

2.

Auflage



Johannes Bergsmann

# Requirements Engineering

## für die agile Softwareentwicklung

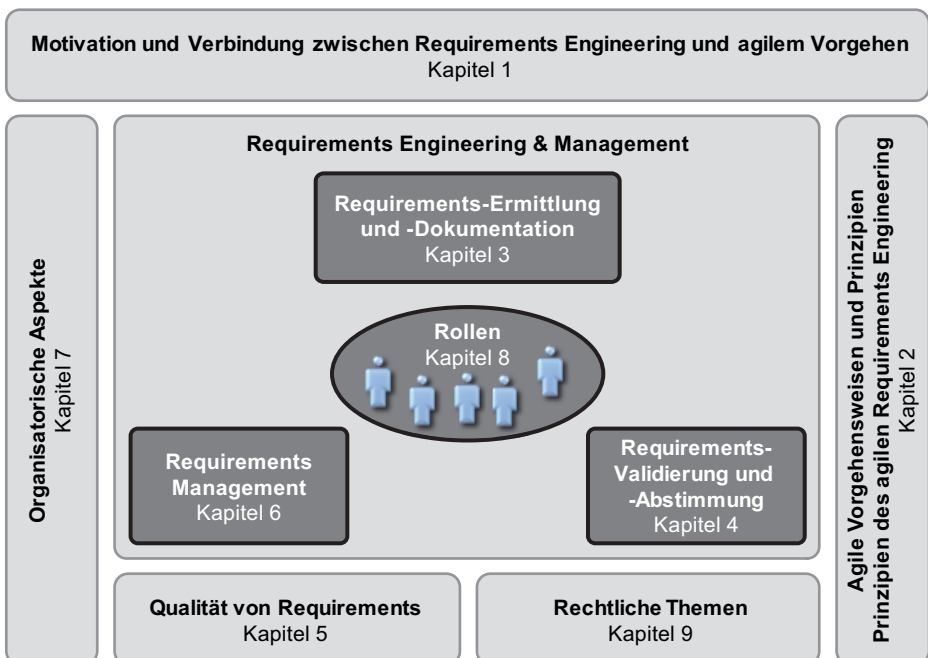
Methoden, Techniken und Strategien

→ Unter Mitwirkung von Markus Unterauer



dpunkt.verlag

# Requirements Engineering für die agile Softwareentwicklung





**Johannes Bergsmann** hat technische Informatik studiert und arbeitete ca. 11 Jahre als Softwareentwickler, Projektleiter, Technischer Leiter, Architekt, Produktmanager und Berater in einem internationalen Systemhaus und als selbstständiger Unternehmer. Im März 2003 gründete er »Software Quality Lab« und begleitet seither als Berater und Trainer viele Unternehmen im Bereich Requirements Engineering und Prozessgestaltung.

Johannes Bergsmann ist zertifizierter Scrum Master, Sachverständiger für Informatik bei Gerichten, als Lehrbeauftragter an Fachhochschulen im Bereich Softwarequalitätsmanagement tätig, ist Autor vieler Fachartikel und hält Fachvorträge bei verschiedenen Veranstaltungen und Konferenzen.

#### **Unter Mitwirkung von Markus Unterauer:**



**Markus Unterauer** hat Wirtschaftsinformatik studiert. In seiner Berufspraxis war er in vielen Bereichen der Softwareentwicklung wie Architektur, Entwurf, Entwicklung, Testen, Testautomatisierung bis zu Deployment tätig. Er lernte dabei sowohl klassische als auch agile Projekte und Methoden intensiv kennen.

Seit 2012 arbeitet Markus Unterauer bei Software Quality Lab als Berater und Trainer. Er ist zertifizierter Scrum Master und hat sich auf die Bereiche Softwareprozesse und Anforderungsmanagement spezialisiert. Markus Unterauer ist auch als Vortragender in diesen Themenbereichen immer wieder auf Konferenzen tätig.

**Johannes Bergsmann**

# **Requirements Engineering für die agile Softwareentwicklung**

**Methoden, Techniken und Strategien**

Unter Mitwirkung von Markus Unterauer

2., überarbeitete und aktualisierte Auflage



**dpunkt.verlag**

Johannes Bergsmann  
johannes.bergsmann@software-quality-lab.com

Markus Unterauer  
markus.unterauer@software-quality-lab.com

Lektorat: Christa Preisendanz  
Copy-Editing: Ursula Zimpfer, Herrenberg  
Satz: Birgit Bäuerlein  
Herstellung: Susanne Bröckelmann  
Umschlaggestaltung: Helmut Kraus, [www.exclam.de](http://www.exclam.de)  
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;  
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:  
Print 978-3-86490-485-1  
PDF 978-3-96088-189-6  
ePub 978-3-96088-190-2  
mobi 978-3-96088-191-9

2., überarbeitete und aktualisierte Auflage 2018  
Copyright © 2018 dpunkt.verlag GmbH  
Wieblinger Weg 17  
69123 Heidelberg

Teile dieses Buches orientieren sich am Lehrplan RE@Agile des IREB e.V. Der Besitz und das Urheberrecht dieses Lehrplans und Studienleitfadens liegt bei IREB e.V. und den Autoren: Lars Baumann, Peter Hruschka, Kim Lauenroth, Markus Meuten, Sacha Reis, Gareth Rogers, Francois Salazar, Thorsten Weyer.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

---

## Vorwort zur 2. Auflage

Als Berater habe ich in meiner bisherigen beruflichen Tätigkeit mehr als 200 verschiedene Projekte bei Kunden begleitet oder auch selbst in unterschiedlichen Rollen (Entwickler, Tester, Projektleiter, Architekt, Produktmanager, Analytiker, Coach, Berater etc.) mitgearbeitet. Viele dieser Projekte – vor allem in den letzten 15 Jahren – waren agile Projekte oder Projekte, in denen die Mitwirkenden zumindest versuchten, die agilen Prinzipien und Praktiken umzusetzen. Am erfolgreichsten und effizientesten waren fast immer diejenigen Projekte, bei denen agile Vorgehensweisen mit Elementen aus dem klassischen Requirements Engineering und Projektmanagement ergänzt und so die Stärken beider Bereiche genutzt wurden.

Requirements Engineering ist ein Aspekt, der in vielen agilen Vorgehensweisen nur sehr grob beschrieben wird. In der Begeisterung der ersten Stunde möchten viele Teams möglichst rasch starten und erste Erfolge zeigen und beginnen mit einem sehr intuitiven Ansatz, wie z. B. einer einfachen Liste von User Stories. User Stories sind eine sinnvolle Technik und ich baue in vielen Abschnitten dieses Buches auf dieser Technik auf. Es gibt jedoch noch viele andere Techniken aus dem Requirements Engineering, die in agilen Vorgehensweisen angewendet werden können und sollen. In Kombination mit einigen passenden, langjährig bewährten Techniken des Requirements Engineering bieten sie gute Lösungen für die Requirements-Herausforderungen in agilen Projekten.

Aus diesem Grund entschloss ich mich auch dazu, im Jahr 2014 die erste Auflage dieses Buches zu schreiben. Darin werden viele verschiedene Techniken und Methoden aus den agilen Vorgehensweisen vorgestellt, mit klassischen Methoden verknüpft und ergänzt und in einen strukturierten und systematischen Zusammenhang mit Requirements Engineering gebracht.

