

7. Auflage

chris RUPP & die SOPHISTen

REQUIREMENTS- ENGINEERING und -MANAGEMENT

Das Handbuch für Anforderungen
in jeder Situation



INKLUSIVE: Praxistipps für Agilität, Systems-
Engineering, Smart Ecosystems/Digitalisierung



NEU: Erklärvideos und
animierte Grafiken

HANSER

SOPHIST 

Chris Rupp & die SOPHISTen

Requirements-Engineering und -Management



Bleiben Sie auf dem Laufenden!

Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter:

www.hanser-fachbuch.de/newsletter



Chris Rupp & die SOPHISTen

Requirements-Engineering und -Management

**Das Handbuch für Anforderungen
in jeder Situation**

7., aktualisierte und erweiterte Auflage

HANSER

Chris Rupp, SOPHIST GmbH, Nürnberg
www.sophist.de

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autoren und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autoren und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

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.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2021 Carl Hanser Verlag München, www.hanser-fachbuch.de
Lektorat: Brigitte Bauer-Schiewek
Copy editing: Christian Schneider, Traunstein

Layout: SOPHIST GmbH, Nürnberg
Umschlagkonzept: Marc Müller-Bremer, www.rebranding.de, München
Umschlagrealisation: Max Kostopoulos
Titelmotiv und Illustrationen: © Assad Bina Khahi
Datenbelichtung, Druck und Bindung: Firmengruppe APPL, aprinta druck, Wemding
Printed in Germany

Print-ISBN 978-3-446-45587-0

E-Book-ISBN 978-3-446-46430-8

Inhaltsverzeichnis

Einleitung	1
Die SOPHISTen: Alt und Neu	2
Das Team	3
Was gibt es Neues?	6
Teil I – Einführung	7
1 In medias RE – Grundlegendes zum Requirements-Engineering	9
1.1 Motivation für ein erfolgreiches Requirements-Engineering	10
1.1.1 Anforderungen an einen Requirements-Engineer	13
1.1.2 Der Wachstumsprozess eines Requirements-Engineers	14
1.2 Das Requirements-Gehirn – die Anforderungssammlung	16
1.3 Die Disziplin Requirements-Engineering	17
1.4 RE kompensiert die Beschränkung des menschlichen Gehirns	20
1.4.1 Wissen verfällt bzw. diffundiert	21
1.4.2 Detailtiefe und Verständnis fehlt	22
1.4.3 Verlust des Gesamtüberblicks	23
1.4.4 Missverständnisse entstehen und bleiben	23
1.4.5 Abweichende Informationen verteilen sich	24
1.5 Typische Probleme im Requirements-Engineering	24
2 Die Meyers und ihr Traum vom Smart Home	27
3 Requirements-Engineering im Überblick – von der Idee zur Anforderung	29
3.1 Anforderungen ins Gesicht geschaut	30
3.1.1 Typen von Anforderungen	30
3.1.2 Zusammenhänge zwischen Anforderungen	35
3.1.3 Gute und perfekte klassische Anforderungen	38
3.1.4 Qualität von agilen Anforderungen	41
3.2 Requirements-Engineering aus der Vogelperspektive	43
3.2.1 Ursachen und Quellen von Anforderungen	43
3.2.2 Vom Wo und Wann des Requirements-Engineerings	47
3.2.3 Requirements-Engineering im Überblick	51
4 RE ist nicht gleich RE – das richtige Maß finden	57
4.1 Requirements-Engineering in drei unterschiedlichen Szenarien	58
4.1.1 Szenario: Kundenanfrage bearbeiten	59
4.1.2 Szenario: Innovative Eigenentwicklung durchführen	60
4.1.3 Szenario: Subunternehmen beauftragen	61

