering Institute*
SOX *Sarbanes-Oxley Act*
SPI *Software Process Improvement*
SPICE *Software Process Improvement and Capability dEtermination*
XP *eXtreme Programming*
ZBM *Zielbetriebsmodell*

E Literaturverzeichnis

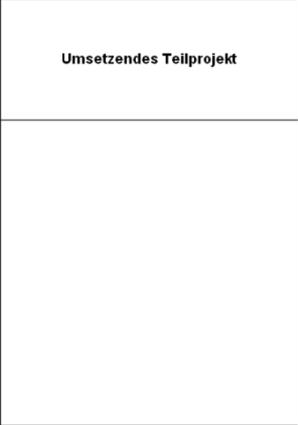
- [1] **Glinz, Martin.** Eine geführte Tour durch die Landschaft der Software-Prozesse und -Prozessverbesserung. *Informatik*. 1999, 6, pp. 6-15.
- [2] **Gorschek, Tony.** *Software Process Assessment & Improvement in Industrial Requirements Engineering*. Karlskrona : Blekinge Inst. of Technology, 2004. 978-91-7295-041-2.
- [3] **Stienen, Hans.** Nach CMM und Bootstrap: SPiCE; Die neue Norm für Prozessbewertung. *Informatik*. 1999, 6, pp. 16-22.
- [4] **Sommerville, Ian and Sawyer, Pete.** *Requirements Engineering: A Good Practice Guide*. s.l. : John Wiley & Sons, 1997. 978-0471974444.
- [5] **Basili, Victor R., Caldiera, Gianluigi and Rombach, H. Dieter.** The Goal Question Metric Approach. [book auth.] John J. Marciniak. *Encyclopedia of Software Engineering*. s.l. : John Wiley & Sons, 1994, pp. 528-532.
- [6] **Briand, Lionel C., Differding, Christiane M. and Rombach, H. Dieter.** Practical Guidelines for Measurement-Based Process Improvement. *Software Process: Improvement and Practice*. 1997, Vol. 2, 4.
- [7] **Paetsch, Frauke, Eberlein, Armin and Maurer, Frank.** Requirements Engineering and Agile Software Development. *Proceedings on the Twelfth IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. June 9-11, 2003, pp. 308-313.
- [8] **Pohl, Klaus and Rupp, Chris.** *Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. Heidelberg : dpunkt.verlag, 2009. 978-3-89864-613-0.
- [9] **Fowler, Martin and Highsmith, Jim.** The Agile Manifesto. *Software Development*. August 8, 2001.
- [10] **Highsmith, Jim and Highsmith, James A.** *Agile Software Development Ecosystems (Agile Software Development Series)*. Boston : Addison-Wesley, 2002. 978-0201760439.
- [11] **Pichler, Roman.** *Scrum - Agiles Projektmanagement erfolgreich einsetzen*. Heidelberg : dpunkt.verlag, 2008. 978-3-89864-478-5.
- [12] **Nawrocki, Jerzy, et al.** Extreme Programming Modified: Embrace Requirements Engineering Practices. *Proceedings IEEE Joint International Conference on Requirements Engineering*. September 9-13, 2002, pp. 303-310.
- [13] **Schwaber, Ken and Sutherland, Jeff.** *Scrum*. [Guide] 2010. <http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf#view=fit>.
- [14] **Ktata, Oualid and Levesque, Ghislain.** Agile development: Issues and avenues requiring a substantial enhancement of the business perspective in large projects. *Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering (C³S²E)*. May 19-21, 2009, pp. 59-66.
- [15] **Kotonya, Gerald and Sommerville, Ian.** Requirements Engineering with Viewpoints. *Software Engineering Journal*. January 1996, Vol. 11, 1, pp. 5-18.
- [16] **Paech, Barbara and Heinrich, Robert.** *Komponenten-basierte und Service-orientierte Systeme*. [Lecture] Heidelberg : Software Engineering Group - Institute for Computer Science - University of Heidelberg, 2010.
- [17] **Eberlein, Armin and do Prado Leite, Julio Cesar Sampaio.** Agile Requirements Definition: A View from Requirements Engineering. *International Workshop on Time-Constrained Requirements Engineering (TCRE)*. September 9, 2002.

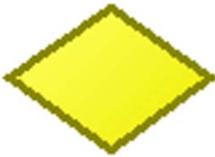
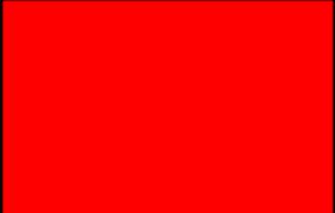
- [18] **Rodriguez, Pilar, et al.** Some Findings Concerning Requirements in Agile Methodologies. [ed.] Frank Bomarius, et al. *Product-Focused Software Process Improvement: Proceedings on the 10th International Conference, PROFES*. June 15-17, 2009, pp. 171-184.
- [19] **Sommerville, Ian.** *Software Engineering*. 8th Edition. s.l. : Addison-Wesley, 2006. 978-0321313799.
- [20] **Sillitti, Alberto, et al.** Managing Uncertainty in Requirements: a Survey in Documentation-driven and Agile Companies. *11th IEEE International Software Metrics Symposium*. September 19-22, 2005, pp. 10-17.
- [21] **Araujo, Joao and Ribeiro, Joao Carlos.** Towards an Aspect-Oriented Agile Requirements Approach. *Eighth International Workshop on Principles of Software Evolution*. September 5 - 6, 2005, pp. 140-143.
- [22] **DIN EN ISO 9241-11.** *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten - Teil 11: Anforderungen an die Gebrauchstauglichkeit; Leitsätze*. [Document] 1998.
- [23] **DIN EN ISO 9241-110.** *Ergonomische Anforderungen der Mensch-System-Interaktion Teil 110: Grundsätze der Dialoggestaltung*. [Document] 2008.
- [24] **Paech, Barbara and Borner, Lars.** *Software Engineering I*. [Lecture] Heidelberg : Software Engineering Group - Institute for Computer Science - University of Heidelberg, 2008.
- [25] **Lauesen, Soren.** *User Interface Design; A Software Engineering Perspective*. s.l. : Addison Wesley, 2005. 978-0321181435.
- [26] **DIN EN ISO 9241-210.** *Ergonomie der Mensch-System-Interaktion Teil 210: Prozess zur Entwicklung gebrauchstauglicher interaktiver Systeme*. [Document] 2010.
- [27] **Lauesen, Soren.** *Mini-course in the design method*. [Presentation] February 28, 2009. http://www.itu.dk/~slauesen/UID/UID_course.pps.
- [28] **Lauesen, Soren and Harning, Morten Borup.** Virtual windows: linking user tasks, data models, and interface design. *IEEE Software*. Jul/Aug 2001, Vol. 18, 4, pp. 67-75.
- [29] **Geis, Thomas.** Usability Engineering und Usability Testing.
- [30] **Allianz.** *Vorgehensmodell ABS*. [Presentation] 2009.
- [31] —. *Ein Leitfaden zum Anforderungsmanagement ABS; Anwendungsdokumentation AMT*. [Presentation] 2009.
- [32] —. *Vorgehensmodell zur Software-Ergonomie und Usability*. [Presentation] 2008.
- [33] —. *Software-Ergonomie und Usability; Schulung für Projektleiter und Führungskräfte*. [Presentation] 2007.
- [34] —. *ABS Entwicklung - Überblick*. [Document] 2010.
- [35] —. *ABS Schulung Fachdesign*. [Presentation] 2010.
- [36] **Verwaltungs-Berufsgenossenschaft (VBG).** VGB. [Online] [Cited: June 20, 2010.] http://www.vbg.de/imperia/md/content/produkte/spschriften/bgi_650_.pdf.
- [37] **BOC Information Technologies Consulting.** ADONIS Geschäftsprozessmanagement-Toolkit. [Online] [Cited: May 2, 2010.] http://www.boc-group.com/index.jsp?file=WP_582571cc1ed802de.46e381.f59775478f.-7f17&lg=de.
- [38] **HP.** HP Quality Center. [Online] [Cited: April 25, 2010.] https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bt_o&cp=1-11-127-24_4000_5__.

- [39] **microTOOL**. objectiF - das Tool für die modellgetriebene Softwareentwicklung mit der UML. [Online] [Cited: April 25, 2010.] <http://www.microtool.de/objectiF/de/index.asp>.
- [40] **Allianz**. *ABS-Kernentwicklung; Styleguide; Spezifikation, Richtlinien und Empfehlungen für die benutzerzentrierte Gestaltung*. [Document] 2009.
- [41] **Software Engineering Institute**. *CMMI for Development Version 1.2*. [Document] 2006.
- [42] **Allianz**. *ErgoNorm - Benutzerfragebogen zu "Arbeit & Software"*. [Document]
- [43] **Sutherland, Jeff**. *ScrumButt Test aka "The Nokia Test"*. [Presentation] 2008.
- [44] **Fay, Andreas and Weiß, Christoph**. *Bewertung des IST-RE-Prozesses bei Allianz*. [Document] 2010.
- [45] —. *Bewertung des IST-RE-Prozesses bei Allianz in agilem Umfeld*. [Document] 2010.
- [46] **Allianz**. *Was ist Agile?* [Presentation] 2008.
- [47] **Hastreiter, Gerhard, Roberts, Simon and Mathis, Christoph**. *Leuchtf Feuer entzünden: Einführung von Enterprise Scrum bei der Allianz Deutschland AG*. 2009.
- [48] **Allianz**. *Product Vision; "A promise for a conversation"*. [Presentation]
- [49] —. *Allianz TOSCA Training*. [[Presentation]]
- [50] **DATEch**. *Leitfaden Usability*. [Document] Frankfurt/Main : Deutsche Akkreditierungsstelle Technik in der TGA GmbH, 2008.
- [51] **Paech, Barbara**. *Requirements Engineering*. [Lecture] Heidelberg : Software Engineering Group - Institute for Computer Science - University of Heidelberg, 2009.