Mittlerweile sind einige Jahre vergangen und es wurde 2017 ein Lehrplan RE@Agile für eine Ausbildung des International Requirements Engineering Board (IREB<sup>®</sup>) herausgegeben, der die Lücke zwischen Requirements-Engineering-Methoden und agilen Methoden schließt. Sehr viele Inhalte dieses Lehrplans decken sich bereits mit der ersten Auflage dieses Buches und die Motivation hinter diesem Lehrplan ist dieselbe, wie sie hinter meinem Buch steht.

Daher waren eine Anpassung und Überarbeitung der ersten Auflage ein logischer und sinnvoller Schritt zur zweiten Auflage dieses Buches. Die Struktur und die Inhalte dieses Buches sind an den offiziellen Lehrplan des International Requirements Engineering Board (IREB<sup>®</sup>) angepasst. Damit bietet es den an dieser offiziellen Ausbildung und Zertifizierung Interessierten nun eine praxisnahe Vorbereitung und Lerngrundlage sowie noch viele ergänzende Inhalte, die über den Lehrplan hinausgehen.

Im Wesentlichen wurden folgende Abschnitte aufgrund des neuen IREB<sup>®</sup>-Lehrplans umfangreicher angepasst: Abschnitte 1.1 und 1.2, 2.1 und 2.2 sowie 3.1. Neu hinzugekommen ist Kapitel 7. Alle anderen Kapitel wurden durchgesehen und teilweise mit kleineren Überarbeitungen und Ergänzungen an aktuelle Erfordernisse angepasst und erkannte Fehler bereinigt. Zusätzlich zu inhaltlichen Anpassungen erfolgte auch eine grundlegende Umstellung der Reihenfolge der Kapitel, die sich nun ebenfalls weitgehend am Lehrplan des IREB<sup>®</sup> orientiert, sowie eine Umbenennung einzelner Kapitelüberschriften.

Ich freue mich, wenn dieses Buch für Sie als Anwender oder Experte in Ihrer täglichen Praxis Unterstützung und Anregung bietet und für Sie als Interessenten an der Schulung und Zertifizierung RE@Agile eine gute Informations- und Lerngrundlage ist.

Für die Weiterentwicklung dieses Buches hoffe ich auch auf zahlreiches Feedback und interessante Diskussionen (bitte direkt an [johannes@bergsmann.com](mailto:johannes@bergsmann.com)).

*Johannes Bergsmann*  
Linz, im Februar 2018

## Vorwort

Viele Projekte werden aus verschiedenen Gründen nicht so effizient und effektiv abgewickelt, wie dies möglich wäre: Zum Beispiel wenn die Fachexperten zwar wissen, dass sie mit den Entwicklern täglich kommunizieren sollten, dies jedoch nicht können,

- weil sie im Tagesgeschäft schon zu stark involviert und überlastet sind,
- weil sie von ihrer Persönlichkeit her keine aktiv kommunizierenden Typen sind,
- weil sie mitten im Projekt andere Aufgaben zugewiesen bekommen,
- weil sie die Abteilung oder Firma wechseln,
- weil es schwierig ist, das Wissen in den Köpfen der Beteiligten bewusst zu machen und an andere zu kommunizieren,
- weil Kommunikation zwischen zwei Personen immer auch mit einer Interpretation und eventuell mit einer Veränderung der Information einhergeht oder
- aus verschiedensten anderen Gründen.

Als Berater habe ich in meiner bisherigen beruflichen Tätigkeit mehr als 200 verschiedene Projekte bei Kunden begleitet oder auch selbst in unterschiedlichen Rollen (Entwickler, Tester, Projektleiter, Architekt, Produktmanager, Analytiker, Coach, Berater etc.) mitgearbeitet. Viele dieser Projekte – vor allem in den letzten 10 Jahren – waren agile Projekte oder Projekte, in denen die Mitwirkenden zumindest versuchten, Teile der agilen Prinzipien umzusetzen.

Am erfolgreichsten und effizientesten waren in dieser langen Zeit immer diejenigen Projekte, bei denen ich agile Vorgehensweisen mit Elementen aus dem klassischen Requirements Engineering und Projektmanagement ergänzte und so die Stärken jeder Methodik voll ausnutzen konnte.

Man könnte alle diese Softwareprojekte mit einer Autofahrt von München nach Rom vergleichen. Im Idealfall fahren wir auf gerader Strecke mit konstanter und optimaler Geschwindigkeit mit unserem Auto alleine auf der Straße. In der Praxis aber hat die Straße Kurven, es gibt Verkehrsbeschränkungen, in den Bergen ist eventuell auch Eis auf der Straße, es gibt Staus, andere Autofahrer, die



rücksichtslos unterwegs sind und nur ihr eigenes Ziel im Blick haben, unser Auto hat eine Panne etc.

Auf alle diese individuellen Situationen sollten wir vorbereitet sein und unser Vorgehen den jeweiligen Situationen entsprechend anpassen, damit wir unser Ziel auch erreichen. Generell werden wir grob planend vorgehen, z.B. den Startzeitpunkt bestimmen, den ungefähren Zeitrahmen der Fahrt abschätzen und den Streckenverlauf z.B. für die Alpenquerung über den Brenner planen. Wir sollten auch ungefähr wissen, welches Wetter zu erwarten ist, und eventuell die Reifen, Frostschutz, Klimaanlage etc. entsprechend vorbereiten. Im Verlauf der Fahrt wird es vielleicht zu Änderungen kommen, z.B. wenn der Brennertunnel wegen eines Unfalls gesperrt ist. In diesem Fall werden wir agil darauf reagieren müssen und die Umleitungsstrecke über den Pass nehmen (schließlich haben wir ja Google Maps dabei ;-). Wenn wir im Vorfeld in der »Architektur« unseres Autos diese Situation mangels vorausschauender Planung nicht berücksichtigt haben und beim ersten kurzen Anstieg der Passstraße feststellen, dass wir ohne Ketten auf der verschneiten Straße nicht weiterkommen, müssen wir wiederum agil reagieren und nun eventuell einen großen Umweg über die Tauernautobahn nehmen. Beide Vorgehensweisen haben daher ihre Berechtigung. Als Softwareentwickler, Projektleiter, Agile Master, Product Owner – oder welche Rolle auch immer wir im Projekt inne haben – sollten wir viele unterschiedliche Methoden im Köcher haben und diese für unser Projekt zur optimalen Vorgehensweise kombinieren.

## Das Agile Manifest

Das Agile Manifest [Agile Manifesto] (siehe Abschnitt 1.2) beschreibt in seinen vier agilen Leitsätzen und zwölf Prinzipien die Eckpfeiler, an denen sich praktisch alle agilen Vorgehensweisen orientieren. Darin wird unter anderem festgehalten, dass funktionierende Software und Zusammenarbeit mit dem Kunden wichtiger sind als umfassende Dokumentation und Vertragsverhandlungen, wobei im Zusatz angeführt ist, dass umfassende Dokumentation und Vertragsverhandlungen auch als wichtig angesehen werden. So sollten Projekte idealerweise ablaufen. Wenn man die Praxis in vielen Softwareprojekten erlebt hat, wird man dem Agilen Manifest begeistert zustimmen und die Aussage und Sichtweise uneingeschränkt unterstützen.

