

Systematische Integration des Requirements Engineerings in den Softwareentwicklungsprozess

Daniela Bommarito^{1,2}

Abstract: Klarheit über die umzusetzenden Anforderungen ist eine zentrale Grundvoraussetzung für ein erfolgreiches Softwareentwicklungsprojekt. Entsprechend ist gutes Requirements Engineering (RE) eine Schlüsseldisziplin im Softwareentwicklungsprozess. Die Projektpraxis zeigt jedoch, dass sich die adäquate Erfassung von Anforderungen häufig als schwierig gestaltet. Insbesondere bei Projekten, die in Ansätzen, aber nicht vollständig nach agilen Grundsätzen durchgeführt werden, bringt das RE diverse Herausforderungen mit sich. Typische Fallstricke sind u.a die in der Praxis oft unzureichende Konsolidierung der kundenseitigen Anforderungen durch den Product Owner oder die interdisziplinären Kompetenzen, über die agil arbeitende Entwicklerteams verfügen müssen. Um die Qualität der Anforderungsdefinition zu verbessern ist es notwendig, die einzelnen RE-Aktivitäten so in den Entwicklungsprozess zu integrieren, dass sie den vorhandenen Rahmenbedingungen des Projektes Rechnung tragen. Dazu zählen beispielsweise die Komplexität des Vorhabens, die Stabilität der Anforderungen oder die verfügbaren Kompetenzen der beteiligten Personen. In dieser Arbeit wird eine systematische Vorgehensweise definiert, um passend zu projektspezifischen Rahmenbedingungen einen geeigneten RE- und Entwicklungsprozess auszuwählen und diesen pragmatisch so anzupassen, dass er den realen Gegebenheiten des jeweiligen Projektes gerecht wird.

Keywords: Requirements Engineering; Softwareentwicklungsprozess; agiles Vorgehen; Tailoring

1 Motivation

“Nur wer sagt, was er möchte, bekommt, was er will” proklamiert sinngemäß der Unternehmensberater und Kommunikationstrainer George Walther. Übertragen auf die Softwareentwicklung bedeutet das, dass ein Softwareentwicklungsprojekt überhaupt nur dann erfolgreich sein kann, wenn geklärt ist, welche Anforderungen die zu erstellende Software eigentlich erfüllen soll. Dabei müssen nicht zwangsweise alle Anforderungen vollständig verstanden sein, bevor überhaupt mit der Implementierung begonnen wird. Aber für dasjenige Inkrement, an dem gerade entwickelt wird, sollte klar sein, was es leisten soll. Nur dann steht fest, was im aktuellen Entwicklungsschritt eigentlich umzusetzen ist.

¹ Hochschule München, Fakultät für Informatik und Mathematik, Lothstraße 64, 80335 München, bommarit@hm.edu

² ARS Computer und Consulting GmbH, Ridlerstraße 37, 80339 München, daniela.bommarito@ars.de

Klarheit über die umzusetzenden Anforderungen ist also ein wesentlicher Erfolgsfaktor in der Softwareentwicklung. Entsprechend ist gutes RE eine Schlüsseldisziplin im Softwareentwicklungsprozess.

Analysen der Projektpraxis vergangener Jahre belegen, dass adäquates RE nach wie vor ein schwieriges Unterfangen ist. Dabei ist jedoch zu beobachten, dass sich die Ursachen der Schwierigkeiten über die Jahre verschieben. Beispielsweise befand der Chaos Report³ von 2003 der Standish Group, dass von den untersuchten Projekten weltweit im Durchschnitt 45% der Features der entwickelten Software nach deren Fertigstellung *nie* verwendet wurde. Gleichzeitig wurden viele der tatsächlich benötigten Funktionalitäten gar nicht oder nur eingeschränkt umgesetzt. Der Bericht führte diese Situation darauf zurück, dass “klassische”, d.h. stark dokumenten-zentrierte Entwicklungsprozesse zu reaktionsträge sind, um den Erfordernissen von sich ändernden Anforderungen während der Entwicklungszeit gerecht zu werden [SGI02].

Abhilfe schaffen sollte das Paradigma der agilen Softwareentwicklung, das maßgeblich geprägt ist durch mehr persönliche Interaktion und weniger formale Dokumentation, eine intensive Einbindung der Kunden in das Entwicklungsgeschehen, kurze Zyklen, kontinuierliches Feedback und eine generelle Offenheit für Veränderungen [Be01]. Der Chaos Report von 2015 belegt, dass die agil durchgeführten Projekte im Mittel erfolgreicher abschneiden als nach dem Wasserfallmodell organisierte Projekte [HW15].

Agile Entwicklungsprozesse verbessern also die Situation, stoßen in der Praxis aber immer wieder an ihre Grenzen. Die Erfahrung zeigt beispielsweise, dass insbesondere kleinere Projekte durch einen agilen Entwicklungsprozess vergleichsweise zuverlässig zum Erfolg geführt werden (58%), während bei großen Projekten lediglich 18% der agil durchgeführten Projekte erfolgreich waren [HW15].