F Anhang

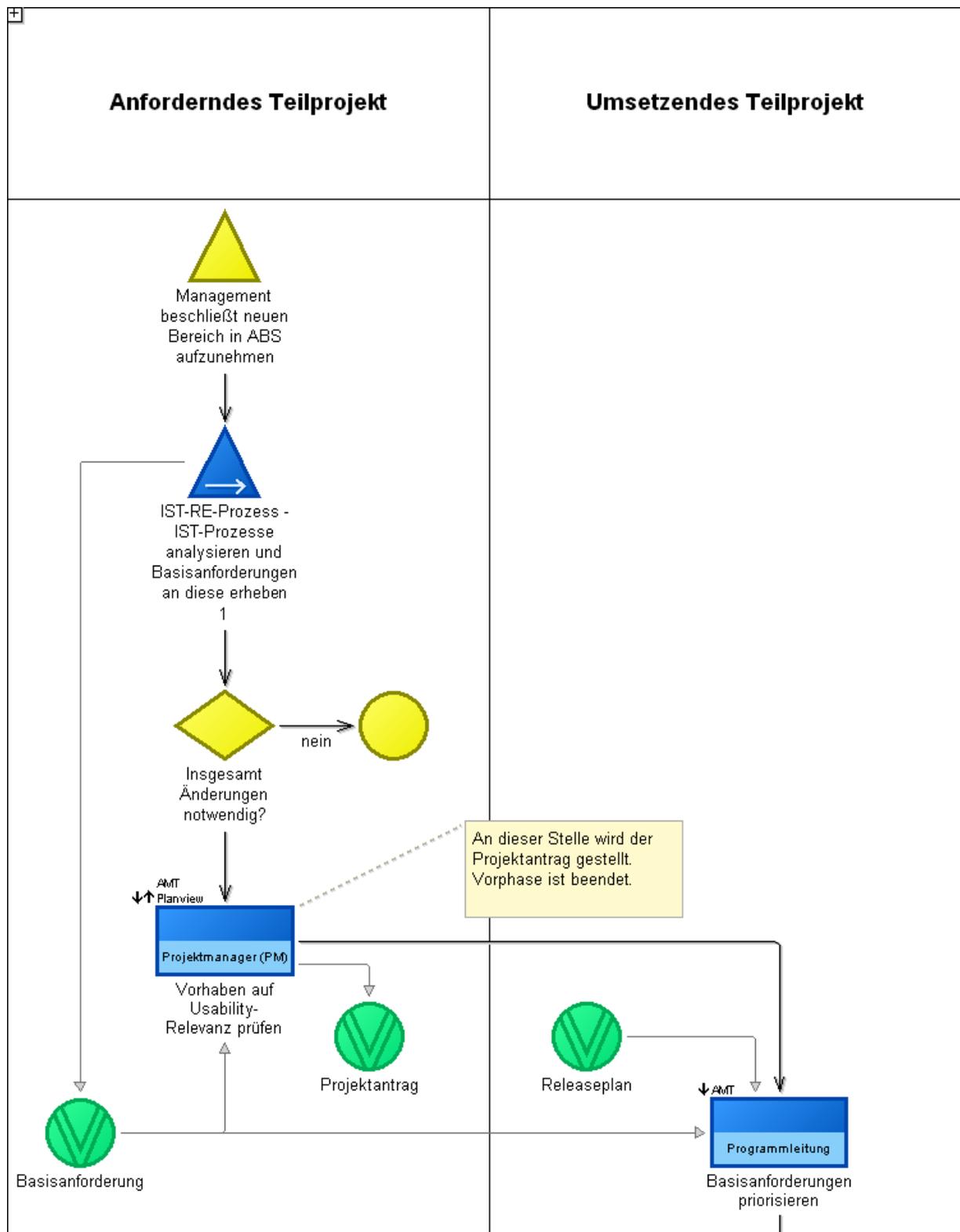
F.1 Adonis-Legende

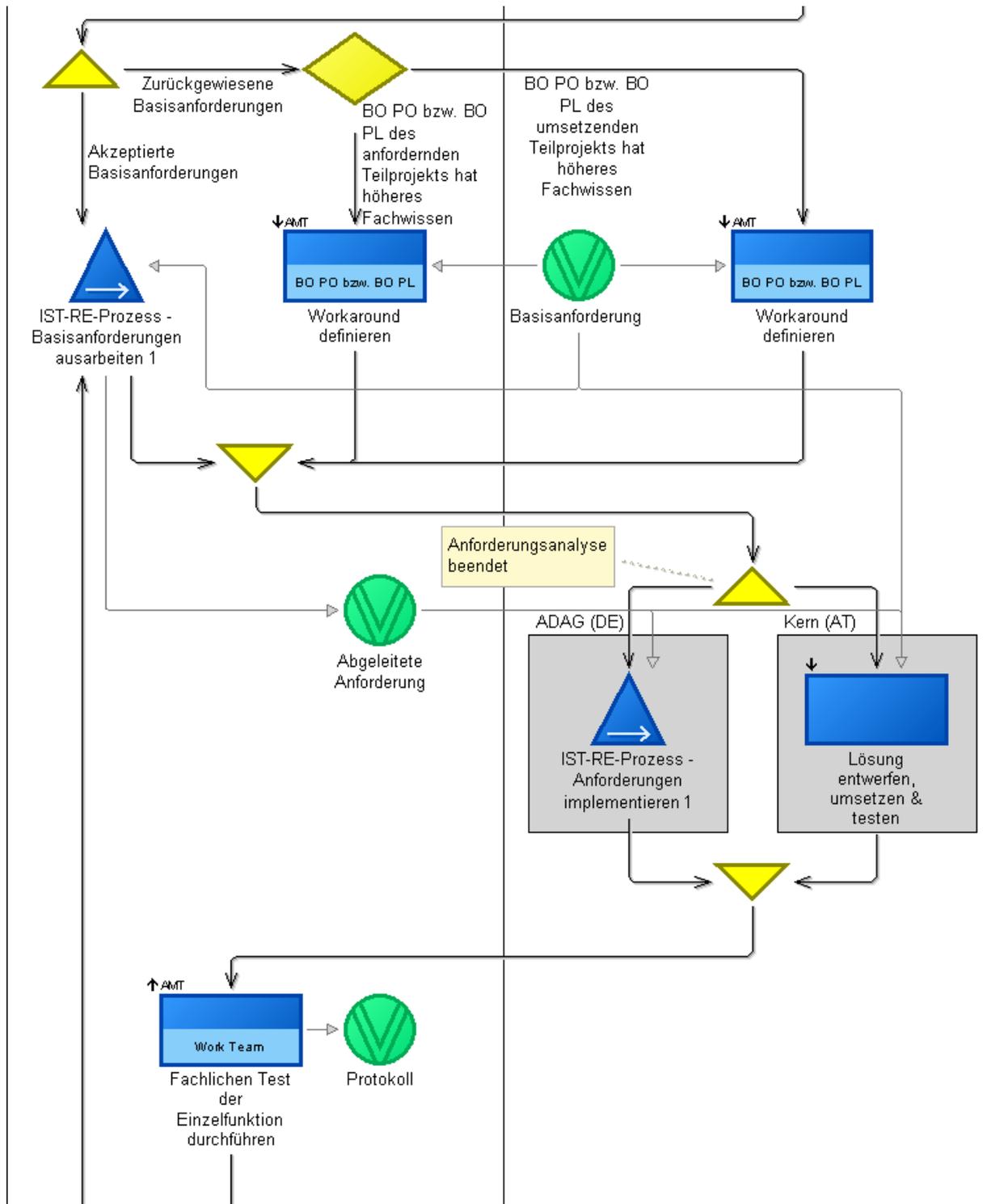
Modellelement	Erklärung
 <p data-bbox="268 667 480 745">IST-Prozesse analysieren</p>	<p data-bbox="584 468 898 499">Aktivität: Arbeitseinheit</p> <p data-bbox="584 504 1418 734">Die Bezeichnung der Aktivität steht unterhalb des Rechtecks („IST-Prozesse analysieren“), der Verantwortliche kann innerhalb stehen („BO PO bzw. BO PL“), beteiligte Systeme können oberhalb stehen („ObjectiF“). Der Pfeil nach unten symbolisiert, dass die Aktivität Dokument(e) als Input nimmt. Analog würde ein nach oben gerichteter Pfeil anzeigen, dass die Aktivität ein Output hätte.</p>
 <p data-bbox="240 958 512 1081">IST-RE-Prozess - Anforderungen überprüfen 2</p>	<p data-bbox="584 891 1418 992">Prozessaufruf: Aufruf eines Unterprozesses. Dessen Name steht unterhalb des Symbols („IST-RE-Prozess - Anforderungen überprüfen 2“).</p>
	<p data-bbox="584 1288 1418 1384">Swimlane: Verdeutlicht eine Rolle. Deren Name steht oben („Umsetzendes Teilprojekt“). Alle Aktivitäten, die in ihr liegen, können dieser Rolle untergeordnet werden.</p>
	<p data-bbox="584 1646 1169 1677">Start-Ereignis: Zeigt den Prozessbeginn an.</p>
	<p data-bbox="584 1843 1134 1874">End-Ereignis: Zeigt das Prozessende an.</p>

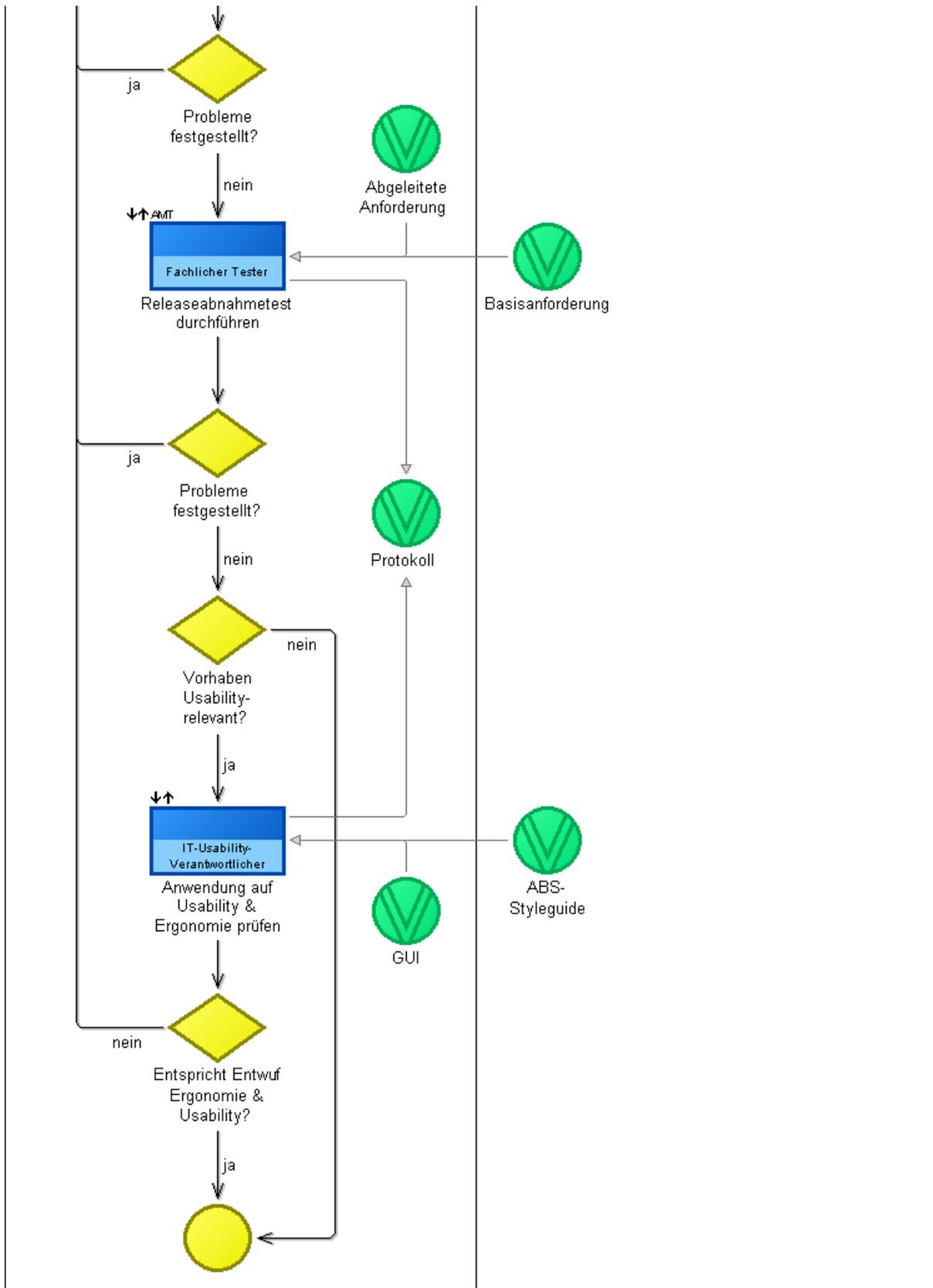
	<p>Sequenz-Fluss: Definiert die Reihenfolge in der Aktivitäten ausgeführt werden.</p>
	<p>Oder-Gatter: Bei dieser Verzweigung wird der Sequenzfluss abhängig von Bedingungen zu genau einer ausgehenden Kante geleitet. Eine Beschreibung der Entscheidung, die hier getroffen wird, kann direkt unterhalb des Gatters stehen.</p>
	<p>Und-Gatter: Bei dieser Verzweigung werden alle ausgehenden Kanten simultan aktiviert.</p>
	<p>Und-Gatter Zusammenführung: Zuvor mit einem einfachen Und-Gatter verzweigte Wege werden mit dem umgedrehten Und-Gatter wieder zusammengeführt. Dabei wird auf alle eingehenden Kanten gewartet, bevor der ausgehende Sequenzfluss aktiviert wird.</p>
	<p>Aggregation: Logische Gruppierung von Aktivitäten.</p>
	<p>Aggregation (neu): Zeigt im SOLL-RE-Prozess neue Elemente gegenüber dem IST-RE-Prozess an.</p>
<p>Anforderungsanalyse beendet</p>	<p>Kommentar: Ein Kommentar zum Prozess wird mit gestrichelter Linie an die betreffende Stelle gebunden oder kann verbindungslos existieren.</p>
 Projektantrag	<p>Dokument: Repräsentiert Informationen, die in Bezug zu Aktivitäten stehen.</p>
	<p>Gerichtete Assoziation: Verbindet Dokumente und Aktivitäten. Der Pfeil zeigt an, ob die Aktivität das Dokument als Eingabe benötigt (Pfeil in Richtung der Aktivität) oder als Ausgabe liefert (Pfeil in Richtung des Dokuments)</p>