Ausgehend vom Agilen Manifest wurden verschiedene Vorgehensweisen entwickelt, z.B. Extreme Programming (XP) und Scrum. Diese Vorgehensweisen sind keine umfassenden Entwicklungsmodelle, sondern greifen bestimmte Aspekte und Themen auf und lassen andere Themen bewusst offen und unkonkret. Dies ermöglicht es und fordert gleichzeitig jedes Projektteam dazu auf, den konkreten Weg unter Berücksichtigung der Projektsituation und Rahmenbedingungen selbst zu finden und festzulegen.

Requirements Engineering ist ein Aspekt, der in vielen agilen Vorgehensmodellen nur sehr grob beschrieben wird. Aus diesem Grund habe ich mich auch dazu entschlossen, dieses Buch zu schreiben.

## Wenn Requirements zum Problem werden

Sehr oft wurde und wird in verschiedenen Stadien eines Projekts das Requirements Engineering und Requirements Management zum Thema, z.B. wenn ...

- ... Juristen oder Einkäufer eines externen Kunden, der schon bei Projektstart einen Festpreis vereinbaren will, vorab wissen wollen, was denn schlussendlich für den vereinbarten Preis geliefert wird.
- ... bei der Abnahme eines Produkts durch den Kunden dann plötzlich wichtige Personen fehlende Funktionen bemängeln und nicht klar ist, ob dies vom Lieferanten noch als Auftragsbestandteil geschuldet wird oder ob das nun ein kostenpflichtiger Change Request ist.
- ... im Laufe des Projekts eine auf Basis von User Stories (siehe Abschnitt 3.4.2) entwickelte Maske schon zum x-ten Mal über mehrere Iterationen hinweg wieder und wieder angepasst und verändert wird, weil der Kunde sich immer wieder etwas anderes als Ergebnis vorstellt.
- ... Entwicklungsverantwortliche, die auf die Frage »Was kann denn das Produkt, das Sie entwickeln, nun eigentlich alles?« nur sagen können, dass sie Tausende User Stories in ihrem Request-Tool dokumentiert haben und dort nachzulesen ist, was umgesetzt wurde.
- ... das Projektteam zwar weiß, was es die letzten zwei, drei Iterationen entwickelt hat und was es die kommenden zwei Iterationen entwickeln wird, jedoch keinen Überblick mehr darüber hat, welche Funktionalität das Produkt insgesamt hat.
- ... »der Wald vor lauter Bäumen nicht mehr erkannt« wird und die Zusammenhänge für die Beteiligten möglicherweise schon verloren gegangen sind.

Alle diese geschilderten Fälle sind primär auf mangelndes Requirements Engineering und Requirements Management zurückzuführen. Tendenziell treten solche Probleme in Projekten auf, in denen ein Vorgehensmodell gewählt wurde, das viele thematische Freiheiten bietet, die offengelassenen Teile aber nicht vorab zwischen den beteiligten Personen festgelegt wurden oder das Modell nicht an die gegebene Situation angepasst wurde.

Gerade agile Vorgehensweisen überlassen die konkrete Ausgestaltung des Requirements Engineering zum Großteil der Entscheidung des Teams. In der Begeisterung der ersten Stunde möchten viele Teams möglichst rasch starten und erste Erfolge zeigen und beginnen mit einem sehr intuitiven Ansatz, wie z.B. einer einfachen Liste von User Stories. User Stories sind eine sinnvolle Technik, und ich baue in vielen Abschnitten dieses Buches auf dieser Technik auf. Es gibt jedoch

noch viele andere Techniken aus dem Requirements Engineering, die in agilen Vorgehensweisen angewendet werden können und sollen. In Kombination mit einigen passenden klassischen Techniken des Requirements Engineering bieten sie gute Lösungen für die Requirements-Herausforderungen in agilen Projekten.

In diesem Buch werden daher viele verschiedene Techniken und Methoden aus den agilen Vorgehensweisen vorgestellt, mit klassischen Methoden verknüpft und ergänzt und in einen strukturierten und systematischen Zusammenhang mit Requirements Engineering gebracht.

## Über dieses Buch

Der Fokus dieses Buches liegt darauf, gute Methoden und Techniken – egal aus welchem Zeitalter oder mit welcher Ausrichtung – unvoreingenommen aufzugreifen und sie nicht von vornherein auszuschließen, nur weil es sich dabei um »klassische« oder »agile« Methoden oder Techniken handelt. Aussagen wie »klassische Methoden sind überfrachtet, ineffizient und schlecht« oder »agile Methoden sind was für Individualisten oder Dokumentationsfeinde« gehören für mich nicht zu einer weitsichtigen und nachhaltigen Denkweise.

Jede genannte Methode und Technik stellt ein Werkzeug im Werkzeugkasten der Softwareentwicklung dar und soll hier mit ihren Vorteilen und Nachteilen in bestimmten Projektsituationen betrachtet werden. Es soll klar werden, wo und wie diese Methoden und Techniken in agilen Projekten eingesetzt werden können, um einen Nutzen für das Projekt zu stiften. Es werden daher auch sinnvolle Anwendungsmöglichkeiten und Fallstricke der einzelnen Methoden und Techniken dargestellt und im Kontext eines nachhaltigen, systematischen und praxisorientierten Requirements Engineering in der agilen Softwareentwicklung betrachtet.

**Ziel ist es, einen Überblick und einen Werkzeugkasten anzubieten, der zeigt, welche Methoden und Techniken zusätzlich zu den sehr oft angewendeten User Stories noch sinnvoll sind und welche Hindernisse und Problemstellungen im Zusammenhang mit Requirements Engineering in agilen Projekten auftreten können.**

Abgerundet wird das Thema durch die Behandlung von Qualitätsaspekten für Requirements, durch einen Blick auf die Zusammenhänge und durch rechtliche Aspekte sowie durch Tipps und Tricks bei der Anwendung von Requirements-Themen im agilen Umfeld.

**Das Buch richtet sich an Product Owner, Produktmanager, Projektmanager, Softwareauftraggeber, Scrum Master, Entwickler, Tester und alle anderen Personen, die sich mit nachhaltigem und systematischem Requirements Engineering und Requirements Management in der agilen Softwareentwicklung beschäftigen oder davon betroffen sind.**

Dieses Buch ist *keine* komplette Einführung in agile Methoden. Kapitel 1 gibt zwar einen kurzen Überblick über das Agile Manifest, wesentliche agile Konzepte und Scrum, als am häufigsten eingesetzten Vertreter agiler Methoden. Der Hauptfokus des Buches liegt jedoch im Requirements Engineering und Requirements Management.

Hilfreich beim Lesen ist ein grundlegendes Verständnis des aktuellen Stands im Bereich Requirements Engineering und insbesondere die Kenntnis der allgemein anerkannten Begriffe, Techniken und Grundlagen aus diesem Themenbereich. Der Inhalt und die Begriffe, die das International Requirements Engineering Board [IREB] für die Ausbildung zum »Certified Professional for Requirements Engineering (CPRE) – Foundation Level« in seinem Lehrplan [IREB CPRE FL 2012] vorgibt, stellen den grundlegenden Stand der Technik zum Thema Requirements Engineering dar, der als Basiswissen für das Lesen dieses Buches empfohlen wird.