In der Praxis gestaltet sich die lehrbuchmäßige Anwendung von agilen Entwicklungsprozessen dabei oft als schwierig. Die Gründe dafür sind vielfältig und hängen von verschiedenen Einflussfaktoren ab. Beispielsweise ist die direkte Kommunikation nicht möglich, wenn die Projektbeteiligten räumlich verteilt arbeiten.

Die Umstellung von einem klassischen Vorgehensmodell auf ein agiles Paradigma stellt zudem neue Anforderungen an die Beteiligten. Während im klassischen Vorgehen meist spezialisierte Teams mit definierten Hierarchien und Verantwortungsbereichen agieren, setzen agile Vorgehen wie Scrum [Sc95b] auf funktionsübergreifende Teams, die selbstorganisiert zusammen arbeiten. Die dafür erforderlichen interdisziplinären Kompetenzen müssen jedoch erst neu aufgebaut werden. Bei Scrum benötigen Product Owner bspw. Domänen- und Requirementskenntnisse, Entwickler benötigen Entwicklungs- und grundlegende Domänenkenntnisse und beide benötigen umfassende Kommunikationsfähigkeiten.

³ Die wissenschaftliche Präzision des Chaos Reports ist umstritten. Es geht hier also weniger um die genauen Werte als um die Größenordnung der Aussage.

Sind diese Voraussetzungen nicht erfüllt und können diese zur Projektlaufzeit nicht rechtzeitig hergestellt werden, müssen geeignete systematische Maßnahmen getroffen werden, um das Projekt dennoch zuverlässig zum Erfolg zu führen.

2 Zielsetzung

Diese Arbeit beschreibt eine systematische Vorgehensweise, die ausgehend von Problemmerkmalen (d. h. den Eigenschaften der Aufgabenstellung an sich) und organisatorischen Projektrahmenbedingungen einen darauf abgestimmten Softwareentwicklungsprozess konzipiert. Dabei wird sichergestellt, dass bei instabilen Anforderungen oder stark innovativen Aufgabenbereichen kurze Feedback-Zyklen zu einer schnellen Validierung der Projektergebnisse führen. Gleichzeitig werden die RE-Aktivitäten in diesem Prozess so gestaltet, dass Anforderungen auf eine Art und Weise erhoben werden, die zu den organisatorischen Rahmenbedingungen passt.

Hierzu wird anhand der Problemmerkmale ermittelt, welches Vorgehensmodell sich am besten für die Durchführung dieses Projektes eignet. Die aus der Literatur bekannten Vorgehensmodelle werden danach pragmatisch auf die realen Gegebenheiten angepasst. Diese systematische Vorgehensweise hilft dem Anwender, die geeigneten Fragen zu stellen, um frühzeitig Risiken zu erkennen und somit den Entwicklungsprozess passend zum Problem und den projektspezifischen, organisatorischen Rahmenbedingungen aufzusetzen.

3 Stand von Wissenschaft und Praxis

Die Herausforderungen, die sich beim Anpassen des Softwareentwicklungsprozesses ergeben, werden in diversen Studien thematisiert. Im Fokus liegen dabei die Ermittlung der relevanten Einflussfaktoren sowie das systematische Vorgehen beim Tailoring.

Boehm und Turner entwickelten eine Vorgehensweise, die als Grundlage zur Prozessanpassung die folgenden Projekteinflussfaktoren berücksichtigt: Kritikalität, Anzahl der Projektbeteiligten, Dynamik, Arbeitskultur und Kompetenzgrad. Abhängig von der Ausprägung dieser Einflussfaktoren wird das geeignete Paradigma (agil oder klassisch) ausgewählt und projektspezifisch angepasst mit dem Ziel, die Projektrisiken zu minimieren [BT03].

Ebenso präsentieren Xu und Ramesh in ihrer Studie ein Framework zur systematischen Prozessanpassung. Dabei bewerten sie anhand einer Methode die Auswirkungen bestimmter Projekteinflussfaktoren. Aus dieser Bewertung leiten sie einen Anpassungsbedarf des Entwicklungsprozesses ab, dem sie mit entsprechenden Maßnahmen nachkommen [XR08].

Im Vergleich dazu liegt der Fokus der Metastudie von Kalus und Kuhrmann nicht auf der Vorgehensweise, sondern auf den Projektrahmenbedingungen, die für eine Prozessanpassung relevant sind. Das Ergebnis dieser Metastudie ist ein Katalog mit 49 Rahmenbedingungen, die in die Kategorien Team, Ziele sowie internes und externes Umfeld eingeteilt sind [KK13].

Nach Rong et al. müssen die Elemente für den Entwicklungsprozess passend zu den Projektrahmenbedingungen ausgewählt werden. Zudem sind auch die Grundprinzipien zu beachten, die sicherstellen, dass die Prozesselemente aufeinander abgestimmt sind und konsistent ineinander greifen. Andernfalls bestehen für das Projekt erhöhte Risiken [Ro14].