F.2 IST-RE-Prozess in Adonis

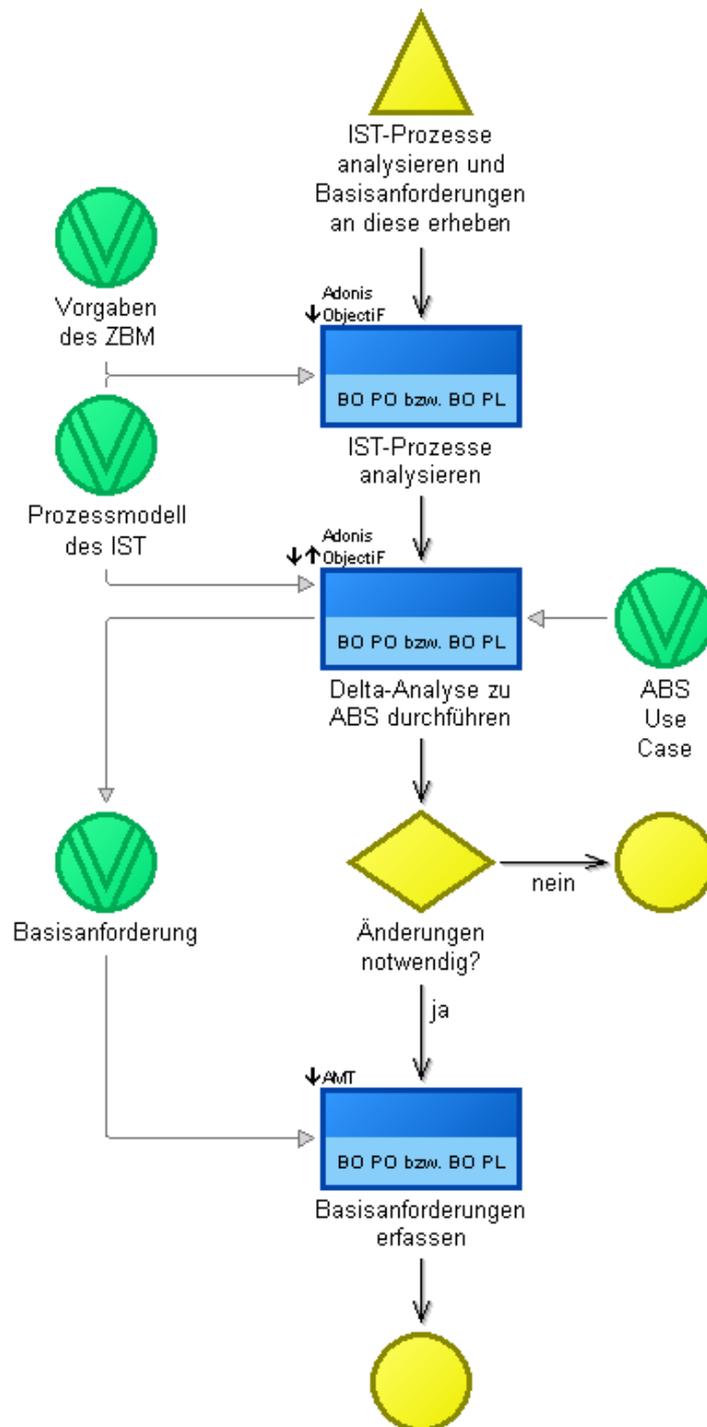
F.2.1 Hauptprozess



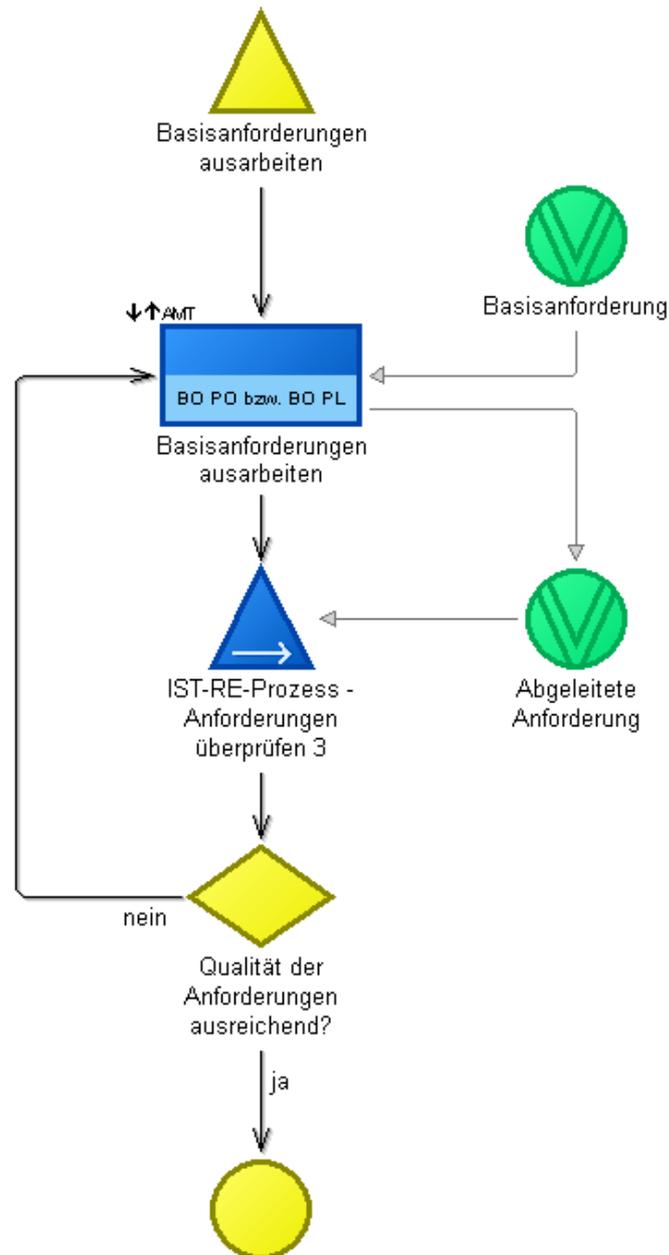




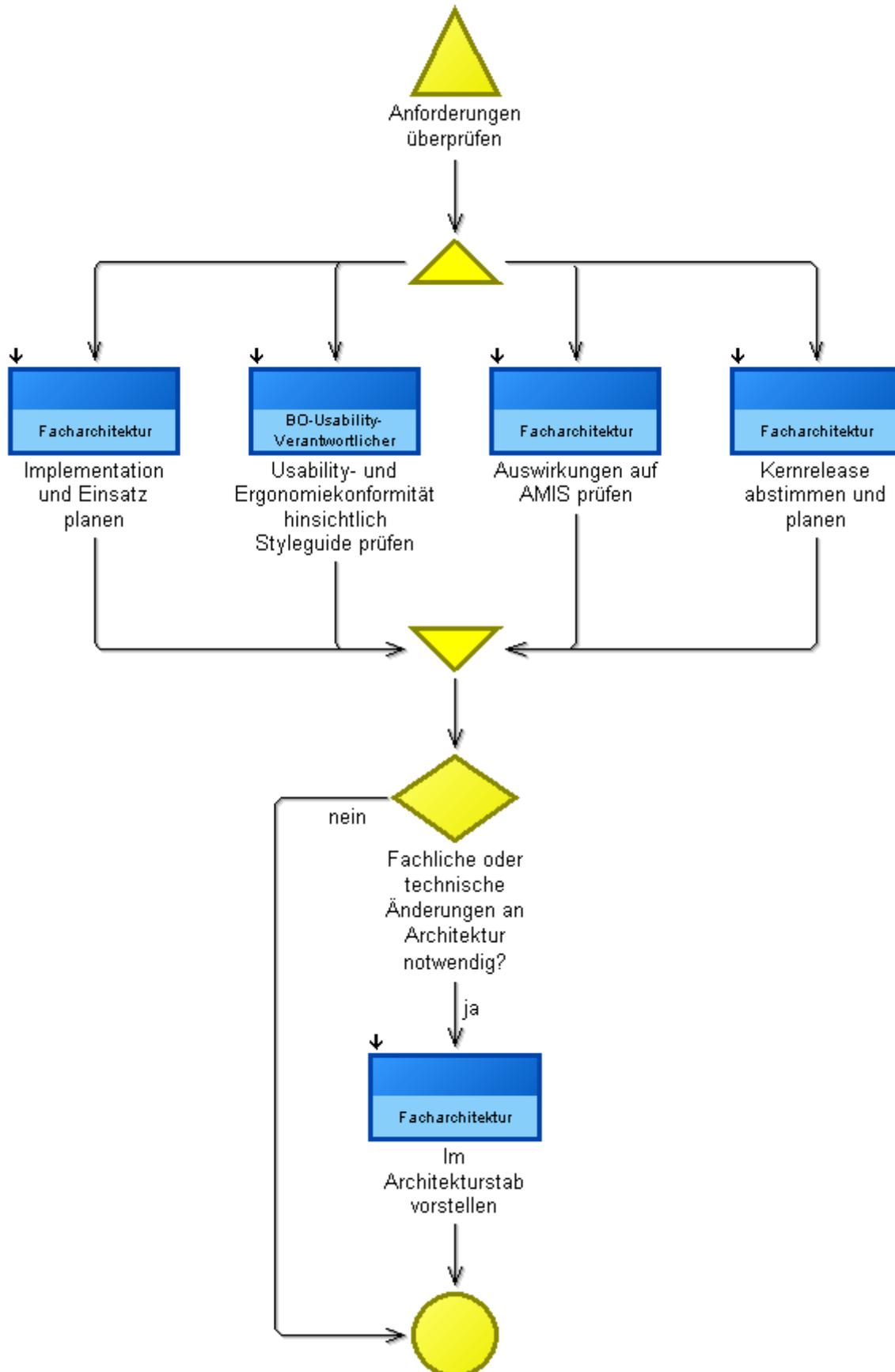
F.2.2 Unterprozess „IST-Prozesse analysieren und Anforderungen an diese erheben“



