



Fraunhofer Institut
Experimentelles
Software Engineering

ReqMan-Praktiken im RUP und XP



Autoren:
Sebastian Adam
Thomas Olsson

IESE-Report Nr. 018.06/D
Version 1.0
20. Januar 2006

Eine Publikation des Fraunhofer IESE

Das Fraunhofer IESE ist ein Institut der Fraunhofer-Gesellschaft. Das Institut transferiert innovative Software-Entwicklungstechniken, -Methoden und -Werkzeuge in die industrielle Praxis. Es hilft Unternehmen, bedarfsgerechte Software-Kompetenzen aufzubauen und eine wettbewerbsfähige Marktposition zu erlangen.

Das Fraunhofer IESE steht unter der Leitung von
Prof. Dr. Dieter Rombach (geschäftsführend)
Prof. Dr. Peter Liggesmeyer
Fraunhofer-Platz 1
67663 Kaiserslautern

Abstract

In dieser Arbeit wird anhand der Literatur eine Untersuchung durchgeführt, in wie weit die von ReqMan [ODK05] als essentiell erachteten Praktiken des Requirements Engineering im Rational Unified Process (RUP) und dem eXtremeProgramming (XP) Beachtung finden. Dazu stellt dieser Bericht eine Abbildung zwischen den ReqMan-Praktiken und den entsprechenden Aktivitäten vor und deckt unberücksichtigte Praktiken in den einzelnen Methoden auf.

Schlagworte: ReqMan, requirements engineering, extreme programming, RUP

Inhaltsverzeichnis

1	Einleitung	1
2	Die ReqMan-Architektur	3
2.1	Basispraktiken	4
2.2	Fortgeschrittene Praktiken	6
3	ReqMan-Praktiken im Rational Unified Process (RUP)	8
3.1	Explizite Requirements Engineering Aktivitäten	8
3.2	Bewertung im Kontext der ReqMan-Praktiken	11
4	ReqMan-Praktiken im eXtremeProgramming (XP)	15
4.1	Explizite Requirements Engineering Aktivitäten	15
4.2	Bewertung im Kontext der ReqMan-Praktiken	18
5	Zusammenfassung und Ausblick	21

1 Einleitung

Speziell für die Bedürfnisse kleinerer und mittelständiger Softwareunternehmen (KMU) wurde im Rahmen des Forschungsprojekts „ReqMan“ [ODK05] ein Rahmenwerk etablierter Praktiken und Techniken des Requirements Engineering entwickelt. Das Ziel hierbei ist es, interessierten Firmen eine Unterstützung für die systematische Verbesserung ihrer Anforderungsprozesse zu geben, ohne umfangreiche Prozessrahmenwerke einführen zu müssen. Vielmehr erlaubt der Ansatz die flexible und inkrementelle Erweiterung bestehender Prozesse um bisher unberücksichtigte und dennoch notwendige Verfahren des Requirements Engineering¹.

Als wesentliche Eigenschaft von ReqMan kann die Unterscheidung von Praktiken und Techniken angesehen werden. Praktiken unterteilen sich hierbei nochmals in Basis-, fortgeschrittene und Kontextpraktiken und können auf verschiedene Phasen des Anforderungsprozesses (Erhebung, Analyse, Dokumentation, Verifikation & Validierung, Management) verteilt werden.

Eine Praktik wird dabei als abstrakte Aktivität wie beispielsweise „Ermitteln der funktionalen Anforderungen“ oder als Verhaltensprinzip wie „Beziehe alle Beteiligten ein“ verstanden. Eine Praktik definiert somit, was innerhalb des Anforderungsprozesses geschehen soll, ohne eine konkrete Umsetzung zu beschreiben. Dies ist die Aufgabe einer Technik.

Zwischen Techniken und Praktiken bestehen in ReqMan n-zu-m-Beziehungen. Eine Praktik kann daher, abhängig vom Kontext, durch verschiedene Techniken realisiert werden, während eine Technik gegebenenfalls zur Umsetzung verschiedener Praktiken herangezogen werden kann (siehe Abbildung 1).

Da ReqMan das Ziel verfolgt, die Anforderungsprozesse von Softwareunternehmen nachhaltig zu verbessern, wurden basierend auf einer Literaturrecherche zahlreiche etablierte Praktiken und Techniken ermittelt. Die ermittelten Praktiken und Techniken können als wesentlicher Stand der Forschung im Requirements Engineering aufgefasst werden.

¹ Da gerade in KMUs häufig Projektmanagement, Anforderungsmanagement und Qualitätssicherung verschmelzen, wird in ReqMan daher das Requirements Engineering etwas breiter ausgelegt, als es in der Literatur üblich ist [DKO04].

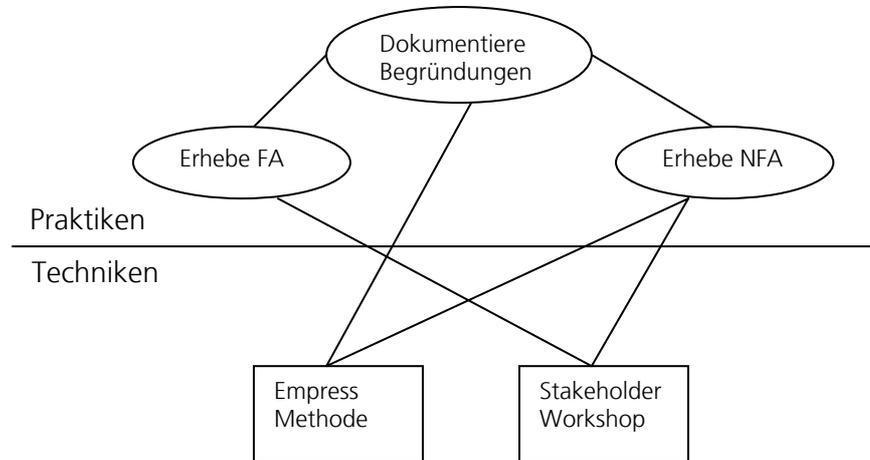


Abbildung 1: Zuweisung von Techniken zu Praktiken

Da ReqMan allerdings keine Prozessvorgaben macht, erscheint die Frage interessant, in wie weit die einzelnen Praktiken bezüglich der Methoden des „Rational Unified Process“ [Kru00], als moderner Vertreter eines umfassenden Prozessrahmenwerks und des „eXtreme Programming“ [Bec00], als Vertreter der agilen und vor allem in kleineren Unternehmen vielbeachteten Entwicklungsmethoden, Beachtung finden.