Saleh verfolgt in seiner Vorgehensweise zur Prozessanpassung einen Bottom-up-Ansatz: Zunächst werden agile Prozesselemente anhand vordefinierter Regeln passend zu den Projektrahmenbedingungen ausgewählt und zusammengesetzt. Das dazu geeignete Vorgehensmodell wird im Gegensatz zu den bisher bekannten Vorgehensweisen erst im Nachhinein bestimmt. Weist hier die Mehrzahl der ausgewählten Prozesselemente auf ein bestimmtes Vorgehensmodell hin, so wird dieses im Projekt angewendet [Sa13].

Mittlerweile ist weitgehend akzeptiert, dass der Entwicklungsprozess in der Regel projektspezifisch angepasst werden muss. Wie bei dieser Prozessanpassung dabei genau vorzugehen ist, ist jedoch in weiten Teilen noch unklar [Ro14]. Entsprechend ist das Tailoring des Entwicklungsprozesses oft intuitiv und aus dem Erfahrungsschatz der Verantwortlichen heraus getrieben. In der Folge ist das Ergebnis oft stark personenabhängig und nur bedingt von außen nachvollziehbar [BGB08]. Auch in den oben aufgeführten Studien sind die Lösungsansätze teilweise ungenau beschrieben, was eine praktische Anwendung erschwert. Zudem zeigen sie, dass oft relevante Informationen nicht berücksichtigt werden. Diese Arbeit stellt daher eine detaillierte systematische Vorgehensweise vor, die anhand relevanter Einflussfaktoren schnell und effektiv einen Entwicklungsprozess projektspezifisch anpasst.

4 Auswählen des geeigneten Paradigmas für den Entwicklungsprozess und Ermitteln von Anpassungsbedarf

In Projekten stehen zur Lösung von Entwicklungsaufgaben unterschiedliche Ansätze zur Auswahl. Entsprechend ist beim Aufsetzen eines projektspezifischen Entwicklungsprozesses zunächst dasjenige Paradigma (agil oder klassisch) auszuwählen, das zu den Erfordernissen der Entwicklungsaufgabe am besten passt. Um die in diesem Kontext relevanten Unterschiede der beiden Paradigmen zu verdeutlichen, werden im Folgenden Scrum (agil) [Sc95b] und V-Modell XT (klassisch) [VXT15] als exemplarische Vertreter gegenüber gestellt.

Einer der Hauptunterschiede liegt im Umgang mit Anforderungsänderungen. Diese werden in Scrum laufend angenommen, um ein den Anforderungen entsprechendes Produkt zu entwickeln. Dagegen werden bei einer Entwicklung nach dem V-Modell XT Änderungen ausschließlich über einen speziellen Änderungsprozess eingebracht.

Des Weiteren schreibt das V-Modell XT die Erstellung einer detaillierten Spezifikation vor. Dementsprechend besitzt eine im Vorfeld erarbeitete Anforderungsdefinition einen deutlich höheren Stellenwert als bei Scrum. Im Gegensatz dazu zielt Scrum weniger auf eine umfassende Anforderungsdokumentation ab, sondern legt den Fokus eher auf das Entwickeln eines funktionsfähigen (Teil-)Produktes.

Um diese Vorgehensweise bei Scrum zu ermöglichen, bedarf es einer engen und kontinuierlichen Zusammenarbeit zwischen den Projektbeteiligten. Durch Face-to-Face-Kommunikation und sehr kurze Feedback-Schleifen bleiben Scrum-Teams in ständigem Kontakt mit dem Anforderungsgeber. Eine solche enge Zusammenarbeit ist beim V-Modell XT nicht zwingend vorgesehen und findet daher meist nicht kontinuierlich über den gesamten Projektverlauf statt, sondern lediglich phasenweise.

Zudem zeichnen sich Scrum-Teams durch eine eigenverantwortliche und unabhängige Arbeitsweise aus. Dafür werden fachübergreifende motivierte Teams gebildet, in denen alle Teammitglieder möglichst alle Tätigkeiten übernehmen können. Im V-Modell XT ist stattdessen eine übergeordnete Projektsteuerung erforderlich, um die nach Disziplinen getrennten Teams zu koordinieren.

Aufgrund dieser Unterschiede zwischen dem agilen und dem klassischen Paradigma sind nicht beide für jede Problemart gleichermaßen gut geeignet. Damit der Entwicklungsprozess den Problemmerkmalen gerecht wird, muss daher zunächst das geeignete Paradigma ausgewählt werden. Sind die organisatorischen Rahmenbedingungen damit nicht vereinbar, wird anschließend der Entwicklungsprozess angepasst und die RE-Aktivitäten darin integriert.

4.1 Auswählen des geeigneten Paradigmas für den Entwicklungsprozess

Bei der Auswahl des geeigneten Prozessparadigmas (klassisch oder agil) stehen die *Problemmerkmale* im Vordergrund. Darunter sind Eigenschaften zu verstehen, die sich unmittelbar aus der Aufgabenstellung selbst ergeben. Diese grenzt man von *organisatorischen Rahmenbedingungen* ab, die durch das Umfeld der Projektabwicklung bedingt sind.