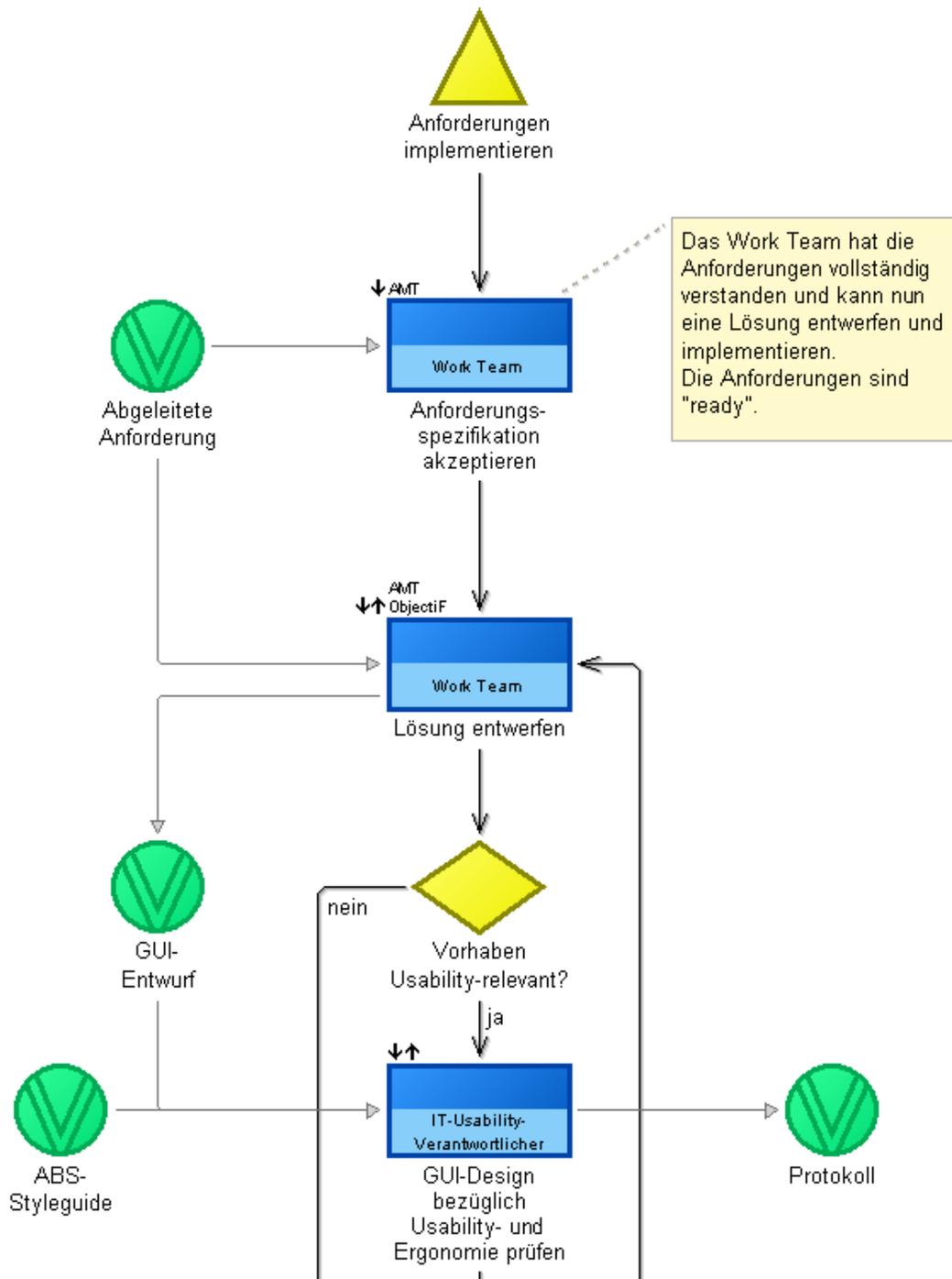
F.2.3 Unterprozess „Basisanforderungen ausarbeiten“

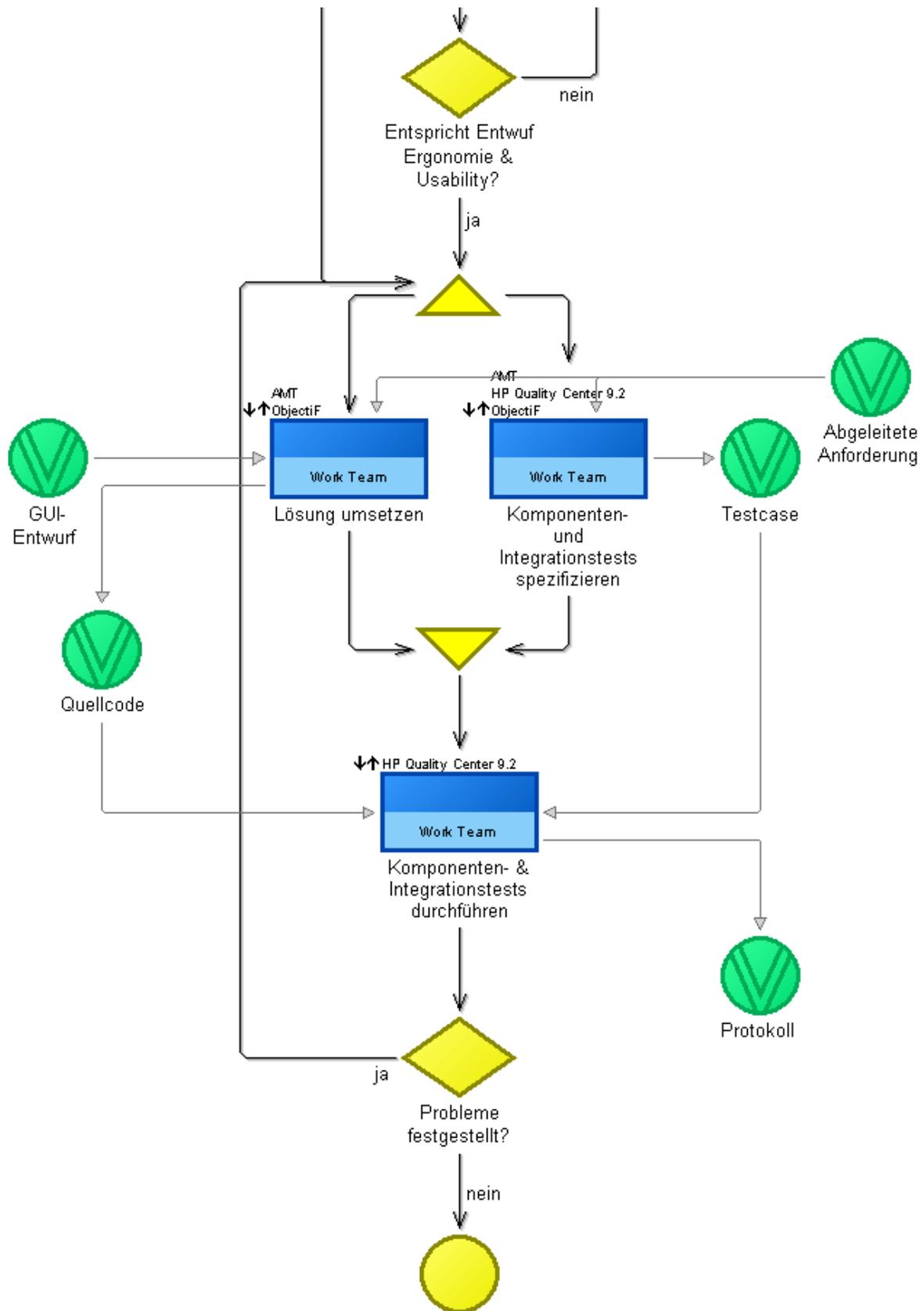


F.2.4 Unterprozess „Anforderungen überprüfen“



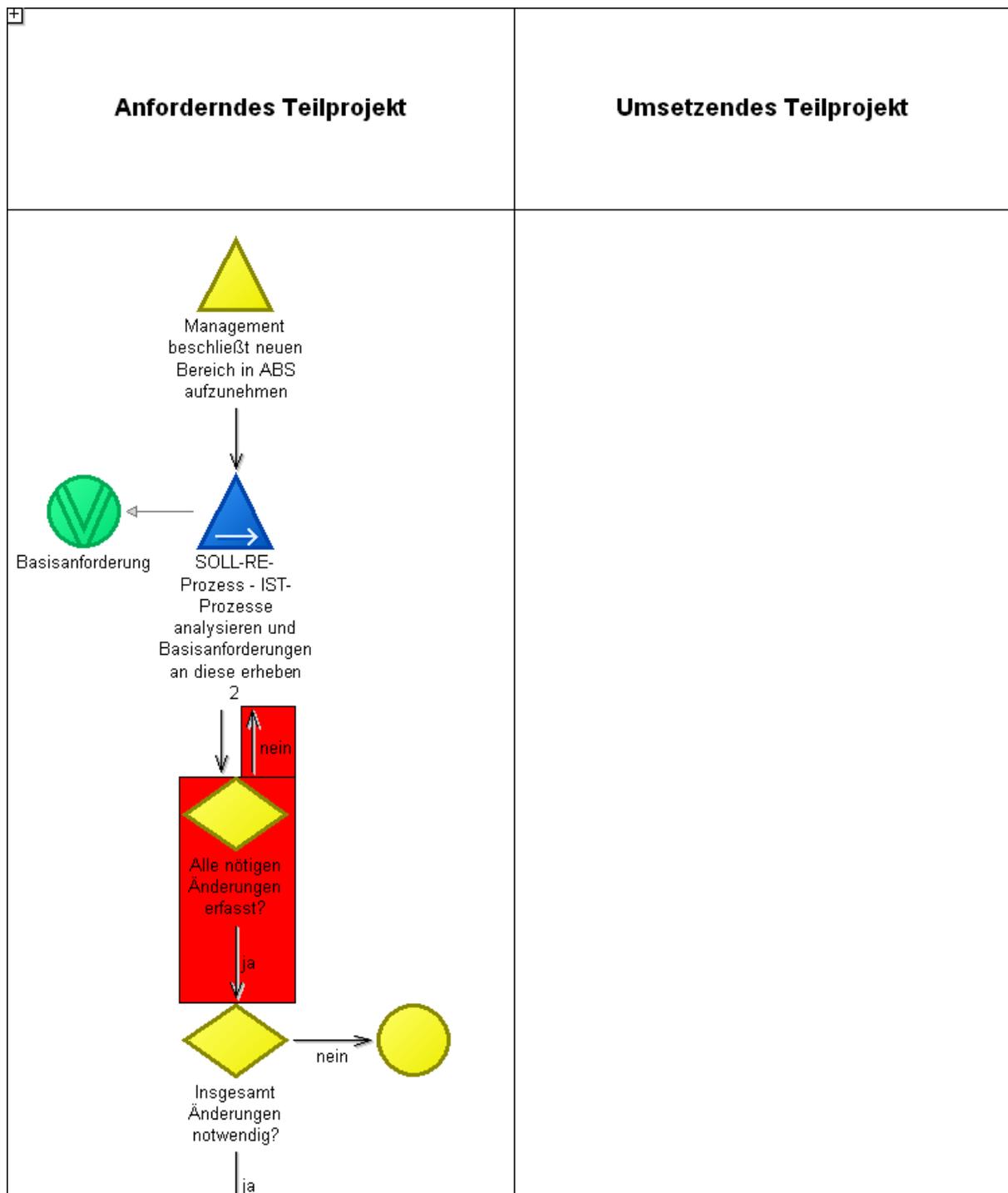
F.2.5 Unterprozess „Anforderungen implementieren“