Die im Rahmen dieses Berichts entwickelte Gegenüberstellung kann zum einen dazu genutzt werden, geeignete Integrationspunkte für ReqMan-Praktiken in den einzelnen Prozessen zu identifizieren, zum anderen aber auch einen einfachen Überblick über essentielle und unberücksichtigte Praktiken des Requirements Engineering geben.

Der weitere Aufbau dieses Berichts untergliedert sich wie folgt: Kapitel 2 gibt zunächst einen Überblick über die Praktiken des ReqMan-Rahmenwerks. In Kapitel 3 und 4 werden dann gesondert Aktivitäten des Requirements Engineering im RUP und XP betrachtet und mit den von ReqMan vorgeschlagenen Praktiken verglichen. In Kapitel 5 findet sich schließlich eine Zusammenfassung der gewonnenen Erkenntnisse.

2 Die ReqMan-Architektur

Das ReqMan-Rahmenwerk kategorisiert Praktiken des Requirements Engineering zum einen nach den Phasen des Anforderungsprozesses, in denen sie eine wesentliche Rolle spielen, zum anderen nach dem Nutzen, den sie in dieser Phase liefern.

Als Phasen werden hierbei

- Anforderungserfassung
- Anforderungsanalyse
- Anforderungsdokumentation
- Anforderungsverifikation und –validierung
- Anforderungsmanagement

voneinander abgegrenzt, während die Praktiken gemäß ihres Nutzen in

- Basispraktiken
- Fortgeschrittene Praktiken
- Kontextpraktiken

untergliedert sind.

Basispraktiken weisen einen hohen Nutzen unabhängig des Kontextes auf und sollten daher in jedem Anforderungsprozess Anwendung finden. Fortgeschrittene Praktiken haben sich ebenfalls kontextunabhängig als hilfreich erwiesen, setzen aber andere Praktiken aus dem Bereich der Basispraktiken voraus und können daher erst einbezogen werden, wenn diese bereits vorhanden sind.

Kontextpraktiken sind schließlich Praktiken, die nur in einem entsprechenden Kontext einen gewissen Nutzen liefern. So ist beispielsweise die Praktik „Geschäftsprozessanalyse“ in der Regel ohne Nutzen in Entwicklungsprozessen, welche sich auf eine Software zur Steuerung technischer Systeme beziehen.

Die Gliederung nach Phasen und unterschiedliche Nutzungsgraden ist in Abbildung 2 anschaulich dargestellt. Der schichtenartige Aufbau verdeutlicht den Grundsatz, dass bei der Nutzung von ReqMan zur Verbesserung des Anforderungsprozesses zunächst die Basispraktiken integriert werden, bevor darauf aufbauend fortgeschrittene oder kontextrelevante Praktiken einbezogen werden können.

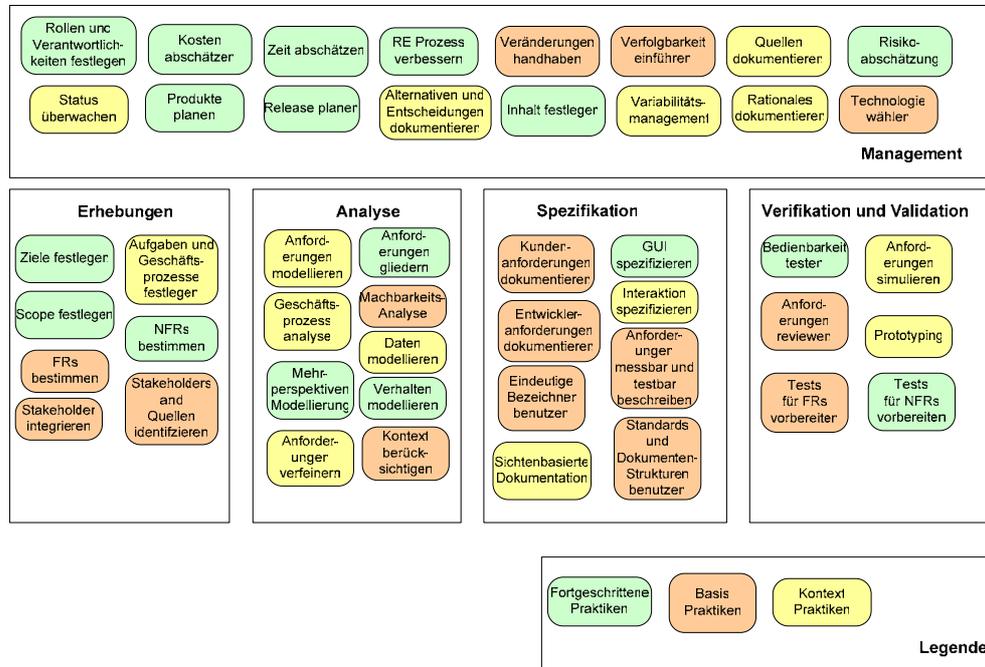


Abbildung 2: Übersicht der ReqMan-Praktiken

In diesem Bericht beschränken wir uns auf die Basis- und fortgeschrittenen Praktiken, da nur diese aufgrund ihrer Kontextunabhängigkeit Bestandteil unserer Untersuchungen des RUP und XP sein werden. Abschnitt 2.1 und 2.2 beschreiben kurz diese Praktiken und ihre Ziele.