Als Problemmerkmale werden Sicherheitskritikalität, Stabilität der Anforderungen, Innovationsgrad und Erfolgsrelevanz von Time-to-Market betrachtet. Dabei werden genau diejenigen Merkmale fokussiert, die einen direkten Bezug zu den oben betrachteten Prinzipien des agilen Paradigmas aufweisen. Auf dieser Grundlage ist schnell erkennbar, ob dieses Paradigma zur Lösung der Problemstellung geeignet ist. Abbildung 1 zeigt die systematische Vorgehensweise zur Auswahl des geeigneten Prozessparadigmas.

Wichtig ist dabei nicht nur die Auswahl der Problemmerkmale, sondern auch die Reihenfolge, in der diese betrachtet werden. Wenn anhand der Ausprägung von einem dieser Merkmale unmittelbar entscheidbar ist, welches der Paradigmen am besten geeignet ist, so müssen die anderen Problemmerkmale nicht weiter betrachtet werden. Entsprechend ist es sinnvoll, diese Problemmerkmale vorrangig zu betrachten. Dadurch wird es möglich, schnell mit vergleichsweise wenig Aufwand ein zur Problemstellung passendes Prozessparadigma auszuwählen.

Ist beispielsweise das Problem sicherheitskritisch, so ist die Entwicklung in vielen Anwendungsbereichen durch Gesetze und Verordnungen streng reglementiert, wie zum Beispiel im Automotive- [Hi12] oder im Medizin-Bereich [TU993]. In der Regel ist dabei verpflichtend

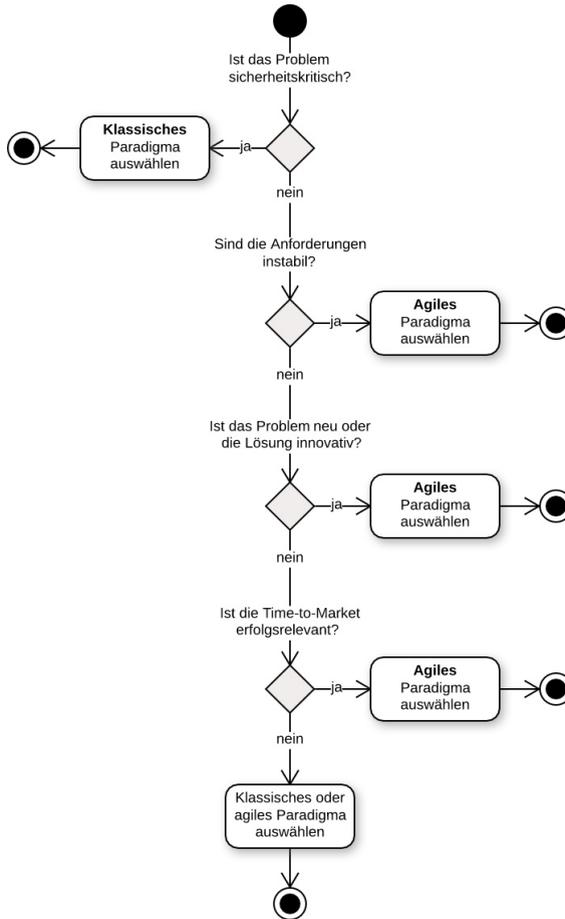


Abb. 1: Auswahl des Prozessparadigmas anhand von Problemmerkmalen

vorgegeben, dass die verschiedenen Entwicklungsschritte und -artefakte detailliert dokumentiert werden müssen, um später die Erfüllung der Anforderungen belegen zu können. Diese Forderung weicht jedoch vom agilen Paradigma ab, welches den mündlichen Dialog höher wertet als das Produzieren von Artefakten. Dabei beschränkt sich die agile Dokumentation meistens auf User Stories. Ergänzt durch Erläuterungen des Product Owners (PO) sind diese ausreichend, um ein fachliches Verständnis zu schaffen, auf dessen Grundlage entwickelt werden kann. Die dabei entstehende Dokumentation erfüllt jedoch meist nicht die formalen Vorgaben. Ist diese detaillierte Dokumentation erforderlich, kommt daher nur das (klassische) dokumenten-zentrierte Vorgehen in Frage. Probleme, die nicht sicherheitskritisch sind, können dagegen prinzipiell mit jedem der beiden Paradigmen gleichwertig gelöst werden. Für eine Entscheidung müssen dann weitere Merkmale herangezogen werden.

Falls die Produktvision noch unklar oder die Entwicklungsaufgabe in einem sehr dynamischen fachlichen Umfeld angesiedelt ist, können bei der Problemlösung auch instabile Anforderungen auftreten. Darunter versteht man Anforderungen, die innerhalb der Entwicklungszeit in kurzen Abständen neu aufgenommen oder geändert werden. Dies ist im agilen Vorgehen zu jedem beliebigen Zeitpunkt willkommen, beim klassischen Vorgehen dagegen mit vermehrtem Aufwand verbunden [Be01]. Bei Projekten mit stabilen Anforderungen kann dagegen wahlweise agil oder klassisch vorgegangen werden. Die Entscheidung bedarf also einer weiteren Betrachtung von Merkmalen.