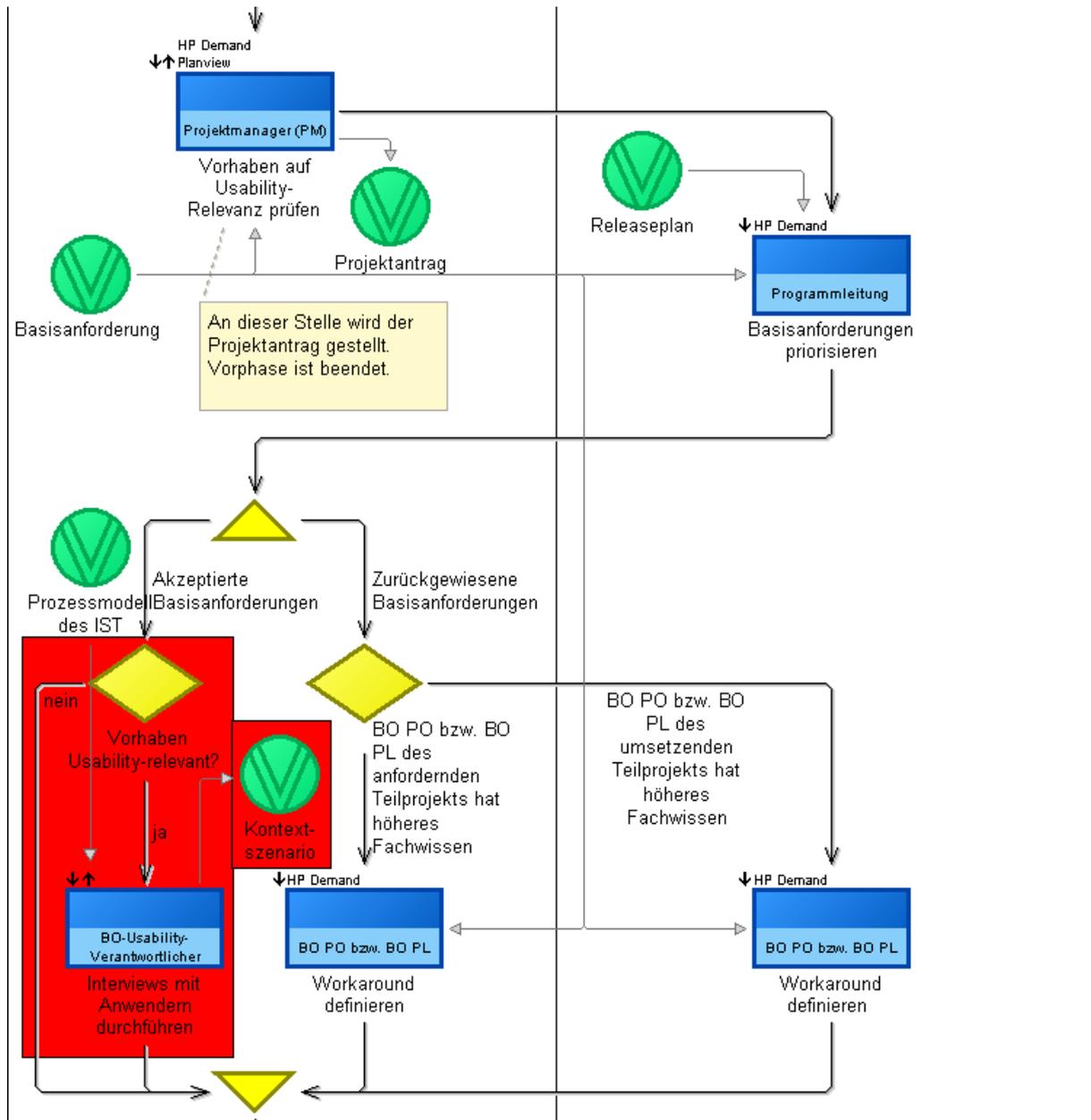


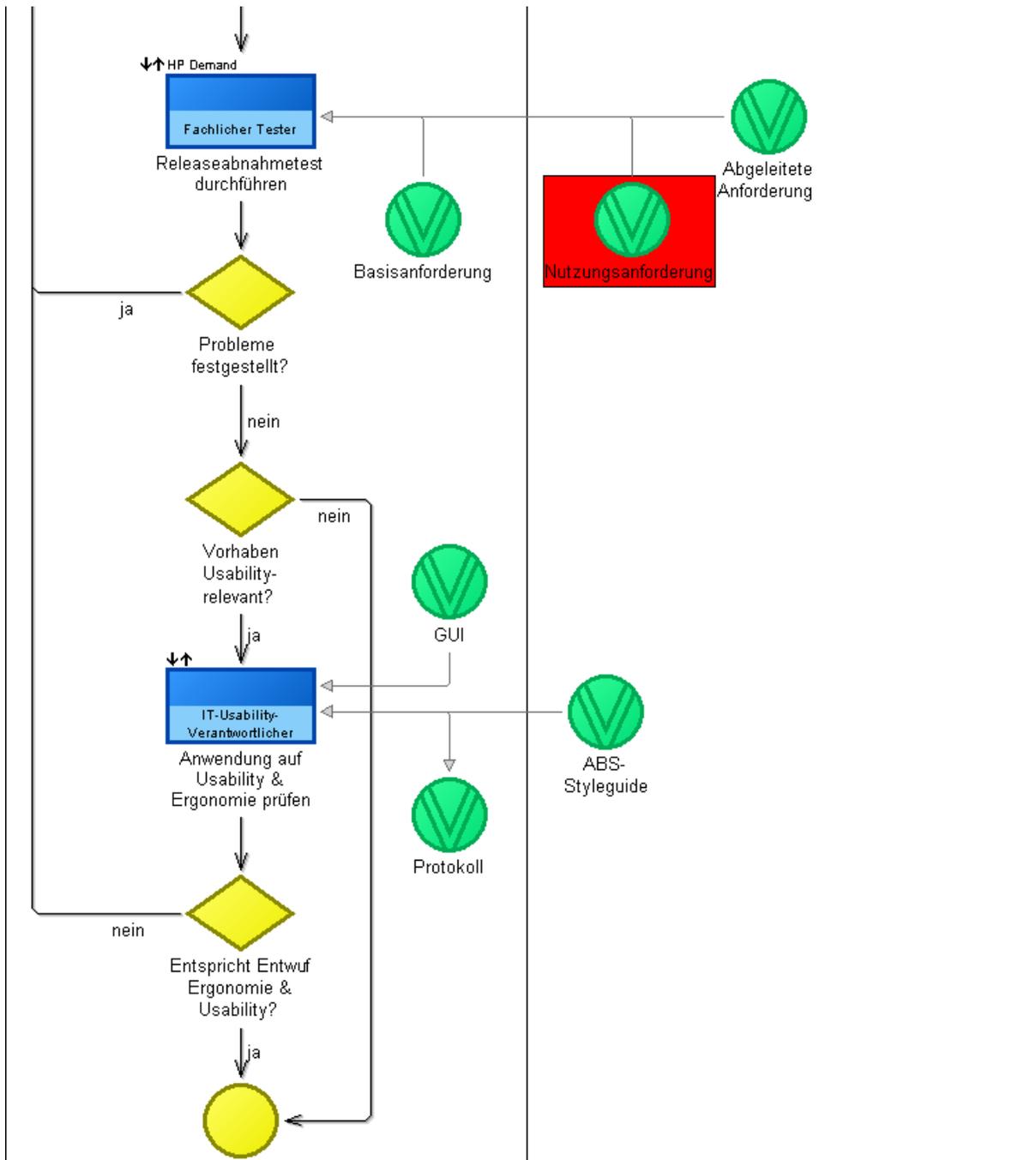


F.3 SOLL-RE-Prozess in Adonis

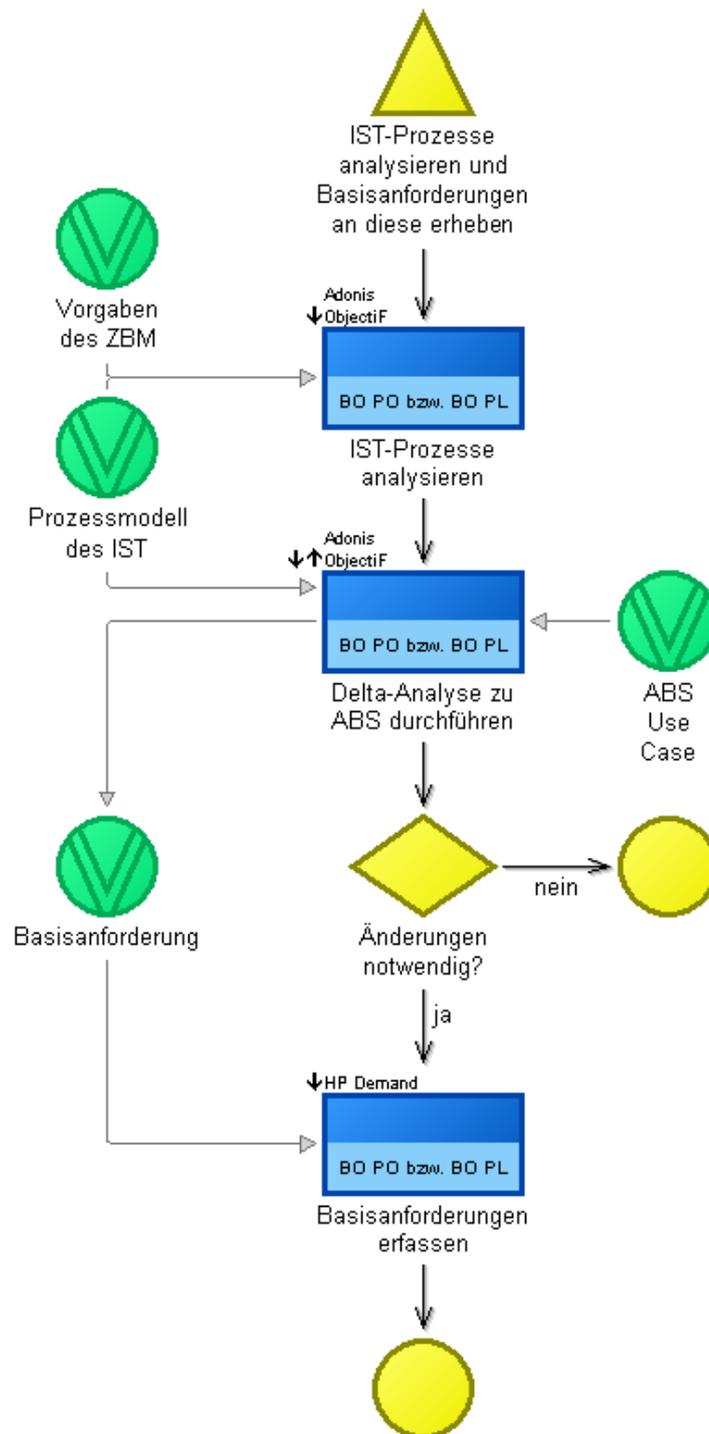
F.3.1 Hauptprozess



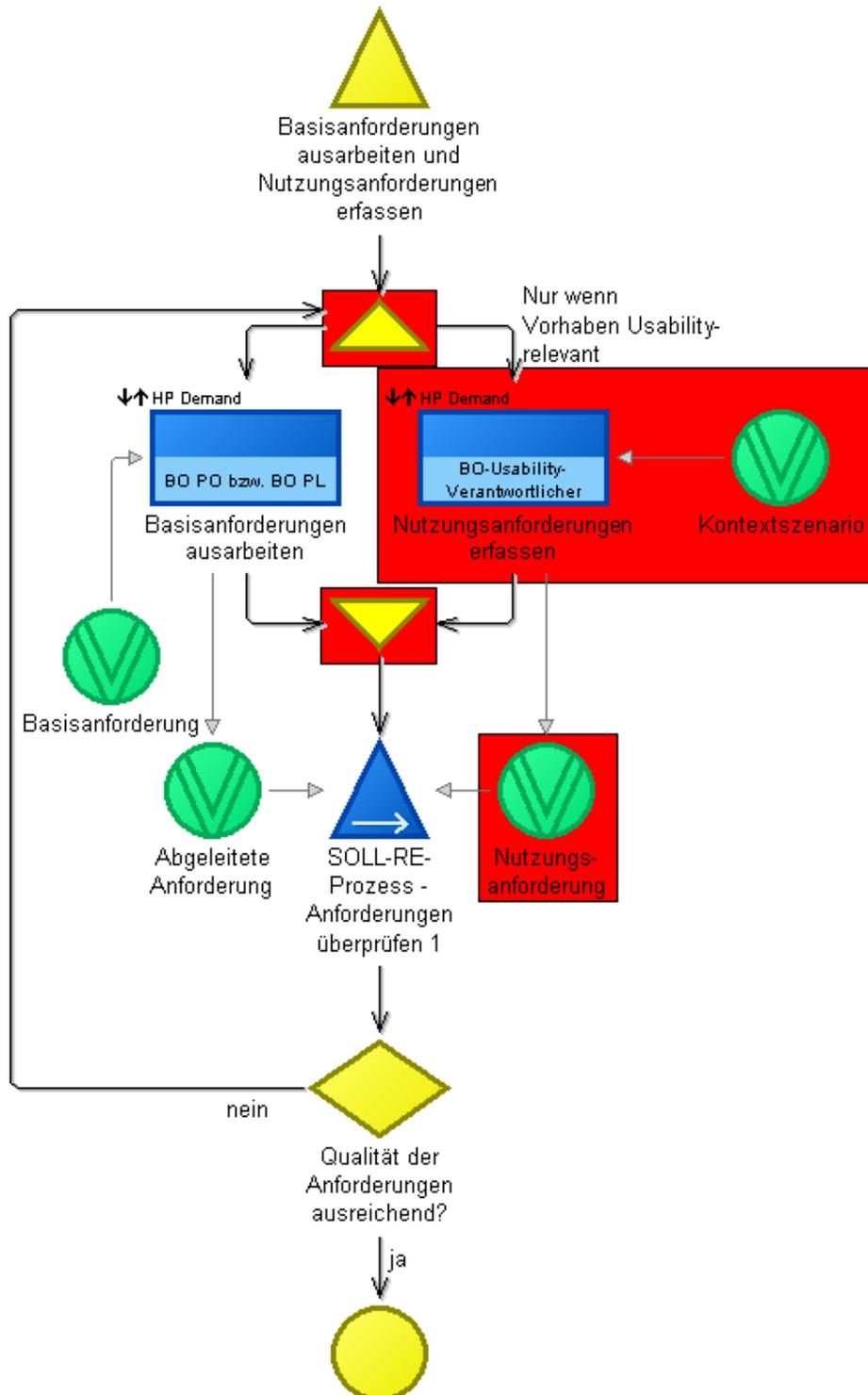




