

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

Anhang

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

1	Einleitung und Motivation	1
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
2	Grundlagen	25
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

3	Requirements-Ermittlung und -Dokumentation	53
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
4	Requirements-Validierung und -Abstimmung	177
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	Schätzmethoden	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
5	Qualität von Requirements	209
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
6	Requirements Management	227
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
7	Organisatorische Aspekte	259
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
8	Requirements-Engineering-Rollen	277
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
9	Rechtliche Themen	291
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

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