Des Weiteren kann das zu lösende Problem in technischer oder fachlicher Hinsicht neuartig sein, bzw. einen innovativen Lösungsansatz erfordern. Um hierzu neue Ideen zu erarbeiten sind Freiräume und eine entsprechende Fehlerkultur notwendig. Darauf legt das agile Vorgehen besonderen Wert [HC01]. Liegt hingegen ein niedriger Innovationsgrad vor, so steht auch hier die Auswahl zwischen agilem und klassischem Paradigma frei.

Hat die bisherige Betrachtung noch zu keiner Entscheidung für ein Paradigma geführt, so wird beurteilt, ob eine möglichst kurze Time-to-Market (d.h. die Zeitspanne zwischen Produktentwicklung und -auslieferung) für den Projekterfolg relevant ist. Ist der Projekt- und Produkterfolg von einer kurzfristigen Auslieferung von funktionsfähigen Produktinkrementen abhängig, so ist das agile Vorgehen vorzuziehen, da hier sehr schnell einsetzbare Produktversionen entstehen, deren Verwendung dem Nutzer bereits einen Mehrwert liefert [Sc95a]. Ist dies jedoch nicht der Fall, sind agiles und klassisches Vorgehen für die Lösung des Problems gleichermaßen gut geeignet.

Das Ergebnis dieser systematischen Vorgehensweise kann also entweder für den Einsatz eines bestimmten Paradigmas sprechen oder die Auswahl zwischen den beiden Vorgehensmodellen offenhalten. Im zweiten Fall kann die Auswahl des Paradigmas anhand der Arbeitskultur oder der Präferenzen und Erfahrungen im Team erfolgen [Ce14]. Nachdem das Paradigma feststeht, muss es zur effektiven Anwendung im nächsten Schritt mit den organisatorischen Rahmenbedingungen abgeglichen werden.

4.2 Abgleichen des Paradigmas mit den organisatorischen Rahmenbedingungen

In der Realität passen die bestehenden organisatorischen Rahmenbedingungen eines Projektes nicht immer optimal zu demjenigen Prozessparadigma, das den Anforderungen des Problems am besten gerecht wird. In diesem Fall muss das Vorgehen angepasst werden, um den organisatorischen Rahmenbedingungen gerecht zu werden. Derzeit existieren nur wenige systematische Tailoring-Ansätze für agile Modelle [XR08, BT03]. Zudem stellen agile Projekte hinsichtlich einiger zentralen organisatorischen Rahmenbedingungen spezifischere Ansprüche und ziehen entsprechend mehr Anpassungsbedarf nach sich, als dies bei klassischen Projekten der Fall ist. Beispielsweise erfordert das agile Prinzip der Face-to-Face-Kommunikation eine örtliche Präsenz der Projektbeteiligten, die bei Bedarf hergestellt werden muss. Dagegen stellt eine örtliche Verteilung für einen stark dokumenten-zentrierten

Entwicklungsprozess weniger ein Hindernis dar. Daher fokussieren wir im Folgenden das agile Vorgehen, um es mit den organisatorischen Rahmenbedingungen abzugleichen und ggf. projektspezifischen Anpassungsbedarf zu ermitteln.

Als mögliche organisatorische Rahmenbedingungen werden örtliche Verfügbarkeit, konsolidierte Anforderungen, Kompetenzen der Team-Mitglieder und Anzahl der Teams betrachtet. Abhängig von ihrer Ausprägung können diese Rahmenbedingungen sich auf die Anwendbarkeit der agilen Prinzipien unmittelbar auswirken.

Sind Anforderungsgeber aufgrund einer räumlichen Trennung vom Entwicklungsteam örtlich nicht verfügbar, so beeinträchtigt das die im agilen Vorgehen vorgesehene enge Zusammenarbeit. Denn agile Projekte setzen für den Informationsaustausch auf eine Face-to-Face-Kommunikation statt auf schriftliche Dokumentation [Be01].

Die Wahrscheinlichkeit, dass in einem Projekt widersprüchliche Anforderungen auftreten, ist besonders hoch bei umfangreichen Projekten sowie bei unterschiedlichen Meinungen, örtlicher Verteilung oder einer hohen Anzahl an Anforderungsgebern. Beim agilen Vorgehen werden widersprüchliche Anforderungen und daraus resultierende Konflikte oft erst relativ spät im Laufe des Projektes bekannt, was zu einem hohen Konsolidierungsaufwand führt.