2.1 Basispraktiken

Erfassung

Stakeholder integrieren: Ziel ist die Integration der möglichen Stakeholder/Stakeholder-Klassen in den Anforderungsprozess zur möglichst vollständigen Erfassung der Anforderungen an ein System.

Stakeholder und Quellen identifizieren: Vollständige Erfassung der Ziele und Anforderungen durch Ermitteln aller relevanten Stakeholder und Quellen (Externe Komponenten, Systemdokumentation usw.), die eine Verbindung zur Systementwicklung beziehungsweise dem System aufweisen.

Funktionale Anforderungen (FRs) bestimmen: Das Ziel der Erhebung der Funktionalen Anforderungen ist die optimale Unterstützung der Benutzenden zur Erfüllung individueller Aufgaben mit dem System.

Analyse

Machbarkeit prüfen: Sicherstellen, dass alle Anforderungen an das System durch das System erfüllt werden können. Erst damit ist die Entwicklung des Systems möglich.

Kontext berücksichtigen: Ziel ist die optimale Integration eines Systems in seine Umgebung.

Dokumentation

Entwickleranforderungen dokumentieren: Ziel ist es, die Anforderungen für die Entwicklung zu dokumentieren.

Kundenanforderungen dokumentieren: Ziel ist es, die Anforderungen der Kunden zu dokumentieren.

Eindeutige Bezeichner benutzen: Ziel ist es, eine eindeutige Identifikation aller Anforderungen zu gewährleisten.

Standards und Dokumentenstruktur benutzen: Ziel ist die Verwendung einer Dokumentenstruktur und Erfüllung von Dokumenten-Standards zur Erhöhung der Qualität und insbesondere der Verständlichkeit und Vollständigkeit eines Anforderungsdokuments.

Anforderungen messbar und testbar beschreiben: Ziel ist es, Anforderungen überprüfbar zu spezifizieren.

Verifikation & Validierung

Anforderungen reviewen: Ziel ist es, die Qualitätseigenschaften von Anforderungen in frühen Phasen der Softwareentwicklung zu überprüfen und sicherzustellen.

Tests für funktionale Anforderungen (FRs) vorbereiten: Ziel ist es, Testfälle für die Testphase erstellen.

Management

Technologie wählen: Ziel ist es, systematisch Techniken, Werkzeuge und Vorgehensweisen zu wählen, die geeignet für den Kontext sind.

Verfolgbarkeit einführen: Diese Aktivität beschäftigt sich mit der Verwaltung der Anforderungen.

Veränderungen handhaben: Ziel ist es, im Falle von Anforderungsänderungen angemessen und effizient zu reagieren.

2.2 Fortgeschrittene Praktiken

Erfassung

Ziele erheben: Festlegen der Ziele der verschiedenen Stakeholder, die von dem System unterstützt werden sollen.

Scope festlegen: Ziel ist es, sicher zu stellen, dass nur die zur Erstellung des Systems notwendigen Anforderungen im Rahmen des Projekts angegeben werden.

Nichtfunktionale Anforderungen (NFRs) festlegen: Ziel ist die Erhebung der Nichtfunktionalen Anforderungen zur besseren Unterstützung der Geschäftsprozesse und Ziele der Unternehmen und der Benutzenden.

Analyse

Anforderungen gliedern: Ziel ist es, Anforderungen zu verfeinern und zu detaillieren, um das Verständnis der Problematik zu erhöhen.

Mehrperspektivenmodellierung: Ziel ist es, die erhobenen Anforderungen von verschiedenen Perspektiven aus betrachtet zu modellieren.

Verhalten modellieren: Ziel ist es, das Verhalten des Benutzers zu modellieren.

Dokumentation

GUI spezifizieren: Ziel ist es, die graphische Schnittstelle des Systems zu beschreiben.

Verifikation & Validierung

Bedienbarkeit testen: Ziel ist es, die Benutzungsfreundlichkeit eines Systems sicherzustellen.

Tests für nichtfunktionale Anforderungen (NFRs) vorbereiten: Ziel ist es, Testfälle für die Testphase zu erstellen.

Management

Rollen und Verantwortlichkeiten festlegen: Ziel ist ein systematischer und strukturierter Anforderungsprozess.

Kosten schätzen: Ziel ist es, möglichst zutreffende Informationen über die Kosten, die mit einem Projekt verbunden sind, zu erhalten.

Zeit abschätzen: Ziel ist es, möglichst genau den Zeitpunkt der Fertigstellung eines Projekts zu bestimmen.

Risikoabschätzung: Das Ziel der Evaluation von Risiken ist es, Probleme und ihre Auswirkungen, die während der Umsetzung einer Anforderung auftreten kön-

nen, zu ermitteln um optimale Vorbereitungen zur Bewältigung der Risiken treffen zu können und somit die Auswirkungen der Risiken zu minimieren.

Produkt planen: Ziel ist es, die Produkte in Bezug auf die zu realisierenden Anforderungen zu planen.

Release planen: Ziel ist es, Inhalte und Zeitpunkt des nächsten Release zu planen.

Inhalt festlegen: Ziel ist es, die Inhalte der Anforderungsdokumente festlegen.

Anforderungsprozess verbessern: Ziel ist das Aufdecken von Problemen und Verbesserungspotentialen bezüglich des Anforderungsprozesses eines Unternehmens, Lösung der Probleme sowie Nutzung der Verbesserungspotentiale zur Steigerung der Qualität des Anforderungsprozesses.

3 ReqMan-Praktiken im Rational Unified Process (RUP)

Der Rational Unified Process [Kru00] gilt als De-Facto-Standardprozess für moderne objektorientierte Softwareentwicklung [Ess03] und basiert auf sechs „Best Practices“, die sich in den letzten Jahrzehnten in der Softwarebranche als nützlich erwiesen haben:

- Iterative Entwicklung
- Gemanagte Anforderungen
- Komponentenbasierte Architekturen
- Visuelle Modellierung
- Kontinuierliche Qualitätssicherung
- Kontrollierte Änderungen

Als die drei wesentlichen Eigenschaften des RUP können die Architekturzentriertheit, die Anwendungsfallgetriebenheit und das iterative Vorgehen bezeichnet werden. Aus Sicht des Requirements Engineering ist insbesondere die starke Beachtung von Anwendungsfällen (Use Cases) in allen Entwicklungsschritten von Interesse.