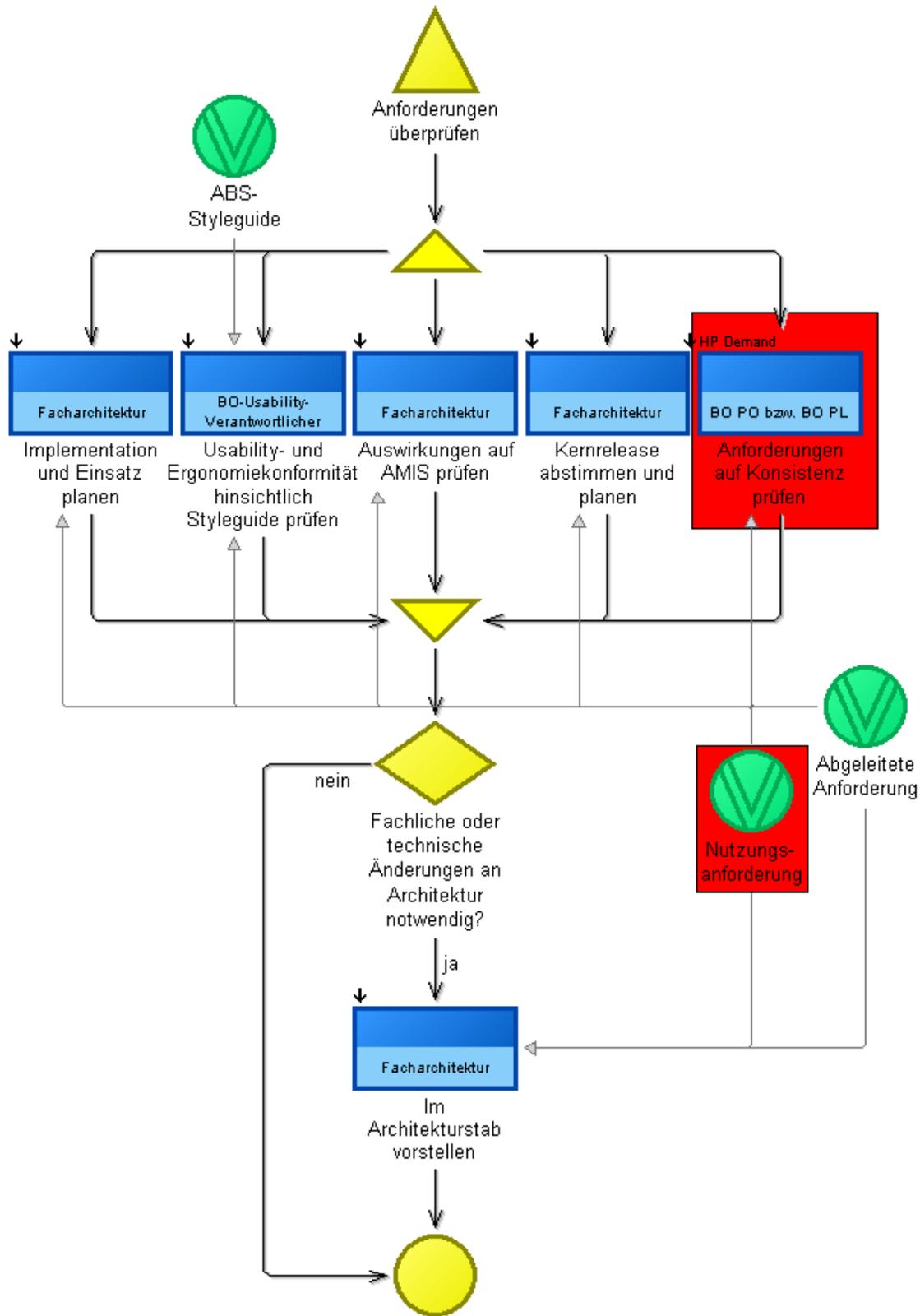
F.3.2 Unterprozess „IST-Prozesse analysieren und Anforderungen an diese erheben“



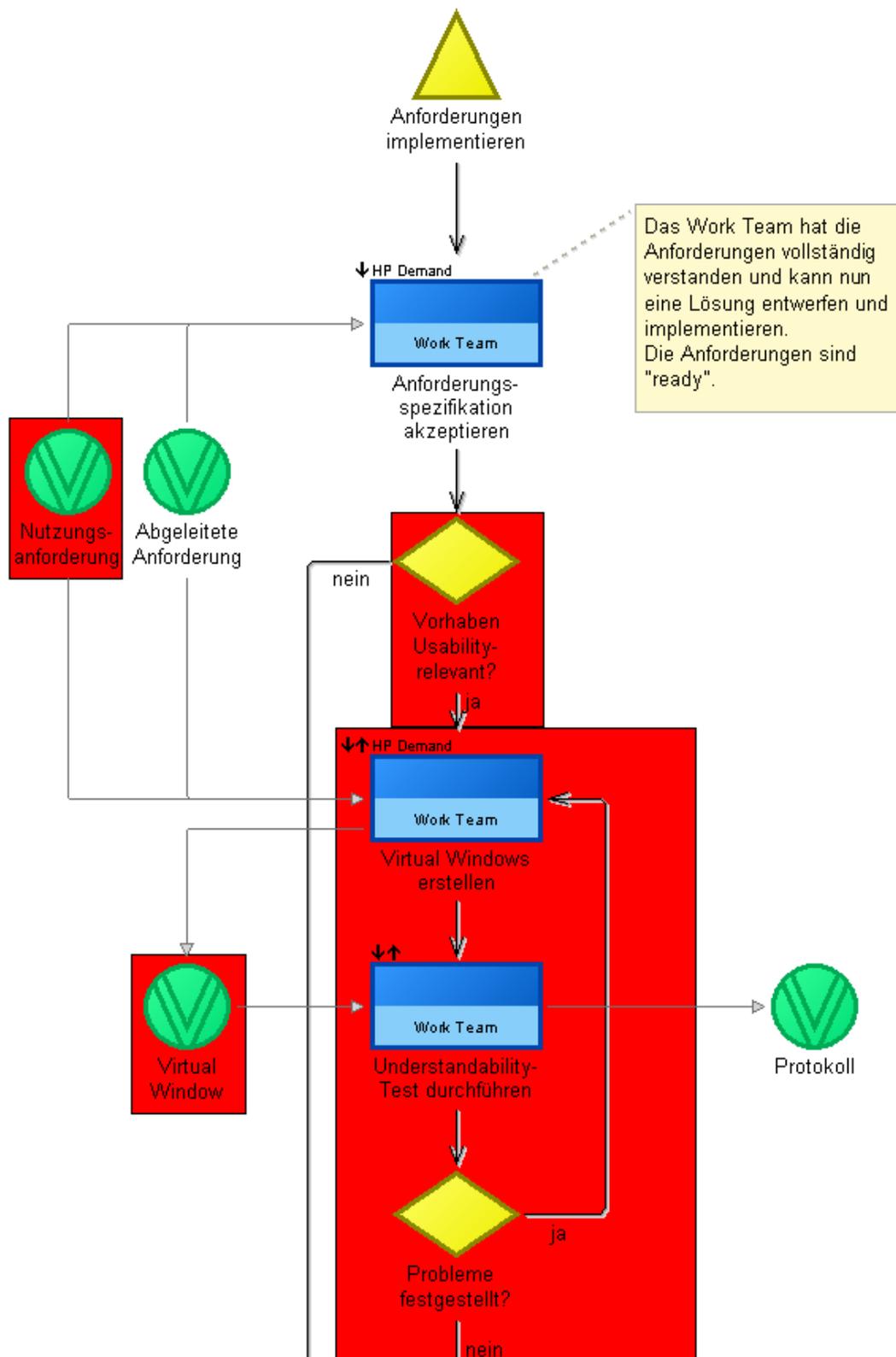
F.3.3 Unterprozess „Basisanforderungen ausarbeiten und Nutzungsanforderungen erfassen“