Das Buch soll Einführungslektüre sowie Praxisleitfaden sein und ist nicht als wissenschaftlich komplette Abhandlung über das Thema Requirements Engineering in der agilen Softwareentwicklung gedacht. Es kann daher sequenziell oder auch auszugsweise gelesen werden. Als Autor ist es mir natürlich am liebsten, wenn Sie dieses Buch ständig an Ihrem Arbeitsplatz griffbereit haben und bei Fragen oder Unklarheiten zu Requirements-Engineering-Techniken oder -Vorgehensweisen darin einen schnellen Rat und brauchbare Infos finden.

In der Praxis hat man leider oft nicht die Zeit, beim Auftauchen einer Frage ein ganzes Buch z.B. zu Use Cases, User Stories oder Behaviour Driven Development zu lesen. Ich habe daher auf eine möglichst große Unabhängigkeit der einzelnen Themen und Kapitel geachtet, sodass das Buch auch als Nachschlagewerk in der täglichen Arbeit verwendet werden kann. Mit entsprechendem Vorwissen in agilen Methoden und Requirements Engineering kann es auch auszugsweise gelesen und verstanden werden. Für Praktiker müssen das Wichtigste und die Zusammenhänge in Kürze ersichtlich sein. Daher gibt es in vielen Kapiteln eine tabellarische Übersicht über wesentliche Punkte und verschiedene Überblicksgrafiken, die die Zusammenhänge auf einen Blick darstellen.

Auch wenn in vielen Kapiteln Begriffe aus Scrum verwendet oder zitiert werden, so wurde darauf geachtet, die einzelnen Themenbereiche möglichst allgemeingültig zu halten. Das Buch adressiert nicht »Requirements Engineering in Scrum«, sondern behandelt generell das Thema Requirements Engineering im agilen Umfeld. Die beschriebenen Techniken und Methoden können auch in anderen agilen Vorgehensweisen angewendet werden.

**Beispiele werden mit einem grauen Hintergrund versehen:**

In diesem Buch werden verschiedene Beispiele und Formulierungen aus der Praxis und für die Praxis aufgeführt. Die meisten dieser Beispiele sind abgeleitet aus einem Projekt zum Thema »Zeiterfassung«. Es geht in diesem Beispielszenario darum, dass die Tagesarbeitszeiten inklusive der Pausen und Abwesenheitszeiten von Mitarbeitern eines Unternehmens erfasst, ausgewertet und verwaltet werden können und dass auch die Erfassung, Auswertung und Verwaltung von Projektarbeitszeiten durchgeführt werden kann – also von einzelnen Zeitblöcken der Tagesarbeitszeit, die bestimmten Projekten zugeordnet werden.

Weiterführende Literatur zum Thema Requirements und agile Vorgehensweisen ist im Anhang D zu finden.

Ich freue mich, wenn dieses Buch für Sie als Anwender oder Experte in Ihrer täglichen Praxis eine Unterstützung und Anregung ist. Für die Weiterentwicklung dieses Themas hoffe ich natürlich auch auf zahlreiches Feedback und interessante Diskussionen (bitte direkt an [johannes.bergsmann@software-quality-lab.com](mailto:johannes.bergsmann@software-quality-lab.com)).

*Johannes Bergsmann*  
Linz, im April 2014

## Danksagung

Mein großer Dank gilt Markus Unterauer, der mich als Koautor bei der Erstellung dieses Buches maßgeblich unterstützt hat. Markus hat primär die Kapitel 4 »Requirements-Validierung und -Abstimmung« und Kapitel 8 »Requirements-Engineering-Rollen« federführend ausgearbeitet und zum Kapitel 6 »Requirements Management« wesentliche Teile beigetragen sowie beim Review und Feinschliff des gesamten Buches viele gute Anregungen gegeben.

Des Weiteren danke ich allen Experten und Personen, mit denen ich im Laufe der letzten Jahre zum Thema Requirements Engineering und agile Methoden diskutieren durfte und die schlussendlich dazu beigetragen haben, dass ich mich zum Schreiben dieses Buches entschlossen habe und entsprechende Sichtweisen aus der Praxis mit einfließen lassen konnte.

Vielen Dank auch dem dpunkt.verlag – insbesondere Christa Preisendanz – für die initiale Anregung und für die wertvolle Unterstützung im Laufe der Ausarbeitung dieses Buches.

Und schlussendlich gilt ein ganz großer Dank meiner Frau Petra und meinen Kindern Beate und Barbara, die mich dadurch unterstützt haben, dass sie mir die Zeit zum Schreiben dieses Buches gegeben haben.



---

# Inhaltsübersicht

<b>1</b>	<b>Einleitung und Motivation</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>25</b>
<b>3</b>	<b>Requirements-Ermittlung und -Dokumentation</b>	<b>53</b>
<b>4</b>	<b>Requirements-Validierung und -Abstimmung</b>	<b>177</b>
<b>5</b>	<b>Qualität von Requirements</b>	<b>209</b>
<b>6</b>	<b>Requirements Management</b>	<b>227</b>
<b>7</b>	<b>Organisatorische Aspekte</b>	<b>259</b>
<b>8</b>	<b>Requirements-Engineering-Rollen</b>	<b>277</b>
<b>9</b>	<b>Rechtliche Themen</b>	<b>291</b>

## Anhang

<b>A</b>	<b>Agile Methoden zur Unterstützung des Requirements Engineering</b>	<b>321</b>
<b>B</b>	<b>Abkürzungen</b>	<b>339</b>
<b>C</b>	<b>Glossar</b>	<b>341</b>
<b>D</b>	<b>Literatur</b>	<b>347</b>
	<b>Index</b>	<b>355</b>





# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Motivation</b>	<b>1</b>
1.1	Über dieses Buch .....	1
1.1.1	Zielgruppen .....	2
1.1.2	Abbildung des Lehrplans IREB RE@Agile .....	3
1.1.3	Allgemeine Begriffseinordnung .....	4
1.2	Verbindung zwischen Requirements Engineering und agilem Vorgehen .....	6
1.2.1	Mindsets und Werte im Requirements Engineering und agilem Vorgehen .....	7
1.2.2	Requirements Engineering im Kontext des Agilen Manifests .....	12
1.2.3	Nutzen von Requirements Engineering im agilen Vorgehen .....	17
1.2.4	Vorurteile und Probleme beim Requirements Engineering im agilen Umfeld .....	18
1.2.5	Fallstricke bei RE@Agile .....	20
1.2.6	Resümee .....	23
<b>2</b>	<b>Grundlagen</b>	<b>25</b>
2.1	Methodenüberblick .....	25
2.1.1	Allgemeine agile Vorgehensweisen .....	25
2.1.2	Scrum »in a Nutshell« .....	27
2.1.3	Methoden zur Unterstützung des Requirements Engineering .....	32
2.2	Requirements Engineering im agilen Umfeld .....	39
2.3	Die fünf Grundprinzipien des Requirements Engineering in der agilen Softwareentwicklung .....	42
2.4	Umfang des Requirements Engineering .....	49