Der Softwareentwicklungsprozess untergliedert sich in die vier Phasen „Inception“, „Elaboration“, „Construction“ und „Transition“, welche wiederum in einzelne Iterationen unterteilt werden können. In jeder Iteration werden die Arbeitsschritte „Business Modeling“, „Requirements“, „Analysis & Design“, „Implementation“, „Test“, „Deployment“, „Configuration & Change Management“, „Project Management“ und „Environment“ (sequentiell) durchlaufen, wobei die Anteile dieser Disziplinen in den einzelnen Phasen variieren.

Abschnitt 3.1 gibt einen Überblick über die für das Requirements Engineering relevanten Aktivitäten des RUP, bevor Abschnitt 3.2 eine Abbildung einzelner ReqMan-Praktiken auf RUP-Aktivitäten vornimmt.

3.1 Explizite Requirements Engineering Aktivitäten

Business Modeling

Die Geschäftsprozessmodellierung ist der erste Arbeitsschritt im RUP. Dieser findet nur bei der Entwicklung von Systemen, welche sich mit der Unterstüt-

zung von Geschäftsprozessen beschäftigen, Anwendung². Ziel ist es, dass alle Stakeholder ein vollständiges Verständnis über die fachlichen Abläufe aller für das System relevanten Geschäftsprozesse erlangen. Dies bildet die Quelle für die anschließende Anforderungserfassung.

Wichtige Aktivitäten mit Bezug zum Requirements Engineering sind hierbei:

- Analyse der gegenwärtigen Organisation und ihrer Prozesse
- Festlegung von (Geschäftsprozess-) Zielen
- Pflege eines fachlichen Wörterbuches (Glossar)
- Erfassung von Geschäftsregeln
- Modellierung geschäftlicher Use Cases
- Modellierung geschäftlicher Objekte
- Definition der Automatisierung, also welche Teile der Geschäftsprozesse durch Software unterstützt werden sollen

Requirements

Das Anforderungsmanagement ist die treibende Kraft für die Entwicklung im RUP [Ess03]. Ziel ist es, einen Konsens der Stakeholder darüber zu erlangen, was das System leisten soll. Somit können die Grenzen und der Umfang des Systems definiert und die Grundlage für die weitere Entwicklung und Planung des Systems gelegt werden.

Der Aktivitäten und „Workflowdetails“ [Ess03] mit Bezug zum Requirements Engineering sind in diesem Arbeitsschritt:

- Problem analysieren
- Verstehen aller Stakeholder und ihrer Bedürfnisse
- Umfang des Projekts managen
- Systemanforderungen definieren
- Systemanforderungen verfeinern
- Sich ändernde Anforderungen managen
- Prototyping der Benutzerschnittstelle

² Der RUP sollte stets auf den gegebenen Einsatzkontext zugeschnitten werden, da er in seiner umfassenden, generischen Form oftmals zu komplex für ein konkretes Projekt ist.

Analysis & Design

Da dieser Arbeitsschritt den Übergang von den Anforderungen hin zum technischen Design des Systems beschreibt, ist er von wichtiger Bedeutung im RUP. Eine interne Abgrenzung zwischen „Analysis“ und „Design“ kann dadurch gezogen werden, dass „Analysis“ den technologieneutralen Entwurf der Anwendung darstellt, während das „Design“ diese um alle notwendigen Ergänzungen für eine konkrete Plattform erweitert. Im Kontext des Requirements Engineering ist daher „Analysis“ nicht mit Anforderungsanalyse zu verwechseln. Ziel des gesamten Arbeitsschrittes ist die Entwicklung einer Systemarchitektur, welche die gegenwärtigen Anforderungen realisiert, aber dennoch flexibel genug für zukünftige Änderungen ist.

Typische Aktivitäten des Requirements Engineering sind in diesem Arbeitsschritt nicht vorgesehen.

Implementation

Der Arbeitsschritt „Implementierung“ hat zum Ziel, lauffähige Einzelkomponenten zu entwickeln, diese zu testen und zu einem lauffähigen Teilsystem zu integrieren.

Typische Aktivitäten des Requirements Engineering sind in diesem Arbeitsschritt nicht vorgesehen.

Test

Der „Test“ hat die Aufgabe, die ausführbare Software auf Erfüllung der Anforderungen hin zu überprüfen. Hierzu sollen Fehlerquellen aufgedeckt und identifiziert werden.

Wichtige Aktivitäten mit Bezug zum Requirements Engineering sind hierbei:

- Erstellung von Testfällen

Deployment

Dieser Arbeitsschritt hat das Ziel, eine reibungslose Auslieferung des Systems und somit die Verfügbarkeit und Benutzbarkeit beim Kunden zu gewährleisten.

Wichtige Aktivitäten mit Bezug zum Requirements Engineering sind hierbei:

- Verifikation des Gesamtprodukts

Configuration & Change Management

Dieser unterstützende Arbeitsschritt bezieht sich auf die Verwaltung und Sicherung von „späteren“ Projektartefakten wie beispielsweise dem Quellcode und darf nicht mit dem Management von Anforderungsänderungen verwechselt werden.

Typische Aktivitäten des Requirements Engineering gibt es in diesem Arbeitsschritt nicht.

Project Management

Der Arbeitsschritt „Project Management“ ist ebenfalls unterstützend und arbeitet eng mit den anderen Arbeitsschritten, insbesondere „Requirements“ zusammen, da die ermittelten Anforderungen eine wichtige Grundlage für die Aufwandsabschätzung und Projektplanung darstellen. Zu den Aufgaben des Projektmanagements gehören unter Anderem konkurrierende Ziele verschiedener Stakeholder in Einklang zu bringen oder Risiken abzuschätzen und zu eliminieren. Vor allem dem letztgenannten Risikomanagement kommt im RUP eine hohe Bedeutung zu.