In umfangreichen Projekten (d.h. Projekten, die mehr als zwei Personenjahre an Arbeitsleistung benötigen) kommen in der Regel mehrere Entwicklerteams zum Einsatz. Dabei besteht die Gefahr, dass durch die getrennte Entwicklung der Produktteile diese beim Zusammenführen nicht optimal zueinanderpassen. Insbesondere bei nicht modularisierbaren Systemen ist in diesem Fall mit einem hohen Konsolidierungsaufwand zu rechnen.

Damit ein Team in der Lage ist, möglichst selbst organisiert und unabhängig zu arbeiten, benötigt jedes Mitglied fachliche Kenntnisse im RE- und SE-Bereich sowie kommunikative Fähigkeiten. Bei Defiziten in einem der Bereiche ist es jedoch möglich, im Projektverlauf die fehlenden Kompetenzen in den Betroffenen aufzubauen, beispielsweise durch Training on the Job [Sc95a]. Demzufolge besteht hier kein Anpassungsbedarf des Entwicklungsprozesses an sich, solange die fehlende Kompetenzen frühzeitig im Projekt erworben werden.

Die organisatorischen Rahmenbedingungen können also von den agilen Werten abweichen, was das Anwenden des agilen Paradigmas in Reinform schwierig gestaltet. Um dieser Situation gerecht zu werden und in Summe eine effektive Entwicklung zu gewährleisten, wird im Folgenden das lehrbuchmäßige agile Vorgehen entsprechend systematisch an die organisatorischen Rahmenbedingungen angepasst.

4.3 Anpassen des Entwicklungsprozesses und Integrieren der RE-Aktivitäten

Diskrepanzen zwischen dem agilen Paradigma und den organisatorischen Rahmenbedingungen sind also ein Indikator dafür, dass Anpassungsbedarf am lehrbuchmäßigen agilen

Vorgehen besteht. Nicht immer lassen sich die bestehenden organisatorischen Rahmenbedingungen eines Projektes so anpassen, dass sie mit dem agilen Paradigma kompatibel sind, da die entsprechenden Maßnahmen dafür oft sehr aufwändig sind. Stattdessen ist eine Anpassung des Entwicklungsprozesses meist eher möglich.

Um den Prozess möglichst schnell und effektiv anzupassen wird schrittweise vorgegangen:

1. Symptome erkennen
2. Konkrete Ursache für den Anpassungsbedarf ermitteln
3. Geeignete Maßnahmen zuordnen
4. Maßnahmen in den Entwicklungsprozess integrieren

Diese Schritte werden auf die oben ermittelten Anpassungsbedarfe angewendet und dafür exemplarisch Maßnahmen ergriffen (siehe Tabelle 1).

Org. Faktoren	Symptome	Ursachen	Maßnahmen und RE-Integration
Örtliche Verteilung	Keine enge Zusammenarbeit	Verschiedene Standorte	Phasenweise örtliche Versetzung des Teams oder RE-erfahrener Person, digitale Tools
Widersprüchliche Anforderungen	Spät erkannte fachliche Konflikte	Viele Anforderungsgeber	Anforderungserhebung und -konsolidierung durch PO, Konsolidierungsworkshops
		Hoher Umfang	Vorab Gesamtüberblick verschaffen
Mehrere Entwicklerteams	Technische Konflikte	Getrennte Umsetzung	Klare Definition unabhängiger Schnittstellen
		Keine Abstimmung	Kurze Entwicklungszyklen, Abstimmung

Tab. 1: Überblick über org. Rahmenbedingungen, deren Folgen und mögliche Maßnahmen

Anpassungsbedarf besteht, wenn Anforderungsgeber und Entwicklungsteam voneinander örtlich getrennt sind. Um dennoch eine enge Zusammenarbeit zu gewährleisten, sind verschiedene Maßnahmen denkbar. Eine Möglichkeit besteht darin, eine der beiden Gruppen zu versetzen, um die Face-to-Face-Kommunikation zu ermöglichen. Eine andere, oft leichter umsetzbare Maßnahme ist, erfahrene Entwickler oder Requirements Engineers phasenweise vor Ort beim Anforderungsgeber einzusetzen, um das erforderliche Domänenwissen und Verständnis der Anforderungen aufzubauen. Anschließend geben diese Personen ihr Wissen an das Entwicklungsteam weiter. Wissensaufbau vor Ort bei den Anforderungsgebern und Weitergabe dieses Wissens an das Entwicklungsteam erfolgen also zeitversetzt im Wechsel.

Ist eine örtliche Zusammenführung nicht möglich, so werden digitale Kommunikationswerkzeuge eingesetzt, um die Trennung so gut wie möglich zu überbrücken. In der Regel lässt sich dadurch jedoch nicht die gleiche Intensität der Kommunikation erreichen, die bei räumlicher Präsenz möglich ist.