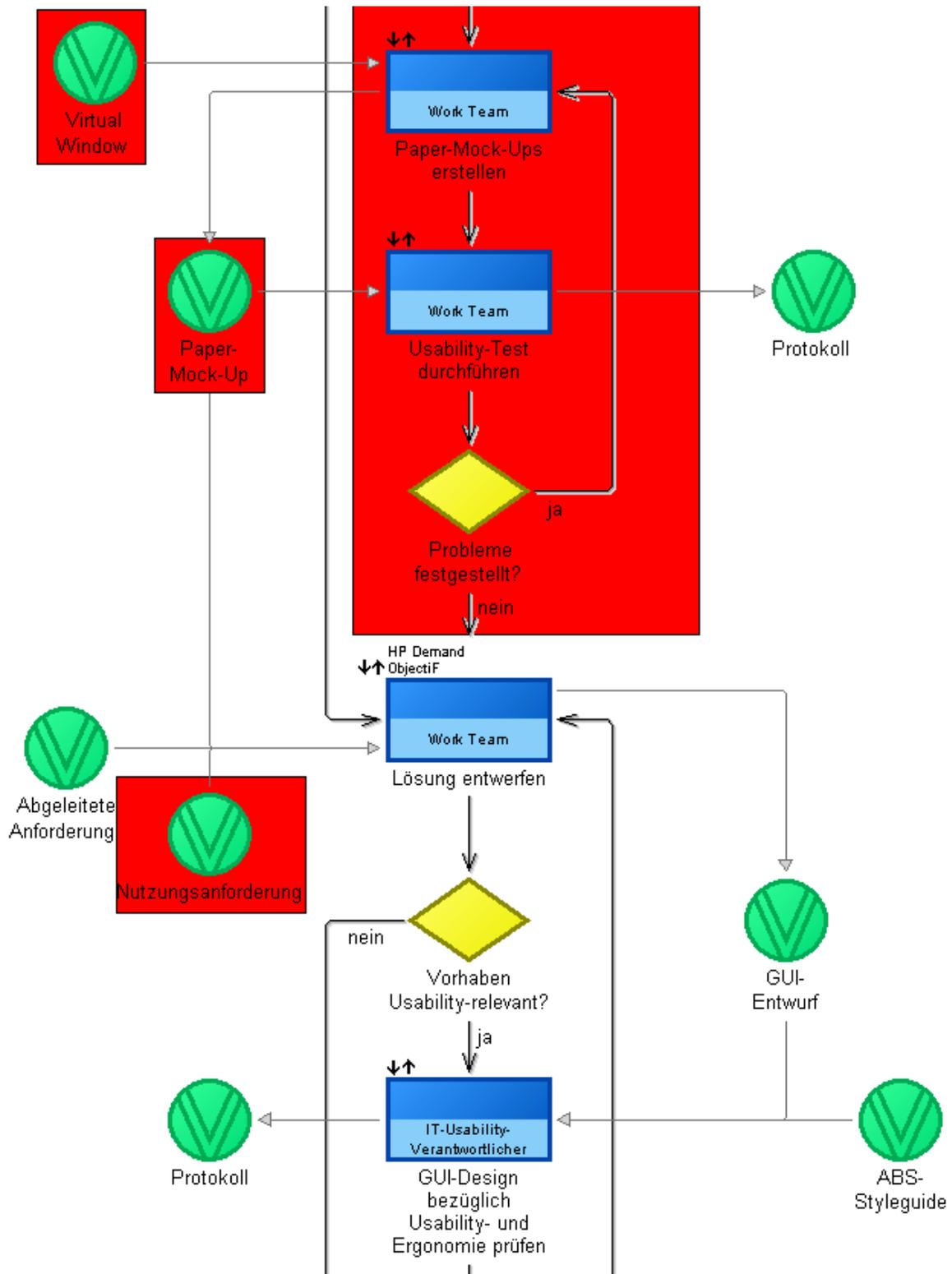


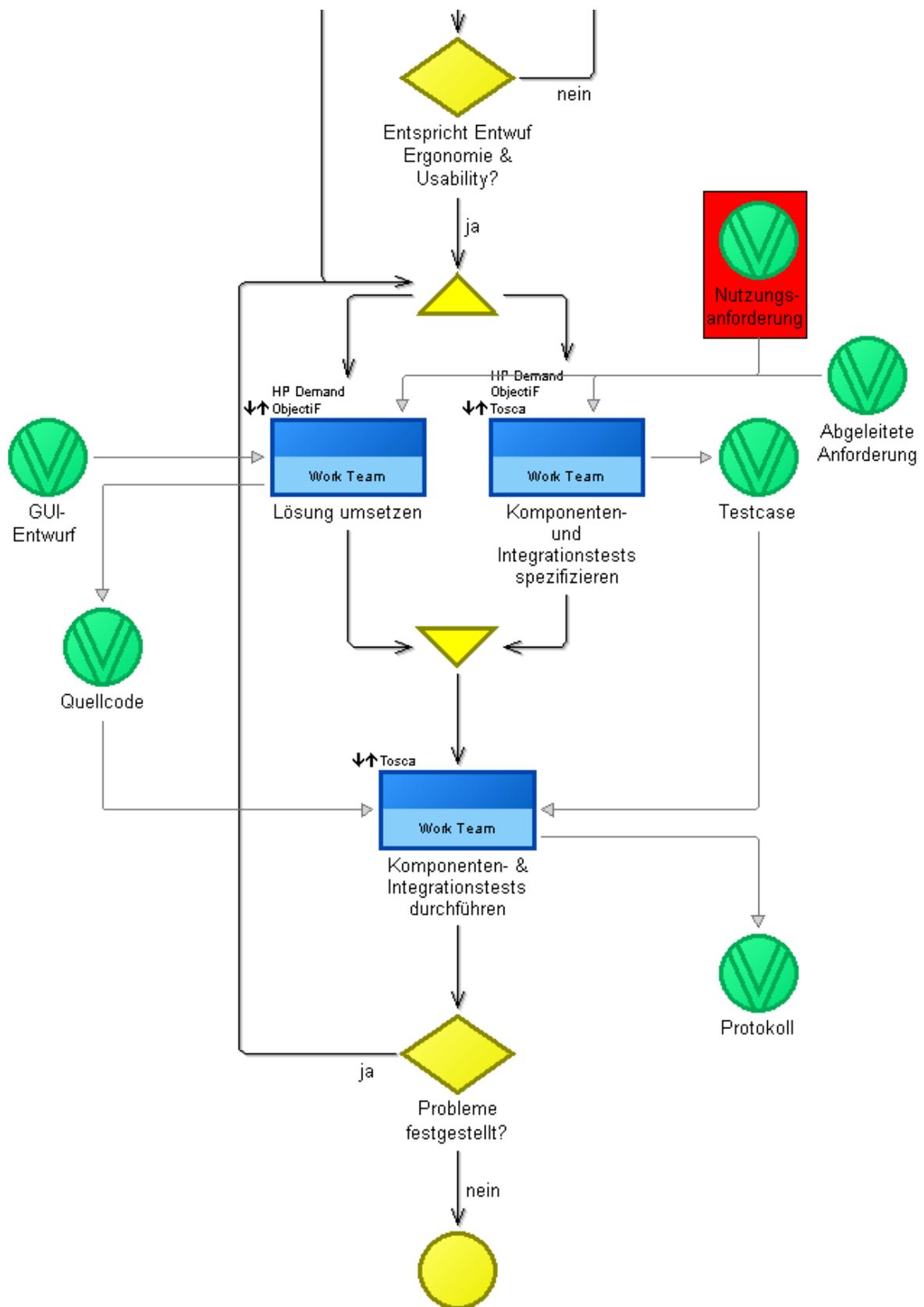
F.3.4 Unterprozess „Anforderungen überprüfen“



F.3.5 Unterprozess „Anforderungen implementieren“







F.4 Fragen & Antworten aus Interviews mit Allianz-Mitarbeiterinnen und -Mitarbeitern

Die nachfolgenden Fragen & Antworten fassen zum Teil fest gegebene und frei formulierte Fragen wie auch Antworten zusammen. Grundlage für die Befragung ist [30] und [31]. Die gewonnenen Erkenntnisse dienen als Grundlage für die Ausführungen in Kapitel 3.

- Wie genau werden die IST-Prozesse bei Allianz analysiert? Wie ergeben sich daraus Anforderungen? Werden dabei die Bestandssysteme hinzugezogen?

Die IST-Prozesse ergeben sich zum einen aus den „Gepros“ (Beschreibungen aller Geschäftsprozesse in der ADAG), die auf Basis der bisherigen Bestandssysteme laufen. Zum anderen aus den Vorgaben des ZBM. Diese IST-Prozesse werden dann mit den Prozessen aus dem ABS-Kern abgeglichen und auf notwendige Anpassungen hin untersucht (Anm.: Die Prozesse liegen in Adonis, die Use Cases in ObjectiF vor). Sind Anpassungen oder Erweiterungen notwendig, ergeben sich daraus Anforderungen.