<b>3</b>	<b>Requirements-Ermittlung und -Dokumentation</b>	<b>53</b>
3.1	Allgemeines	53
3.1.1	Requirements	53
3.1.2	Ermittlung	54
3.1.3	Dokumentation	55
3.1.4	Spezifikationsdokumente vs. Product Backlog	58
3.1.5	Granularität funktionaler Requirements	59
3.1.6	Grafische Modelle und textuelle Beschreibungen	61
3.1.7	Artefakte	62
3.1.8	Definition von Begriffen, Glossare und Informationsmodelle	68
3.1.9	Akzeptanz- und Abnahmekriterien	68
3.2	Übergeordnete Artefakte	70
3.2.1	Vision und Goals (Ziele)	70
3.2.2	Epics	75
3.2.3	Kontextmodelle	80
3.2.4	Stakeholder-Liste	83
3.2.5	Personas	86
3.3	Geschäftsprozesse und Systemverhalten	88
3.3.1	Prozesse	88
3.3.2	Use Cases	96
3.3.3	Use-Case-Szenario bzw. -Template	103
3.4	Funktionale und nicht funktionale Sicht	112
3.4.1	Features	112
3.4.2	User Stories	115
3.4.3	Nicht funktionale Anforderungen (NFA)	121
	3.4.3.1 Qualitätsanforderungen	122
	3.4.3.2 Randbedingungen (Constraints)	133
3.5	Benutzerschnittstelle	136
3.5.1	Wireframes	140
3.5.2	Sketchy UI /Sketches	141
3.5.3	Finales UI	142
3.5.4	Szenariobasierte UI-Spezifikation	145
3.5.5	Hinweise zur GUI-Spezifikation	147
3.6	Systemschnittstelle	149
3.7	Prototypen und Inkremente	152
3.8	Entwicklersicht	154
3.8.1	Spikes	154
3.8.2	Architektur und technisches Design	158
3.8.3	Developer Story	162

3.8.4	System szenarien	165
3.8.5	Developer Constraints	167
3.8.6	Tasks	172
3.9	Inhaltliche Strukturierungshilfsmittel	174
3.9.1	Themen	174
3.9.2	Epics und Features	176
<b>4</b>	<b>Requirements-Validierung und -Abstimmung</b>	<b>177</b>
4.1	Verfeinerung von Anforderungen	179
4.1.1	Backlog Refinement	179
4.1.2	Refinement-Meeting	181
4.2	Machbarkeitsanalyse	183
4.2.1	Technische und funktionale Analyse mit Spikes	183
4.2.2	Organisatorische und personelle Machbarkeit	183
4.3	Analyse von Nutzen und Geschäftswert	184
4.3.1	Messung des Nutzens	184
4.3.2	Das Kano-Modell	185
4.3.3	Ordnung nach relativem Nutzen	186
4.3.4	Abstrakter Geschäftswert (Business Value)	187
4.4	Risikobewertung	187
4.4.1	Risiken identifizieren und bewerten	188
4.4.2	Maßnahmen planen	192
4.4.3	Risiken überwachen und steuern	194
4.5	Aufwands- und Kostenschätzung	195
4.5.1	Aufwandsschätzung in nicht agilen Projekten	195
4.5.2	Prinzipien agiler Schätzungen	198
4.5.3	Schätzen im Projektverlauf	200
4.5.4	Schätzmethode n	201
4.5.5	Ermitteln von Aufwand und Kosten aus Story Points	203
4.6	Bewertung der Qualität der Anforderungen	206
4.7	Priorisierung	206
4.7.1	Prioritätsskala	207
4.7.2	Basis für die Priorisierung	207
<b>5</b>	<b>Qualität von Requirements</b>	<b>209</b>
5.1	Qualitätskriterien für Requirements	210
5.1.1	Qualitätskriterien nach IEEE 830-1998 und IREB	210
5.1.2	DEEP-Qualitätskriterien	214
5.1.3	INVEST-Qualitätskriterien	215

5.2	Definition of Ready (DoR) .....	216
5.3	Definition of Done (DoD) .....	218
5.4	Review von Requirements .....	224
<b>6</b>	<b>Requirements Management</b>	<b>227</b>
6.1	Inhalt vs. Management des Inhalts .....	228
6.2	Requirements-Management-Aktivitäten .....	231
6.3	Planende Aktivitäten des Requirements Managements .....	231
6.3.1	Portfolio- und Programmplanung .....	232
6.3.2	Systemplanung .....	234
6.3.3	Releaseplanung .....	238
6.3.4	Sprint-Planung .....	240
6.3.5	Daily Meeting .....	242
6.4	Artefakte für das Requirements Management .....	243
6.4.1	Backlog .....	243
6.4.2	Story Maps .....	246
6.4.3	Listenbasierte Requirements-Verwaltung .....	247
6.4.4	Story-Card-basierte Requirements-Verwaltung ...	247
6.4.5	Agiles Requirements-Board .....	249
6.4.6	Taskboard .....	254
<b>7</b>	<b>Organisatorische Aspekte</b>	<b>259</b>
7.1	Einfluss der Organisation .....	259
7.2	Agile Entwicklung im nicht agilen Umfeld .....	260
7.2.1	Interaktion mit Stakeholdern außerhalb der Softwareorganisation .....	260
7.2.2	Produkt- vs. Projektorganisation .....	261
7.2.3	Die Rolle des Managements im agilen Kontext ...	262
7.3	Der Umgang mit komplexen Problemen durch Skalierung ...	264
7.3.1	Motivation für die Skalierung .....	264
7.3.2	Ansätze für das Organisieren von Teams .....	265
7.3.3	Ansätze für das Organisieren der Kommunikation .....	266
7.3.4	Beispiel-Frameworks für das Skalieren von RE@Agile .....	268
7.3.5	Auswirkungen der Skalierung auf RE@Agile .....	269
7.4	Vorab- und kontinuierliche Aufgaben des Requirements Engineering im Zusammenhang mit Skalierung .....	270
7.4.1	Initiale Requirements-Definition .....	271
7.4.2	Detaillierungsgrad für Backlog Items .....	272
7.4.3	Validität von Backlog-Einträgen .....	272

7.4.4	Feedback zum Backlog und dessen Aktualisierung . . . . .	273
7.4.5	Zeitlicher Ablauf des Entwicklungszyklus . . . . .	274
<b>8</b>	<b>Requirements-Engineering-Rollen</b>	<b>277</b>
8.1	Product Owner . . . . .	278
8.1.1	Der Product Owner als Stellvertreter des Kunden im Team . . . . .	278
8.1.2	Schwierige Ausprägungen von Product Ownern ..	279
8.2	Agiles Entwicklungsteam . . . . .	280
8.2.1	Das Entwicklungsteam als Umsetzer und Berater des Product Owners . . . . .	280
8.2.2	Schwierige Ausprägungen im Entwicklungsteam . . . . .	282
8.3	Agile Master . . . . .	283
8.3.1	Der Agile Master als Coach und Problemlöser . . .	283
8.3.2	Schwierige Ausprägungen von Agile Master . . . . .	284
8.4	Tester . . . . .	285
8.4.1	Der Tester als Prüfer und Qualitätsberater . . . . .	285
8.4.2	Schwierige Ausprägungen von Testern . . . . .	286
8.5	Architekt . . . . .	286
8.5.1	Der Architekt als Berater für das Gesamtsystem ..	286
8.5.2	Schwierige Ausprägungen von Architekten . . . . .	287
8.6	Produktmanager . . . . .	288
8.6.1	Der Produktmanager als Dirigent mehrerer Teams . . . . .	288
8.6.2	Schwierige Ausprägungen von Produktmanagern . . . . .	289
<b>9</b>	<b>Rechtliche Themen</b>	<b>291</b>
9.1	Allgemeine rechtliche Aspekte . . . . .	292
9.2	Vertragsbasis und Vertragserfüllungspflicht . . . . .	294
9.3	Gewährleistung . . . . .	301
9.4	Agile Vorgehensweisen und Festpreis . . . . .	303
9.5	Das Vier-Stufen-Modell für agile Festpreisprojekte . . . . .	306
9.5.1	Stufe 1: Definition der Projektziele und ersten Kundenanforderungen . . . . .	306
9.5.2	Stufe 2: Agiles Erstellen der Vertragsbasis . . . . .	307
9.5.3	Stufe 3: Festpreisangebot durch den Lieferanten ..	309
9.5.4	Stufe 4: Agile Projektabwicklung . . . . .	309