Wichtige Aktivitäten des Requirements Engineering sind in diesem Arbeitsschritt:

- Planung der gesamten Entwicklung hinsichtlich Phasen, Iterationen, Ressourcen und Kontrollindikatoren
- Planung einzelner Iterationen
- Projekt- und Risikokontrolle

Environment

Der unterstützende Arbeitsschritt „Environment“ dient der Bereitstellung der infrastrukturellen Umgebung im Projekt durch Werkzeuge, Entwicklungsrichtlinien und insbesondere die kontextspezifische Zuschneidung des RUP. Dieser Arbeitsschritt ist somit weitestgehend unabhängig von den operativen Schritten.

Im Hinblick auf das Requirements Engineering sind folgende Aktivitäten von Bedeutung:

- Entwicklung von Richtlinien und Vorlagen für einzelne Aktivitäten (z.B. für die Use-Case-Modellierung)
- Entwicklung einer (projektspezifischen) Anpassung/Optimierung des RUP
- Analyse der IST-Organisation und Entdeckung von Verbesserungspotentialen

3.2 Bewertung im Kontext der ReqMan-Praktiken

Wie die Ausführungen in Abschnitt 3.1 zeigen, sind im Rational Unified Process zahlreiche Aktivitäten vorgesehen, die aus Sicht des Requirements Engineering von Bedeutung sind.

Tabelle 1 gibt einen Überblick über die Abbildung einzelner ReqMan-Praktiken auf RUP-Aktivitäten.

Wie anschaulich zu erkennen berücksichtigt der Rational Unified Process annähernd alle Basis- und fortgeschrittenen Praktiken des ReqMan-Rahmenwerks. Auch einige Kontextpraktiken, welche in Kapitel 2 nicht näher dargelegt wurden, finden Beachtung.

Die nicht explizit im RUP abgebildete ReqMan-Praktiken (Zeilen ohne Kreuz in der Tabelle) können dennoch prinzipiell von diesem unterstützt werden, soweit entsprechende Richtlinien, Vorlagen oder Werkzeuge hierfür existieren. Als Beispiel kann die ReqMan-Aktivität „Eindeutige Bezeichner verwenden“ angeführt werden, welche sich nicht durch eine konkrete Aktivität, sondern nur durch entsprechende Richtlinien und Vorlagen realisieren lässt. Entsprechende Dokumente könnten durch die Meta-Aktivität „Entwicklung von Richtlinien und Vorlagen für einzelne Aktivitäten“ aus dem Arbeitsschritt „Environment“ realisiert werden.

Da dies oftmals eine Frage der individuellen Anpassung des RUP ist, konnten hierzu in der Literatur [Ess03, Kru00] jedoch keine konkreten Angaben gefunden werden.

Deutlich festzustellen ist, dass oftmals mehrere ReqMan-Praktiken von einer gemeinsamen RUP-Aktivität realisiert werden, anders herum allerdings abhängig vom Kontext auch eine ReqMan-Praktik durch verschiedene RUP-Aktivitäten umgesetzt werden kann.

Zusammenfassend lassen sich im RUP alle wesentlichen Praktiken des Requirements Engineering und somit eine hohe Abdeckung mit den von ReqMan vorgeschlagenen Praktiken identifizieren. Auch die Zuordnung zu den entsprechenden Phasen im Anforderungsprozess ist in ReqMan annähernd identisch.

	Verstehen aller Stakeholder und ihrer Bedürfnisse	Systemanforderungen definieren	Systemanforderungen verfeinern	Festlegung von (Geschäftsprozess-) Zielen	Umfang des Projekts managen	Analyse der gegenwärtigen Organisation und ihrer Prozesse	Entwicklung von Richtlinien und Vorlagen für einzelne Aktivitäten	Prototyping der Benutzerschnittstelle	Erstellung von Testfällen	Anpassung/Optimierung des RUP	Sich ändernde Anforderungen managen	Planung der Entwicklung	Planung einzelner Iterationen
Stakeholder integrieren	(X)												
Stakeholder und Quellen identifizieren	X												
Funktionale Anforderungen bestimmen		X											
Ziele erheben	X			X									
Scope festlegen					X								
Nichtfunktionale Anforderungen festlegen			X										
Machbarkeit prüfen													
Kontext berücksichtigen						X							
Anforderungen gliedern			X										
Mehrperspektivenmodellierung	X												
Verhalten modellieren													
Entwickleranforderungen dokumentieren			X										
Kundenanforderungen dokumentieren			X										
Eindeutige Bezeichner benutzen													
Standards und Dokumentenstruktur benutzen						X							
Anforderungen messbar und testbar beschreiben													
GUI spezifizieren								X					
Anforderungen reviewen													
Tests für FRs vorbereiten									X				
Bedienbarkeit testen								X					
Tests für NFRs vorbereiten									X				
Technologie wählen						X				X			
Verfolgbarkeit einführen											X		
Veränderungen handhaben											X		
Rollen und Verantwortlichkeiten festlegen												X	X
Kosten schätzen												X	X
Zeit abschätzen												X	X
Risikoabschätzung												X	X
Produkt planen					X								
Release planen					X								X
Inhalte festlegen						X							
Anforderungsprozess verbessern							X			X			

Tabelle 1: Abbildung von ReqMan-Praktiken auf RUP-Aktivitäten

In wie weit die konkreten Ausführungen dieser Aktivitäten in einzelnen Unternehmen aber den Zielen der zugewiesenen Praktiken entsprechen, kann an-

hand der Literatur nicht eindeutig beantwortet werden. Durch die individuellen Ausgestaltungsmöglichkeiten (Instanziierungen) des RUP bleibt daher die Qualität der Anforderungsprozesse offen.