Widersprüchliche Anforderungen können verschiedene Ursachen haben. Wichtig ist dabei mögliche fachliche Konfliktstellen frühzeitig zu identifizieren und aufzulösen. Je mehr Anforderungsgeber an einem Projekt beteiligt sind, umso höher ist das Potential für widersprüchliche Anforderungen. Hier ist es Aufgabe des PO, die Anforderungen der

einzelnen Anforderungsgeber zu ermitteln und zu konsolidieren, bevor diese ins Backlog einfließen. Falls der PO keine RE-Kenntnisse besitzt, kann eine RE-erfahrene Person oder ein RE-Team dabei unterstützen. Konsolidierungswshops sind eine weitere Maßnahme, um diese Konflikte zu adressieren und eine gemeinsame Zielvorstellung zwischen mehreren Anforderungsgebern zu erarbeiten.

Bei sehr umfangreichen Projekten mit hohem Entwicklungsaufwand besteht die Möglichkeit mit einem RE-Team vor Beginn der eigentlichen Entwicklung ein grobes Verständnis der Anforderungen zu erarbeiten, um sich so einen Gesamtüberblick zu verschaffen und Konflikte frühzeitig zu erkennen.

Neben den fachlichen Widersprüchen müssen eventuell auch technische bzw. umsetzungsbedingte Konflikte konsolidiert werden. Diese entstehen in der Regel, wenn für die Realisierung eines Produktes mehrere getrennte Entwicklerteams eingesetzt werden. Als Maßnahmen sind kurze Entwicklungszyklen zu etablieren, um Konflikte frühzeitig zu erkennen. Des Weiteren sind klare, unabhängige Schnittstellen zwischen den Produktteilen zu definieren, damit die Teams weit gehend unabhängig voneinander entwickeln können. So lassen sich spätere Produktpassungen reduzieren.

5 Erste Ergebnisse

Das hier vorgestellte systematische Vorgehen zur Auswahl des Entwicklungsparadigmas und der darauf basierende Anpassungsprozess wurden bereits erfolgreich in ersten Projekten angewendet.

In einem Softwareentwicklungsprojekt wurde durch den Kunden das V-Modell XT als Prozess vorgegeben. Dabei ging es um die Erstellung eines sicherheitskritischen Systems mit einem Gesamtaufwand von 60 Personentagen. Die Analyse der Problemmerkmale wies jedoch für einen Teilbereich (ca. 25 Personentage), aufgrund von instabilen und noch nicht konsolidierten Anforderungen, auf ein lokal anzuwendendes agiles Vorgehen hin. Die Betrachtung der organisatorischen Rahmenbedingungen ergab einen Anpassungsbedarf am agilen Vorgehen, da die (zeitweise) örtliche Verteilung von PO und Entwicklungsteam für die Projektdauer nicht aufgelöst werden konnte. Darüber hinaus gab es für bestimmte Bereiche der Anwendung Dokumentationsvorschriften, die eingehalten werden mussten.

Um das RE besser in den Entwicklungsprozess zu integrieren wurden verschiedene Maßnahmen ergriffen. Die örtliche Verteilung wurde ausgeglichen, indem die Kernanforderungen bereits vor Beginn der Entwicklung von einer Teilmenge des Entwicklerteams erhoben und detailliert beschrieben wurden. Dem agilen Ansatz folgend wurde die fachliche Dokumentation für die Entwickler auf das Wesentliche beschränkt. Im Vordergrund stand, dass sich die Beteiligten ein grundlegendes Verständnis der kundenseitigen Anforderungen durch intensivere Kommunikation aneignen. Zusätzlich wurde eine Person aus diesem Team als Wissensträger bezüglich der Kundendomäne aufgebaut, mit dem Ziel, später im Entwicklungsprozess den PO zu vertreten, wenn dieser nicht verfügbar ist.

In einem anderen Projekt mit einem Entwicklungsaufwand von ca. 20 Personenjahren wurde durch den Kunden eine agile Vorgehensweise vorgegeben. Die Analyse der Problemmerkmale ergab für einen Teil der Problemstellung jedoch einen Anpassungsbedarf am agilen Vorgehen. Die fertige Software sollte in über 50 verschiedenen Organisationen eingesetzt werden und flexibel an deren unterschiedliche Bedürfnisse anpassbar sein (Customizing). Wegen des hohen Gesamtaufwandes und der Diversität der Anforderungen war bei einem agilen Ansatz mit hohem Abstimmungsbedarf zwischen den Entwicklerteams zu rechnen.

Um den Abstimmungsaufwand und ggf. damit verbundene Fehlentwicklungen zu minimieren wurde ein Team gebildet, das mit dem Product Owner einen systematischen Überblick über die Anforderungen für dieses Teilproblem erarbeitet hat. Auf dieser Grundlage wurden Gemeinsamkeiten und Unterschiede in den Customizing-Anforderungen identifiziert und systematisch Problemklassen gebildet. So konnte das Problem in unabhängige Module aufgeteilt und an parallel arbeitende Entwicklerteams übergeben werden.

In beiden Fallbeispielen konnten durch die systematische Vorgehensweise Risiken und Hindernisse im Projekt bereits im Vorfeld erkannt und zuverlässig ausgeräumt werden.