9.6	Öffentliche Ausschreibungen .....	311
9.7	Standards und Normen .....	313
9.8	Absicherung des Auftraggebers .....	316
9.9	Absicherung des Lieferanten .....	317

## **Anhang**

<b>A</b>	<b>Agile Methoden zur Unterstützung des Requirements Engineering</b>	<b>321</b>
A.1	Specification by Example .....	321
A.2	Test Driven Development .....	325
A.3	Behaviour Driven Development .....	330
<b>B</b>	<b>Abkürzungen</b>	<b>339</b>
<b>C</b>	<b>Glossar</b>	<b>341</b>
<b>D</b>	<b>Literatur</b>	<b>347</b>
	<b>Index</b>	<b>355</b>

---

# 1 Einleitung und Motivation

## 1.1 Über dieses Buch

Der Fokus dieses Buches liegt darauf, gute Methoden und Techniken – egal welchen Alters oder welcher Ausrichtung – unvoreingenommen aufzugreifen und sie nicht von vornherein auszuschließen, nur weil es sich dabei um »klassische« oder »agile« Methoden oder Techniken handelt. Aussagen wie »nicht agile Methoden sind überfrachtet, ineffizient und schlecht« oder »agile Methoden sind was für Individualisten oder Dokumentationsfeinde« gehören nicht zu einer weitsichtigen und nachhaltigen Denkweise.

Jede der in diesem Buch genannten Methoden und Techniken stellt ein Werkzeug im Werkzeugkasten der Softwareentwicklung dar und soll hier mit ihren Vorteilen und Nachteilen in bestimmten Projektsituationen betrachtet werden. Es soll klar werden, wo und wie diese Methoden und Techniken in agilen Projekten eingesetzt werden können, um einen Nutzen für das Projekt zu stiften. Es werden daher auch sinnvolle Anwendungsmöglichkeiten und Fallstricke der einzelnen Methoden und Techniken dargestellt und im Kontext eines nachhaltigen, systematischen und praxisorientierten Requirements Engineering in der agilen Softwareentwicklung betrachtet.

Abgerundet wird das Buch durch die Behandlung von Qualitätsaspekten für Requirements, organisatorische Aspekte sowie durch rechtliche Aspekte bei der Anwendung von Requirements im agilen Umfeld.

Dieses Buch ist keine komplette Einführung in agile Methoden. Die Kapitel 1 und 2 geben zwar einen kurzen Überblick über das Agile Manifest, wesentliche agile Konzepte und Scrum, als am häufigsten eingesetzten Vertreter agiler Methoden. Der Hauptfokus des Buches liegt jedoch im Requirements Engineering und Requirements Management im agilen Kontext.

Hilfreich beim Lesen ist ein grundlegendes Verständnis des aktuellen Stands im Bereich Requirements Engineering und agiles Vorgehen sowie insbesondere die Kenntnis der allgemein anerkannten Begriffe, Techniken und Grundlagen aus diesen Themenbereichen. Zur vertiefenden Betrachtung wird z.B. auch der Lehrplan und die darin genannten Originalquellen für die Ausbildung zum »Certified



Professional for Requirements Engineering (CPRE)« [IREB CPRE FL 2017] des International Requirements Engineering Board [IREB] empfohlen, der den grundlegenden Stand der Technik zum Thema Requirements Engineering darstellt.

Das Buch soll Einführungslektüre sowie Praxisleitfaden sein und ist nicht als wissenschaftlich komplette Abhandlung über das Thema Requirements Engineering in der agilen Softwareentwicklung gedacht. Es kann sequenziell oder auch auszugsweise gelesen werden. Als Autor ist es mir natürlich am liebsten, wenn Sie dieses Buch ständig an Ihrem Arbeitsplatz griffbereit haben und bei Fragen oder Unklarheiten zu Requirements-Engineering-Techniken oder -Vorgehensweisen darin einen schnellen Rat und brauchbare Infos finden.

In der Praxis hat man oft nicht die Zeit, beim Auftauchen einer Frage ein ganzes Buch z. B. zu Use Cases, User Stories oder Behaviour Driven Development zu lesen. Ich habe daher auf eine möglichst große Unabhängigkeit der einzelnen Themen und Kapitel geachtet, sodass das Buch auch als Nachschlagewerk in der täglichen Arbeit verwendet werden kann. Mit entsprechendem Vorwissen in agilen Methoden und Requirements Engineering kann es auch auszugsweise gelesen und verstanden werden. Für Praktiker müssen das Wichtigste und die Zusammenhänge in Kürze ersichtlich sein. Daher gibt es in vielen Kapiteln eine tabellarische Übersicht über wesentliche Punkte und verschiedene Überblicksgrafiken, die die Zusammenhänge auf einen Blick darstellen.

Auch wenn in vielen Kapiteln Begriffe aus Scrum verwendet oder zitiert werden, so wurde darauf geachtet, die einzelnen Themenbereiche möglichst allgemeingültig zu halten. Das Buch adressiert nicht »Requirements Engineering in Scrum«, sondern behandelt generell das Thema Requirements Engineering im agilen Umfeld. Die beschriebenen Techniken und Methoden können auch in anderen agilen Vorgehensweisen angewendet werden.

#### **Beispiele werden mit einem grauen Hintergrund versehen:**

In diesem Buch werden verschiedene Beispiele und Formulierungen aus der Praxis und für die Praxis aufgeführt. Die meisten dieser Beispiele sind abgeleitet aus einem Projekt zum Thema »Zeiterfassung«. Es geht in diesem Beispielszenario darum, dass die Tagesarbeitszeiten inkl. der Pausen und Abwesenheitszeiten von Mitarbeitern eines Unternehmens erfasst, ausgewertet und verwaltet werden können und dass auch die Erfassung, Auswertung und Verwaltung von Projektarbeitszeiten durchgeführt werden können – also von einzelnen Zeitblöcken der Tagesarbeitszeit, die bestimmten Projekten zugeordnet werden.

### **1.1.1 Zielgruppen**

Das Buch richtet sich an Product Owner, Produktmanager, Projektmanager, Softwareauftraggeber, Scrum Master, Entwicklungsleiter, Entwickler, Tester und alle anderen Personen, die sich mit nachhaltigem und systematischem Requirements Engineering und Requirements Management in der agilen Softwareentwicklung beschäftigen oder davon betroffen sind.