Assessments und Verbesserungen auf ReqMan-Basis sind daher auch in Unternehmen sinnvoll, welche den RUP anwenden. Die in Tabelle 1 vorgestellte Abbildung gibt für solche Untersuchungen eine Hilfestellung, da hierdurch die betroffenen Stellen im Prozess leichter aufgedeckt werden können.

4 ReqMan-Praktiken im eXtremeProgramming (XP)

eXtremeProgramming (XP) [Bec00] zählt zu den bekanntesten Vertretern der agilen Softwareentwicklungsmethoden. Agile Methoden zielen auf die besonderen Anforderungen von Softwareentwicklungsprojekten, die von kleinen Teams durchgeführt werden und sich durch ständig ändernde Anforderungen auszeichnen [Bec00]. Die Einfachheit und Flexibilität stehen im Mittelpunkt, weshalb diese Ansätze vor allem in kleineren Softwareunternehmen besondere Beachtung gefunden haben [ODK05]. Die Ziele sind analog zu anderen Methoden auch hier, Risiken und Kosten zu minimieren und die Kundenzufriedenheit zu erhöhen.

In XP treten an Stelle eines umfangreichen, gut dokumentierten Prozessrahmenwerks einige wenige Techniken, Grundprinzipien (z.B. Unmittelbares Feedback, Einfachheit anstreben, Inkrementelle Veränderung, Veränderungen wollen, Qualitätsarbeit) und Werte (Kommunikation, Feedback, Einfachheit, Mut), die die Basis eines Softwareentwicklungsprozesses bilden. Ein konkreter Vorgehensprozess wird bei der Entwicklung nicht definiert, vielmehr nur die Einhaltung der 12 Techniken vorgeschrieben.

Abschnitt 4.1 gibt einen Überblick über die aus Sicht des Requirements Engineering relevanten Techniken des XP, bevor Abschnitt 4.2 Abbildung einzelner ReqMan-Praktiken auf diese vornimmt.

4.1 Explizite Requirements Engineering Aktivitäten

Kunde vor Ort

Das eXtremeProgramming setzt voraus, dass Vertreter der Kundenorganisation am Entwicklungsprojekt teilnehmen und möglichst oft vor Ort sind. Der Kunde beschreibt die Anforderungen an das zu entwickelnde System mit Hilfe informeller, so genannter Nutzergeschichten, die jeweils eine Eigenschaft des Systems aus seiner Sicht beschreiben. Daneben steht der Kunde den Entwicklern während der Programmierung als Ansprechpartner zur Verfügung, um fachliche Fragen schnell zu klären. Diese XP-Technik bildet auch die Grundlage für andere Techniken wie beispielsweise das Planungsspiel und Metapher.

Diese XP-Technik hat somit einen Bezug zum Requirements Engineering.

Planungsspiel

Das Planungsspiel dient dazu, das nächste Release oder eine nächste Iteration der Software zu planen. Dazu werden zunächst vom Kunden die Anforderung

in Form von Nutzergeschichten formuliert, vom Entwickler hinsichtlich Realisierungsaufwand geschätzt und anschließend vom Kunden priorisiert, um den Umfang der nächsten Iteration festzulegen. In XP richtet sich jedoch nicht die Dauer der Iteration nach den Anforderungen, sondern die Anzahl realisierter Anforderungen in einer Iteration nach deren (festen) Dauer.

Durch die Einbeziehung des Kunden hat auch diese XP-Technik somit einen Bezug zum Requirements Engineering.

Metapher

Metaphern sind bildliche Vorstellungen davon, wie ein System funktionieren soll. Sie liefern für alle Projektbeteiligten eine gemeinsame Basisvorstellung durch Analogie zu einem realen Gegenstand. Durch eine Metapher werden auch technische Vorgaben mitgegeben, welche insbesondere in Objektorientierten Systemen unmittelbar auf die Architektur Einfluss nehmen können.

Da Metapher als gemeinsame Vorstellungsbasis zwischen Kunde und Entwickler eingesetzt werden, hat diese XP-Technik somit einen Bezug zum Requirements Engineering.

Kurze Releasezyklen

In XP werden Releases in kurzen Zeitabständen entwickelt. Ziel ist es, praktische Erfahrungen mit den bereits realisierten Funktionen zu sammeln, um insbesondere zu erkennen, ob die Vorstellung des Kunden erfüllt werden konnten. Eventuelle Fehlentwicklungen werden frühzeitig erkannt, was Risiko reduziert. Releases werden nochmals in Iterationen unterteilt, wobei diese, trotz neuer Funktionalität aufgrund organisatorischer Engpässe (z.B. Schulung) nicht in diesen kurzen Zeitabständen ausgeliefert werden können.

Zusammenfassend dienen die kurzen Releasezyklen, möglichst schnell einen Kundennutzen zu generieren, aber auch Feedback für die weitere Entwicklung zu erhalten. Diese XP-Technik hat somit einen Bezug zum Requirements Engineering.

Testen

In XP kommt dem Testen die zentrale Qualitätssicherungsaufgabe zu, da zunächst Testfälle geschrieben werden und erst anschließend die Programmierung dessen erfolgt, was getestet werden soll (Test first). Entsprechend werden (Komponenten-)Tests in sehr kurzen Zeitabständen (wenige Minuten) ausgeführt und während der gesamten Entwicklung durch mehrmalige Wiederholung zur Qualitätssicherung eingesetzt. So können Programmierfehler frühzeitig gefunden werden und Fehler durch späteres Refactoring verhindert werden. Daneben dienen Akzeptanztest der Überprüfung der Anforderungen des Kunden und des Projektfortschritts.

Aufgrund seines qualitätssichernden Charakters hat diese XP-Technik somit einen Bezug zum Requirements Engineering.