6 Zusammenfassung und Ausblick

Die vorliegende Arbeit definiert eine systematische Vorgehensweise um einen Entwicklungsprozess passend zu den individuellen Projektgegebenheiten zu gestalten. Die Beispiele verdeutlichen, dass durch diese Vorgehensweise eine Identifizierung und zielgerichtete Auflösung von Hindernissen möglich ist, beispielsweise wenn die Vorgaben des Auftragsgebers nicht zum empfohlenen Prozessparadigma passen. Zu beachten ist, dass die Vorgehensweise bisher nur in wenigen Pilotprojekten erprobt wurde. Eine weitere Überprüfung im Projekteinsatz ist also erforderlich, um die Tragfähigkeit der Vorgehensweise zu validieren.

Aktuell betrachtet die Vorgehensweise die Auswahl des geeigneten Prozessparadigmas und die projektspezifische Prozessanpassung zur Integration des RE. Im nächsten Schritt wird eine systematische Auswahl für einzelne RE-Praktiken in Abhängigkeit des Projektumfeldes behandelt. Für die Zukunft ist zudem eine Erweiterung um einen Maßnahmenkatalog sinnvoll, damit viele Maßnahmen möglichst reproduzierbar abgeleitet werden können. Dieser Katalog könnte gängige Kombinationen an Rahmenbedingungen mit einer empfohlenen Maßnahmenkonstellation und vordefinierte angepasste Prozessmuster enthalten.

Dank

Ein besonderer Dank geht an Prof. Dr. Veronika Thurner (Hochschule München) und Christopher Köster (ARS) für die tatkräftige Unterstützung und fortwährende Betreuung bei der Anfertigung dieser Arbeit. Ferner danke ich der ARS Computer und Consulting GmbH für die Möglichkeit, das im Rahmen des Hochschulstudiums erforderliche Praxissemester dort zu absolvieren sowie eine anwendungsnahe Bachelor-Arbeit zu schreiben.

Literaturverzeichnis

- [Be01] Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marik, B.; Martin, R.C.; Mellor, S.; Schwaber, K.; Sutherland, J.; Thomas, D.: , Manifesto for Agile Software Development. agilemanifesto.org/, 2001. Abgerufen am 22.04.2017.
- [BGB08] Bygstad, B.; Ghinea, G.; Brevik, E.: Software development methods and usability: Perspectives from a survey in the software industry in Norway. *Interacting with Computers*, 20(83):375–385, 2008.
- [BT03] Boehm, B.; Turner, R.: Using Risk to Balance Agile and Plan-Driven Methods. *Computer*, 36:57–66, 2003.
- [Ce14] de Cesare, S.; Patel, C.; Iacovelli, N.; Merico, A.; Lycett, M.: Tailoring Software Development Methodologies in Practice: A Case Study. *Journal of Computing and Information Technology*, 16:183–184, 2014.
- [HC01] Highsmith, J.; Cockburn, A.: Agile Software Development: The Business of Innovation. *Computer*, 34(9):120–127, 2001.
- [Hi12] Hillenbrand, M.: Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen. Dissertation. Karlsruher Institut für Technologie (KIT), KIT Scientific Publishing, Februar 2012.
- [HW15] Hastie, S.; Wojewoda, S.: , Standish Group 2015 Chaos Report — Q&A with Jennifer Lynch. www.infoq.com/articles/standish-chaos-2015, Oktober 2015. Abgerufen am 22.04.2017.
- [KK13] Kalus, G.; Kuhrmann, M.: Criteria for Software Process Tailoring: A Systematic Review. S. 171–180, 2013.
- [Ro14] Rong, G.; Boehm, B.; Kuhrmann, M.; Tian, E.; Lian, S.; Richardson, I.: Towards Context-Specific Software Process Selection, Tailoring, and Composition. S. 183–184, 2014.
- [Sa13] Saleh, M. H.: Methodology for Selection of Agile Practices. Master’s thesis. American University of Sharjah, United Arab Emirates, June 2013.
- [Sc95a] Scrum.org: , The Scrum Guide. <https://www.scrumalliance.org/why-scrum/scrum-guide>, 1995. Abgerufen am 04.05.2017.
- [Sc95b] Scrum.org: , What is Scrum? www.scrum.org/resources/what-is-scrum, 1995. Abgerufen am 22.04.2017.
- [SGI02] Standish Group International, Inc.: , What are Your Requirements? 2003 Chaos Top 10. www.standishgroup.com, 2002.
- [TU993] TÜV SÜD Industrie Service GmbH, München. RICHTLINIE 93/42/EWG DES RATES vom 14. Juni 1993 über Medizinprodukte, 1993. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:1993L0042:20071011:de:PDF>, abgerufen am 18.06.2017.
- [VXT15] Verein zur Weiterentwicklung des V-Modell XT e.V. (Weit e.V.), München. V-Modell XT, version 2.0. Auflage, 2015. www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html, abgerufen am 22.04.2017.
- [XR08] Xu, P.; Ramesh, B.: Using Process Tailoring to Manage Software Development Challenges. *IT Professional*, 10(4):39–45, 2008.