Folgende Zielgruppen werden primär durch den [IREB CPRE FL 2017] adressiert:

- Requirements Engineers, die sich mit agiler Entwicklung befassen und ihre Techniken in dieser Umgebung erfolgreich anwenden möchten.
- Requirements Engineers, die etablierte Konzepte und Techniken agiler Ansätze anwenden und ihre Requirements-Engineering-Prozesse verbessern möchten.
- Fachkräfte für agile Entwicklungsprozesse, die die Werte und Vorteile des Requirements Engineering in agilen Projekten verstehen möchten.
- Fachkräfte für agile Entwicklungsprozesse, die die agile Entwicklung durch bewährte Requirements-Engineering-Techniken und -Methoden verbessern möchten.
- Personen aus verwandten Disziplinen – IT-Manager, Tester, Entwickler, Architekten und andere Vertreter im Bereich der Entwicklung (überwiegend, aber nicht ausschließlich Softwareentwicklung) –, die verstehen möchten, wie sie Requirements-Engineering- und agile Ansätze in Entwicklungsprozessen erfolgreich kombinieren können.

### 1.1.2 Abbildung des Lehrplans IREB RE@Agile

[IREB RE@Agile FL 2017] LE 1.3

Seit 2017 gibt es einen neuen Lehrplan des International Requirements Engineering Board (IREB®), der die Lücke zwischen Requirements-Engineering-Methoden und agilen Methoden schließt.

Einerseits wird die Sicht des IREB-Standards auf agile Werte abgebildet und andererseits wird eine agile Sicht auf die Werte des Requirements Engineering dargestellt. Zum Inhalt des Lehrplans IREB RE@Agile gehören Klassifizierung und Beurteilung von Requirements-Engineering-Artefakten und -Techniken im Zusammenhang mit Agilität, agilen Artefakten und Techniken, Requirements Engineering und wesentlichen Prozesselementen in der agilen Produktentwicklung. RE@Agile zeigt die Motivation für die Verwendung agiler Methoden in Entwicklungsprozessen auf und betont die Synergie zwischen Requirements Engineering und Agilität.

Da dieses Buch viele ergänzende und vertiefende Inhalte umfasst, die nicht im Lehrplan IREB RE@Agile enthalten sind, ist in einigen Bereichen eine andere Strukturierung gegeben. Nachfolgend ist ein erster Überblick und einige Hinweise angeführt, wo in diesem Buch die wichtigsten Inhalte des Lehrplans zu finden sind:

Lehrplan RE@Agile	in diesem Buch
<b>LE 1 Motivation und Mindset</b>	<ul style="list-style-type: none"> <li>■ Abschnitt 1.2 »Verbindung zwischen Requirements Engineering und agilem Vorgehen« und Abschnitt 2.1 »Methodenüberblick«</li> </ul>
<b>LE 2 Grundlagen von RE@Agile</b>	<ul style="list-style-type: none"> <li>■ Kapitel 2 »Grundlagen«</li> </ul>
<b>LE 3 Artefakte und Techniken in RE@Agile</b>	<ul style="list-style-type: none"> <li>■ Kapitel 3 »Requirements-Ermittlung und -Dokumentation« (enthält das Kapitel Artefakte LE 3.1 sowie die Techniken LE 3.2.1 und 3.2.2)</li> <li>■ Kapitel 4 »Requirements-Validierung und -Abstimmung« (LE 3.2.3)</li> <li>■ Kapitel 6 »Requirements Management« (LE 3.2.4)</li> <li>■ Einige Aspekte aus LE3 sind auch in Kapitel 8 »Requirements-Engineering-Rollen« enthalten.</li> </ul>
<b>LE 4 Organisatorische Aspekte von RE@Agile</b>	<ul style="list-style-type: none"> <li>■ Kapitel 7 »Organisatorische Aspekte«</li> </ul>
–	<ul style="list-style-type: none"> <li>■ Die Kapitel 8 »Requirements-Engineering-Rollen« und 9 »Rechtliche Themen« sind Ergänzungen in diesem Buch, die detailliertere Aspekte beschreiben bzw. über den Lehrplan hinaus gehen.</li> </ul>

Die für den Lehrplan [IREB RE@Agile FL 2017] relevanten Kapitel und Stellen werden mit »[IREB RE@Agile FL 2017]« sowie ggf. mit der Ergänzung »LE xxx« gekennzeichnet und sind prüfungsrelevant. LE xxx referenziert auf die jeweilige Lerneinheit (LE bzw. EU = »Educational Unit«) des Lehrplans. Alle anderen Kapitel sind ergänzende Informationen und Hinweise oder Anregungen für die Praxis.

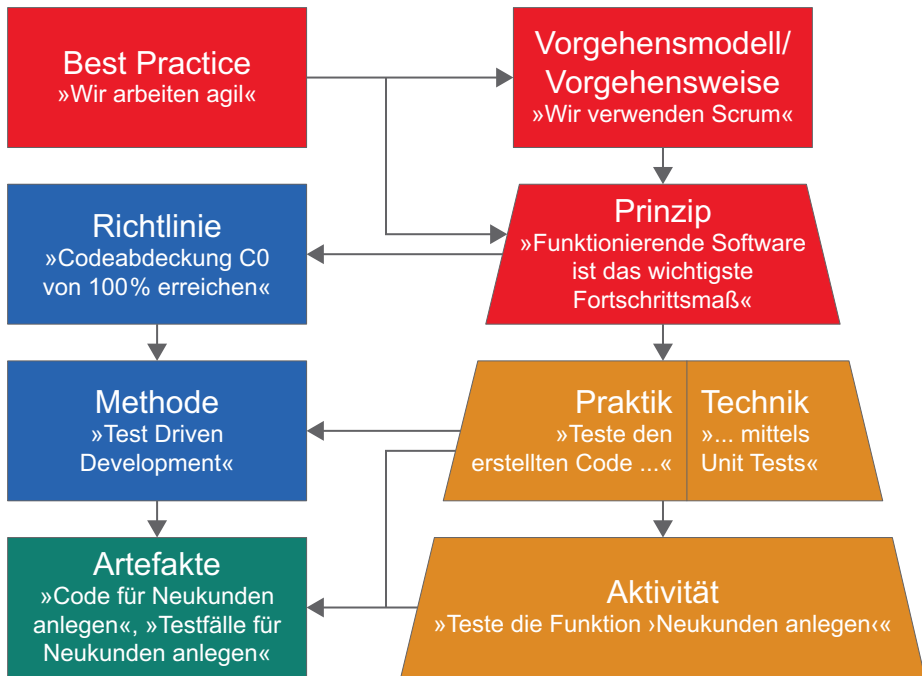
Wenn die Bezeichnung »RE@Agile« als Kurzbegriff verwendet wird, so bezieht sich dies immer auf den Lehrplan [IREB RE@Agile FL 2017].

Weiterführende Literatur zum Thema Requirements und agile Vorgehensweisen ist im Literaturverzeichnis im Anhang des Buches zu finden.

### 1.1.3 Allgemeine Begriffseinordnung

[IREB RE@Agile FL 2017]

Das umfangreiche Wissen in der Softwarebranche wird in unterschiedlichen Abstraktionsebenen beschrieben: [Meyer 2014] unterscheidet zwischen Prinzipien, Praktiken und Techniken. Diese Ebenen werden durch die Begriffe »präskriptiv« (vorschreibend), »abstrakt« und »falsifizierbar« (widerlegbar) unterschieden (siehe auch das Glossar im Anhang des Buches).