Einfaches Design

Ein gutes Design genießt in XP ebenso wie in der herkömmlichen Softwareentwicklung besondere Beachtung. Ziel in XP ist es allerdings, das Design so einfach zu halten wie möglich, um nur die gerade benötigten Anforderungen umsetzen zu können. „Design auf Vorrat“ [Lip02] gibt es in XP nicht, ebenso fehlt ein Bezug zum Requirements Engineering.

Refactoring

Das Ziel von Refactoring ist die Verbesserung der Struktur eines Softwaresystems bei Beibehaltung der bestehenden Funktionalität. Eine Grundvoraussetzung sind Komponententests, um die Gefahr unerwünschter Seiteneffekte zu verringern.

Diese XP-Technik ist auch Sicht des Requirements Engineering ohne Bedeutung.

Programmieren in Paaren

Beim Programmieren in Paaren erarbeiten zwei Entwickler gemeinsam an einem Rechner eine Lösung und implementieren sie. Die zugrunde liegende Idee besteht in einem in die Programmierung integrierten Code- und Design-Review. Während beispielsweise ein Entwickler codiert, verifiziert der andere dessen Ausführungen und überlegt sich Optimierung oder Alternativen.

Diese qualitätssichernde Technik hat somit einen Bezug zum Requirements Engineering.

Gemeinsame Verantwortlichkeit

In einem XP-Projekt gehören alle erstellten Produkte dem ganzen Team. Jeder ist daher berechtigt, Dokumente und Quellcode zu ändern. Gemeinsame Verantwortlichkeit bedeutet aber auch, dass bei Problemen das ganze Team und nicht einzelne Mitarbeiter verantwortlich sind. Ein Bezug zum Requirements Engineering besteht nicht.

Fortlaufende Integration

Änderungen und Erweiterung werden sofort in ein lauffähiges Gesamtsystem integriert. Dadurch wird geänderter Quellcode den anderen Entwicklern zur Verfügung gestellt und gleichzeitig das lauffähige System ergänzt.

Im Hinblick auf das Requirements Engineering ist diese Technik ohne Bedeutung.

Programmierstandards

Programmierstandards legen die äußere Form des Quelltextes fest. Daneben werden häufig angewendete Werkzeuge und deren Konfiguration als Programmierstandards aufgefasst. Ziel ist es dadurch die Bearbeitung und Kontrolle von Quellcode anderer Teammitglieder zu erleichtern.

Diese XP-Technik hat somit einen Bezug zum Requirements Engineering.

40-Stunden-Woche

Um stets ausgeruhte und konzentrierte Mitarbeiter zu haben, sollte die 40-Stunden-Woche nicht länger als zweimal hintereinander überschritten werden. Ständige oder häufige Überstunden werden als Symptom für Probleme im Entwicklungsprozess oder der Organisationsstruktur gesehen.

Diese XP-Technik hat allerdings keinen Bezug zum Requirements Engineering.

4.2 Bewertung im Kontext der ReqMan-Praktiken

Wie die Ausführungen in Abschnitt 4.1 gezeigt haben, sind im eXtremeProgramming nur wenigen Techniken explizit vorgesehen und es wird bewusst auf einfache und schnelle Lösungen gezielt.

Tabelle 2 gibt einen Überblick, in wie weit die von ReqMan vorgeschlagenen Praktiken im XP Anwendung finden. Die geklammerten Kreuze deuten auf eine nur eingeschränkte Abbildung der Praktiken auf XP-Techniken hin, was bedeutet, dass das Ziel der Praktik nur eingeschränkt durch die entsprechende Technik gewährleistet wird.

	Kunde vor Ort	Planungsspiel	Metapher	Kurze Releasezyklen	Testen	Einfaches Design	Refactoring	Programmieren in Paaren	Programmierstandards
Stakeholder integrieren	(X)								
Stakeholder und Quellen identifizieren									
Funktionale Anforderungen bestimmen	X	X	X						
Ziele erheben									
Scope festlegen	X	X							
Nichtfunktionale Anforderungen festlegen	X								
Machbarkeit prüfen				(X)					
Kontext berücksichtigen									
Anforderungen gliedern		X							
Mehrperspektivenmodellierung									
Verhalten modellieren									
Entwickleranforderungen dokumentieren									
Kundenanforderungen dokumentieren									
Eindeutige Bezeichner benutzen									
Standards und Dokumentenstruktur benutzen									(X)
Anforderungen messbar und testbar beschreiben					X				
GUI spezifizieren									
Anforderungen reviewen	X							X	
Tests für FRs vorbereiten					X				
Bedienbarkeit testen	X								
Tests für NFRs vorbereiten					X				
Technologie wählen									
Verfolgbarkeit einführen									
Veränderungen handhaben	X	(X)		(X)		(X)	(X)		
Rollen und Verantwortlichkeiten festlegen									
Kosten schätzen		X							
Zeit abschätzen		X							
Risikoabschätzung		X		(X)					
Produkt planen		X							
Release planen		X		(X)					
Inhalte festlegen									
Anforderungsprozess verbessern									

Tabelle 2: Abbildung von ReqMan-Praktiken auf XP-Techniken

Wie deutlich zu sehen, werden von XP insbesondere nicht die ReqMan-Praktiken unterstützt, welche sich mit der Dokumentation und Verfolgbarkeit

von Anforderungen beschäftigen. An die Stelle von Spezifikationen treten informelle Nutzergeschichten und mündlich formulierte Anforderungen. Auch eine Modellierung spezieller Sachverhalte ist nicht vorgesehen.

Die Probleme, die sich hieraus ergeben, sind erhebliche Schwierigkeiten bei späteren Änderungen, da außer Code und Testfällen keine Softwareprodukte vorliegen. Auch können getroffene Entscheidungen und mögliche Alternativen im Nachhinein nur noch schwer nachvollzogen werden.

Daneben erfolgt die Planung von Projekten und –iterationen nur unsystematisch auf implizitem Erfahrungswissen der Mitarbeiter. XP geht weiterhin davon aus, dass es nur einen kundenseitigen Stakeholder gibt und somit nicht unterschiedliche Interessensgruppen berücksichtigt werden müssen [Lip02]. Auch hat sich in Fallstudien bereits gezeigt, dass der „Kunde vor Ort“ nicht immer repräsentativ für die späteren Mitarbeiter ist, beziehungsweise sich im Laufe des Projektfortschritts immer mehr von diesen entfernt [Lip02].