- Können Sie ein Beispiel für je eine typische Anforderung, die aus dem Prozessschritt der Analyse der IST-Prozesse bei Allianz, „Erfassung der Anforderung“, „Ausarbeitung der Anforderung“ und „Übergabe an Fachl. Architektur“ hervorgeht, geben? (Ziel ist hierbei den Unterschied in der Granularität zu sehen)

Bsp.: Anforderung # 4423 „Erweiterung des VVG-Nebenabreden-Prozesses“. Diese sog. Basis-Anforderung wird im ersten Schritt der Erfassung einer Anforderung ermittelt. Aus der Ausarbeitung und weiteren Analyse ergeben sich die sog. abgeleiteten Anforderungen, wie z.B. „Implementierung von neuen Tabellen für VVG für die VVG Workflowsteuerung“ oder „Neue PEZ-Kategorie und Routingkombinationen für Nebenabredenprozess“. Die Fachl. Architektur überprüft diese Anforderungen u.a. auf die Einhaltung der ABS Architekturrichtlinien.

- Sind die aus dem Prozessschritt „Erfassung der Anforderung“ hervorgehenden Anforderungen Grundlage für den Projektantrag, der in der Vorphase gestellt wird? Wenn nein, welche Anforderungen werden stattdessen als Grundlage genommen? Welchem Prozessschritt entstammen diese?

Wie aus dem Vorgehensmodell (siehe Abb. 9 und [30]) ersichtlich: bereits aus der groben Releaseplanung ergeben sich die Projektanträge, aus diesen Projektanträgen dann die Basisanforderungen (siehe oben).

- Wer genau ist mit Verantwortliche(r) „Anf. TP“ und „Ums. TP“ gemeint?

Die Projektstruktur ist aus unterschiedlichen Teilprojekten (z.B. Vertrag, Schaden, In-/Exkasso, etc.) aufgebaut. Daher gibt es Anforderungen, die z.B. vom Teilprojekt Schaden angefordert werden und

von Vertrag umgesetzt werden, Daher „anf. TP“ = anforderndes Teilprojekt, „ums. TP“ = umsetzendes Teilprojekt.

- Wie genau wird eine Anforderung im Prozessschritt „Ausarbeitung der Anforderung“ ausgearbeitet?

Die Basisanforderung wird in mehrere abgeleitete Anforderungen an unterschiedliche Anwendungskomponenten (z.B. Batch, Datenmodell, etc.) ausgearbeitet.

- Wer trifft die Entscheidung, ob eine Anforderung eine Basis- oder abgeleitete Anforderung ist? Wie trifft er sie?

Die Entscheidung wird vom jeweiligen Teilprojekt getroffen. Eine Basisanforderung beschreibt nur den Funktions-/Themenblock, in der abgeleiteten Anforderung werden die einzelnen Tätigkeiten zur Umsetzung beschrieben. Eine Basisanforderung enthält dementsprechend eine oder mehrere abgeleitete Anforderungen.

- Wie genau bewertet und priorisiert die Programmleitung eine Basisanforderung? Wird dabei u.U. mit Österreich kommuniziert?

Abgleich der Anforderungen mit der groben Releaseplanung (auch Kern-Entwicklung), Rückmeldungen aus den DG'en (Dienstleistungsgebieten) über die Notwendigkeiten von Verbesserungen an ABS. Ansonsten sind die einzelnen Teilprojekte für „ihre“ Basisanforderungen verantwortlich.

- An welcher Stelle im Prozess wird das Vorgehen auf Usability-Relevanz geprüft?

Während des Schrittes „QS der Anforderung“ durch die Facharchitektur wird ein Check zur Usability und Ergonomie durchgeführt (z.B. bei GUI-Anforderungen Überprüfung auf Einhaltung des Style-Guide)

- Wer genau ist mit Verantwortliche(r) „FA“ gemeint?

Facharchitektur.

- Werden IST-Prozesse systematisch erhoben? Also beispielweise erst Geschäftsprozesse, dann ZBM?

Die Prozesse sind den meisten Zuständigen meist schon bekannt und werden ja anhand der Bestandssysteme noch so praktiziert. Eine systematische Erhebung gibt es daher nicht. Die Dokumentation zu den bestehenden Prozessen ist zudem von Sparte zu Sparte sehr unterschiedlich.

- Wie genau werden die IST-Prozesse in der Delta-Analyse abgeglichen mit ABS?

Auch hier geht das direkt über das Wissen der Zuständigen. Es kann auch die Dokumentation des ABS Kerns herangezogen werden.

- Welcher Art sind die aus der Delta-Analyse resultierenden Anforderungen?

Das sind Basisanforderungen.

- Wird für den Projektantrag geprüft, ob das Vorhaben Usability-relevant ist?

Nein. Die Prüfung erfolgt bei der Ausarbeitung der einzelnen Basisanforderungen.

- Kann der Projektantrag abgeleitete Anforderungen enthalten oder nur Basisanforderungen?

Der Projektantrag stützt sich auf grobe Basisanforderungen.

- In wie fern ist zwischen der Ausarbeitung und der Umsetzung eine Pause (da hier ein Wechsel zwischen den Verantwortlichen stattfindet, also von „Anf. TP“ zu „Ums. TP“)?

Dazwischen ist meist keine Pause. Im Übrigen sind das anfordernde und umsetzende Teilprojekt zu 80% dieselben Teilprojekte.

- Wer aus den Dienstleistungsgebieten meldet sich bezüglich der Priorisierung von Basisanforderungen zurück?

Letztlich kommen die Anforderungen bzw. groben Wünsche bereits priorisiert aus den Dienstleistungsgebieten rein. Sollte das nicht so sein, so wird die Anforderung im „Anf. TP“ priorisiert.

- Werden abgeleitete Anforderungen priorisiert?

Nein.

- Wer definiert einen Workaround falls eine Basisanforderung abgelehnt wird?

Je nach dem Fachwissen das anfordernde oder das umsetzende Teilprojekt.

- Welcher Abteilung entstammen „Anf. TP“ und „Ums. TP“?

Aus BO.

- Wie muss eine Anforderung aussehen, damit Sie wirklich die Entscheidung für die Einbindung von Anwendern fällen?

Die Bedingungen einen Anwender zu einer Usability-Prüfung einzubinden, können sehr unterschiedlich sein:

So kam es vor, dass in einer Tabelle, in der jede zweite Zeile zur besseren Lesbarkeit einen hellgrauen Hintergrund hat, dieser Grauwert bei neuen TFT-Bildschirmtypen so blass war, dass er sich zu

wenig von dem anderen, weißen abgehoben hat. Wir haben zwar ein Tool zur Bestimmung von Helligkeitsdifferenzen, aber hier war die subjektive Meinung von Anwendern zur Darstellung auf unterschiedlicher Hardware gefragt. Es musste ein Grauwert gefunden werden, der auf den neuen TFT-Bildschirmen kräftig genug war, aber auf den alten nicht zu dominant.

Wird ein bestehender Prozess nur geringfügig, z.B. um einige, wenige Attribute mit bekannten Steuerelementen erweitert, so darf man davon ausgehen, dass der Benutzer damit keine Handhabungsprobleme hat und somit eine Usability-Überprüfung mit Anwendern keinen Mehrwert erbringt. Kommt dagegen ein ganz neuer Prozess hinzu, dann macht das schon Sinn mit Anwendern etwas zu testen.

Wenn aus den prozessorientierten Spartensystemen bestehende Prozesse ins objektorientierte ABS übernommen werden, kann die Integration nicht 1:1, sondern muss konsequenterweise im Stil von ABS objektorientiert erfolgen. Mittels Anwenderschulungen und Leitfäden wird hier der Übergang erleichtert. Erst wenn nach einer ausreichenden Einarbeitungszeit der Benutzer sich an die neue Arbeitsweise gewöhnt hat, sollte beim Auftreten von Handhabungsproblemen eine Usability-Prüfung erfolgen.

F.5 Kriterienkatalog

Kriterium	Betrifft ...	Bewertung durch ...
Einhaltung der firmeninternen Richtlinien	Gebrauchstauglichkeit	Aktivität zur Prüfung im RE-Prozess vorgesehen?
Einhaltung der gesetzlichen Richtlinien	Gebrauchstauglichkeit	Aktivität zur Prüfung im RE-Prozess vorgesehen?
Zufriedenheit der Kundin bzw. dem Kunden	Gebrauchstauglichkeit RE + agile Methoden	Subjektive Bewertung der Kundin bzw. des Kunden mittels Fragebogen
Zufriedenheit der Benutzerinnen und Benutzer	Gebrauchstauglichkeit RE + agile Methoden	Subjektive Bewertung der Benutzerinnen und Benutzer mittels Fragebogen
Einbindung des Product Owners & der Benutzerinnen und Benutzer	RE + agile Methoden	ScrumButt Test (Question 4 – Product Owner)
		% aller relevanten Stakeholder eingebunden; subjektive Bewertung der jeweiligen Stakeholder
Anpassbarkeit der Anforderungen	RE + agile Methoden	Aktivität zur Anpassung von Anforderungen vorgesehen?
Qualität der Anforderungsdokumentation	RE + agile Methoden	ScrumButt Test (Question 3 – Agile Specification)
Konsistenz der Anforderungen	RE + agile Methoden	Automatische Syntaxprüfung durch verwendete Tools möglich und in RE-Prozess integriert?
		Aktivität zum Review der Anforderungen durch Product Owner im RE-Prozess vorgesehen?
Berücksichtigung von Abhängigkeiten zwischen Anforderungen	RE + agile Methoden	Aktivität zur Prüfung von Abhängigkeiten zwischen Anforderungen für gesamtes Projekt vorhanden?
		... jeden Sprint vorhanden?
Systematische Erfassung nicht-funktionaler Anforderungen	RE + agile Methoden	Werden nicht-funktionale Anforderungen systematisch erfasst und dokumentiert?
Abdeckung des CMMI-Ziels „Develop Customer Requirements“	CMMI	Wird Practice „Elicit Needs“ ausgeführt?
		Wird Practice „Develop the Customer Requirements“ ausgeführt?

Abdeckung des CMMI-Ziels „Develop Product Requirements“	CMMI	Wird Practice „Establish Product and Product Component Requirements“ ausgeführt?
		Wird Practice „Allocate Product Component Requirements“ ausgeführt?
		Wird Practice „Identify Interface Requirements“ ausgeführt?
Abdeckung des CMMI-Ziels „Analyze and Validate Requirements“	CMMI	Wird Practice „Establish Operational Concepts and Scenarios“ ausgeführt?
		Wird Practice „Establish a Definition of Required Functionality“ ausgeführt?
		Wird Practice „Analyze Requirements“ ausgeführt?
		Wird Practice „Analyze Requirements to Achieve Balance“ ausgeführt?
		Wird Practice „Validate Requirements“ ausgeführt?
Abdeckung des CMMI-Ziels „Manage Requirements“	CMMI	Wird Practice „Obtain an Understanding of Requirements“ ausgeführt?
		Wird Practice „Obtain Commitment to Requirements“ ausgeführt?
		Wird Practice „Manage Requirements Changes“ ausgeführt?
		Wird Practice „Maintain Bidirectional Traceability of Requirements“ ausgeführt?
		Wird Practice „Identify Inconsistencies Between Project Work and Requirements“ ausgeführt?

G Ehrenwörtliche Erklärung

Hiermit erkläre ich, Andreas Fay, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und erlaubten Hilfsmittel verwendet habe.

Heidelberg, den 30. Juni 2010