**Abb. 1-1** Abstraktionsebenen beim Wissen über die Softwareentwicklung

- Ein **Prinzip** ist z. B. »Prüfe die Requirements vor der Implementierung auf ausreichende Qualität«. Dies ist präskriptiv, da eine Aktion vorgegeben wird. Und es ist abstrakt, da keine konkreten Praktiken oder Techniken, die verwendet werden sollen, vorgegeben werden. Diesem Prinzip kann in bestimmten Situationen auch widersprochen werden. Zum Beispiel könnte es in einer Forschungssituation sinnvoll sein, ohne vorherige Requirements-Spezifikation mehrere Prototypen zu erstellen und diese gegeneinander zu validieren. Dieses Prinzip ist somit auch falsifizierbar.

Die Kenntnis der Prinzipien ist wichtig, um bewusste Entscheidungen treffen zu können. Die Falsifizierbarkeit ist wichtig, um Diskussionen und Nachdenken über Prinzipien und auch Praktiken in Gang zu bringen und zu entscheiden, ob diese in einem bestimmten Kontext angewendet werden sollen.

- Eine **Praktik** (in [IREB RE@Agile FL 2017] auch als »Verfahren« bezeichnet) ist ebenfalls präskriptiv, jedoch nicht so abstrakt, sondern eine konkretere »Instanz« des Prinzips in einem bestimmten Kontext, die Hinweise auf die Umsetzung des Prinzips gibt, ohne jedoch die tatsächliche Durchführung einer Aktivität zu beschreiben. Zum Beispiel »Prüfe jedes Backlog Item gegen die DoR (Definition of Ready), bevor es in den Sprint Backlog eingefügt wird« ist eine allgemeine Praktik, die schon konkret bestimmte Artefakte (Backlog Items, DoR und Sprint Backlog) zur Verwendung anspricht (die

Begriffe werden in den Kapiteln dieses Buches erläutert – siehe auch den Index im Anhang des Buches).

Abhängig von der jeweiligen Situation und Kontext können auch jeweils unterschiedliche Praktiken zur Erfüllung eines Prinzips angewendet werden.

- Eine **Aktivität** ist die konkrete Umsetzung einer Praktik bzw. Technik wie z. B. »Teste die Funktion ›Neukunden anlegen«.

Sowohl der Lehrplan [IREB RE@Agile FL 2017] als auch dieses Buch mit seinen erweiterten Ausführungen und Inhalten vermitteln Wissen von Requirements Engineering im Kontext der agilen Vorgehensweisen und stellen diese Abstraktionselemente in sachlicher Form vor bzw. regen zum Nachdenken und zur Diskussion über deren Anwendbarkeit an.

## 1.2 Verbindung zwischen Requirements Engineering und agilem Vorgehen

[IREB RE@Agile FL 2017] LE 1.1

Heute ist die Software und IT ein »Enabler« und Treiber in vielen Bereichen. Viele über lange Jahre etablierte plangetriebene Entwicklungsmethoden wurden nicht für ein sich schnell änderndes Umfeld ausgelegt. Agile Methoden (siehe auch Abschnitt 2.1) haben sich hier mittlerweile etabliert und schließen diese Lücke.

»Agile« oder »Agilität« wird in [Sheppard & Young 2006] wie folgt definiert: »A rapid whole-body movement with change of velocity or direction in response to a stimulus.«

Die Motivation für die Verwendung agiler Methoden ist hier definiert als Reaktion (z. B. eine schnelle Veränderung der Ausrichtung oder der Umsetzungsgeschwindigkeit) auf eine Stimulation (z. B. aus dem Projektumfeld).

Wenn man sich das Agile Manifest [Agile Manifesto] und die agilen Methodenbeschreibungen durchliest, dann geht es darin um mehr als nur Schnelligkeit. Im Grunde fokussieren alle agilen Prinzipien und Vorgehensweisen auf eine Lieferung von für den Kunden nützlichen Ergebnissen in kurzen Intervallen.

Agile Methoden oder Agilität sind jedoch kein Selbstzweck. Nicht in allen Situationen passt dieses Vorgehen gleich gut. Wenn z. B. längerfristige Planung, Vorhersagbarkeit oder Nachvollziehbarkeit erforderlich ist, müssen ggf. andere Vorgehensweisen oder Anpassungen der agilen Methoden angewendet werden.

Es ist daher wichtig, im Vorfeld darüber nachzudenken und ein passendes bzw. angepasstes Entwicklungsvorgehen zu wählen. Dies kann grundsätzlich auch dazu führen, dass in unterschiedlichen Teams entsprechend dem jeweiligen Bedarf unterschiedliche Vorgehensweisen oder auch Kombinationen aus agilen und plangetriebenen Vorgehensweisen angewendet werden (z. B. wenn im Maschinenbau

in der Hardwareentwicklung plangetriebenes Vorgehen und in der dazugehörigen Softwareentwicklung agiles Vorgehen in Kombination angewendet werden). Gartner hat dafür den Begriff »bimodale IT« geprägt. Dieses angepasste pragmatische Vorgehen ist auch einer der wesentlichen Erfolgsfaktoren der Softwareentwicklung und IT in dem heute sehr dynamischen Umfeld.

### 1.2.1 Mindsets und Werte im Requirements Engineering und agilem Vorgehen

[IREB RE@Agile FL 2017] LE 1.2

#### Requirements Engineering (RE)

In der IREB-Definition von Requirements Engineering im Lehrplan von 2017 sind Mindset und Werte von Requirements Engineering angegeben:

Requirements Engineering ist eine systematische und disziplinierte Herangehensweise zur Spezifikation und Management von Anforderungen mit den folgenden Zielen:

1. Die relevanten Anforderungen kennen, einen Konsens zwischen den Stakeholdern dieser Anforderungen erreichen, konform zu vorgegebenen Standards dokumentieren und die Anforderungen systematisch managen.
2. Die Wünsche und Bedürfnisse der Stakeholder verstehen und dokumentieren.
3. Die Anforderungen spezifizieren und managen, um ein System auszuliefern, das möglichst weitgehend den Wünschen und Bedürfnissen der Stakeholder entspricht.

Im IREB CPRE FL sind die vier Hauptaktivitäten des Requirements Engineering definiert:

- Ermittlung,
- Dokumentation,
- Prüfen und Abstimmen und
- Verwalten von Anforderungen.

Diese bilden keinen Prozess oder eine vorgegebene Folge von Aktivitäten, sondern stellen einen prozessunabhängigen Ansatz dar. Es sind Aktivitäten, die unabhängig von der gewählten Vorgehensweise durchgeführt werden sollen. In jedem dieser Aktivitäten des Requirements Engineering gibt es verschiedene Methoden und Techniken, die in allen Vorgehensweisen – sowohl in nicht agilen als auch in agilen – angewendet werden können. Dabei gibt es natürlich Unterschiede in den Detailausprägungen der angewandten Methoden und Techniken.

Zu beachten ist auch, dass im Requirements Engineering grundsätzlich von »System« gesprochen wird und nicht von Software, Produkt, Hardware, Prozess