Aus diesen Gründen ergänzen Arbeiten wie beispielsweise [Lip02] oder [Naw02] das ursprüngliche XP um Lessons Learned oder gar spezielle Modifikationen. So konnte [Naw02] beispielsweise ohne die Leichtgewichtigkeit der Methodik aufzugeben durch nur drei geringfügige Änderungen die Qualität des Requirements Engineering in XP gemäß der Sommerville-Sawyer-Skala [Som97] deutlich steigern.

Im ursprünglichen XP lassen sich hingegen nur unzureichend Praktiken des Requirements Engineering wiederfinden. Insbesondere solche, die der langfristigen Wartbarkeit dienen, fehlen.

5 Zusammenfassung und Ausblick

Dieser Bericht hat einen kurzen Überblick über die Aktivitäten beziehungsweise Techniken im Rational Unified Process und eXtremeProgramming gegeben. Der Fokus lag auf der Gegenüberstellung der ReqMan-Praktiken und der Techniken der beiden Methoden.

Es hat sich gezeigt, dass der RUP alle wichtigen ReqMan-Praktiken abdeckt und somit prinzipiell ähnliche Vorstellungen von „gutem“ Requirements Engineering besitzt. Durch die kontextspezifischen Instanziierungen und gegebenenfalls individuelle Ausgestaltung einzelner Aktivitäten kann allerdings keine Aussage darüber getroffen werden, wie erfolgreich die tatsächlichen Anforderungsprozesse in Unternehmen sind, welche den RUP verwenden. Hier bietet ReqMan durch Erfassung der IST-Situation und Verbesserungen durch die Einbeziehung adäquater Praktiken und Techniken, Unterstützung. Die im Rahmen dieses Berichts dargestellte Abbildungstabelle kann dabei verwendet werden, um betroffene Stellen (Aktivitäten) im Prozess leichter zu identifizieren.

Anders zeigt sich die Betrachtung des eXtremeProgramming als Vertreter der agilen Entwicklungsmethoden. Hier hat die Literaturrecherche ergeben, dass wichtige Aktivitäten des Requirements Engineering grundsätzlich nur sehr unzureichend Anwendung finden. Insbesondere Praktiken, die sich auf die Dokumentation und Verfolgbarkeit von Anforderungen und die systematische Einbeziehung von kundenseitigen Stakeholdern beziehen, werden im ursprünglichen XP-Ansatz nicht beachtet.

Die agile Philosophie hinter diesen Ansätzen erlaubt es anders als in „klassischen“ Entwicklungsmethoden jedoch nicht, einfach geeignete Praktiken in den Prozess zu integrieren.

Vielmehr sind analog zu Ansätzen wie [Naw02], welche versuchen, durch Modifikation einiger XP-Techniken die Qualität des XP-Anforderungsprozesses signifikant zu steigern, spezielle Strategien für das ReqMan-Framework im Umfeld agiler Methoden zu entwickeln. Dies wird insbesondere dadurch motiviert, dass in den von ReqMan adressierten (kleineren) Unternehmen oft solch agile und wenig strukturierte Entwicklungsmethoden gelebt werden [ODK05].

Als weitere Arbeit ist diesbezüglich zu prüfen, ob im Umfeld agiler Prozesse die ReqMan-Praktiken modifiziert beziehungsweise alternativ klassifiziert werden müssen oder ob die spezielle Anpassung ausschließlich auf Basis der kontextspezifischen Techniken erfolgen kann.

Literatur

- [Bec00] Beck, K.: Extreme Programming – das Manifest. Addison-Wesley, München, 2000
- [DKO04] Dörr, J., König, T., Olsson T: Das ReqMan Prozessrahmenwerk. Draft. 2004
- [Ess03] Essigkrug, A., Mey., T: Rational Unified Process kompakt. Spektrum Akademischer Verlag, Heidelberg, 2003
- [IBM05] IBM Rational Software, <http://www-306.ibm.com/software/rational/>. Besucht: 15.12.05
- [Kru00] Kruchten, P.: The Rational Unified Process An Introduction. Addison-Wesley, Reading Massachusetts, 2000
- [Lip02] Lippert, M., Roock, S., Wolf, H.: Software entwickeln mit eXtreme Programming. dpunkt.verlag, Heidelberg, 2002
- [Naw02] Nawrocko, J., Jasinski, M., Walter, B., Wojciechowski, A.: Extreme Programming Modified: Embrace Requirements Engineering Practices. Proceedings of the IEEE Joint International Conference on Requirements Engineering, 2002
- [ODK05] Olsson, T., Dörr, J., König, T., Ehresmann, M.: A Flexible and Pragmatic Requirements Engineering Framework for SME. Proceedings of SREP'05, Paris, 2005
- [Som97] Sommerville, I., Sawyer, P: Requirements Engineering: A Good Practice Guide. John Wiley & Sons, Chichester, 1997

Dokumenten Information

Titel: ReqMan-Praktiken im RUP
und XP

Datum: 20. Januar 2006

Report: IESE-018.06/D

Status: Final

Klassifikation: Öffentlich

Copyright 2006, Fraunhofer IESE.

Alle Rechte vorbehalten. Diese Veröffentlichung darf für kommerzielle Zwecke ohne vorherige schriftliche Erlaubnis des Herausgebers in keiner Weise, auch nicht auszugsweise, insbesondere elektronisch oder mechanisch, als Fotokopie oder als Aufnahme oder sonstwie vervielfältigt, gespeichert oder übertragen werden. Eine schriftliche Genehmigung ist nicht erforderlich für die Vervielfältigung oder Verteilung der Veröffentlichung von bzw. an Personen zu privaten Zwecken.