4.2	So skalieren Sie RE	62
4.2.1	Einflussfaktoren	63
4.2.2	Variationspunkte im RE	68
4.3	RE in verschiedenen Vorgehensweisen	71
4.3.1	RE im Agilen	72
4.3.2	RE im klassischen Umfeld	76
Teil 2 – Wissen ermitteln		81
5	Wegweiser: Wissen ermitteln	83
5.1	Die Grundlagen für eine Planung der Ermittlung	85
5.1.1	Ermittlungsgegenstand Ziele/Produktvision	85
5.1.2	Ermittlungsgegenstand Anforderungsquellen	86
5.1.3	Ermittlungsgegenstand Systemkontext	86
5.1.4	Ermittlungsgegenstand Anforderungen	86
5.1.5	Verknüpfung Ermittlungsgegenstand – Ermittlungstechnik	86
5.2	Das Vorgehen in der Planung der Ermittlung	87
5.2.1	Living Lab für eine kooperativ getriebene Ermittlung	87
6	Ziele, Informanten und Fesseln – der erfolgreiche Start ins Requirements-Engineering	93
6.1	Ziele und Zielfindung oder Visionsbildung	94
6.1.1	Die derzeitige Realität unter die Lupe nehmen	96
6.1.2	Ziele definieren und bewerten	97
6.1.3	Arten von Zielen	97
6.1.4	Ziele beschreiben	98
6.1.5	Natürlichsprachliche Dokumentation mit Zielschablonen	99
6.1.6	Zieldokumentation als Produkt-/Projekt-Canvas	100
6.2	Anforderungsquellen – Ausgangspunkt und Mittelpunkt im RE-Universum	106
6.2.1	Der Stakeholder – das unbekannte Wesen	107
6.2.2	Das Persona-Konzept	111
6.3	Systemumfang und -kontext	113
6.3.1	Die Kontextabgrenzung	113
6.3.2	System- und Kontextgrenzen bestimmen	114
6.3.3	Dokumentation/Visualisierung des Systemumfangs und -kontext	116
7	Geschäftsprozesse ermitteln und verfeinern – Einbettung in die Realität	119
7.1	Geschäftsprozessmanagement vs. Geschäftsprozessanalyse	120
7.2	Business-Use-Cases	121
7.2.1	Business-Use-Case-Diagramm	121
7.2.2	Business-Use-Case-Beschreibung	123

7.3 Business Process Model and Notation 124

7.4 Geschäftsregeln 126

 7.4.1 Definition und Einsatzgebiete 126

 7.4.2 Decision Model and Notation (DMN) 126

8 Anforderungsermittlung – Hellsehen für Fortgeschrittene 129

8.1 Ermittlung in der normalen und der smarten Welt 130

 8.1.1 Vorbedingungen für eine gute Ermittlung 131

 8.1.2 Kano-Modell 132

8.2 Kriterien für die Auswahl von Ermittlungstechniken 134

8.3 Ermittlungstechniken 140

 8.3.1 Befragungstechniken 141

 8.3.2 Beobachtungstechniken 147

 8.3.3 Artefaktbasierte Techniken 153

 8.3.4 Kreativitätstechniken 156

 8.3.5 Co-Creation-Modelle, CrowdRE und Living Labs – neue Ansätze
 und Frameworks 158

 8.3.6 Unterstützende Techniken 159

8.4 SOPHIST-Ermittlungstechnikenauswahlmatrix 163

9 Das SOPHIST-REgelwerk – Psychotherapie für Anforderungen 165

9.1 Vom Phänomen der Transformation sprachliche Effekte 166

9.2 Die Wurzeln – das Neurolinguistische Programmieren 166

 9.2.1 Transformationsprozesse 167

 9.2.2 Kategorien der Darstellungstransformation 170

9.3 Der Umgang mit sprachlichen Effekten mit dem SOPHIST-REgelwerk 171

9.4 Die 17 Regeln des SOPHIST-REgelwerks 174

9.5 Anwendung des SOPHIST-REgelwerks 192

 9.5.1 Anwendungsbeispiele 192

 9.5.2 Sichten des REgelwerks 194

9.6 Wie erlerne ich das REgelwerk? 195

10 CrowdRE – wenn die Masse Klasse bringt 197

10.1 Crowdsourcing 200

 10.1.1 Der Crowdsourcing-Prozess 201

 10.1.2 Crowdsourcing planen 201

 10.1.3 Crowdsourcing durchführen 204

 10.1.4 Crowdsourcing abschließen 205

10.2 Crowdsourcing leichtgemacht 206

Teil III – Gute Anforderungen herleiten 209

11 Wegweiser: Gute Anforderungen herleiten 211

11.1 Was sind gute Anforderungen? 212

11.2 Der Prozess zur Herleitung guter Anforderungen 212

 11.2.1 Die Vorbereitung – realistische Ziele setzen 214

 11.2.2 Durchführung – ran an die Arbeit 217

 11.2.3 Evaluierung. 218

11.3 SHS-Szenarien 220

 11.3.1 Szenario 1: Kundenanfrage bearbeiten 220

 11.3.2 Szenario 2: Innovative Eigenentwicklung durchführen 221

 11.3.3 Szenario 3: Subunternehmen beauftragen 222

12 Anforderungen analysieren – vom Wunsch zur Absicht. 223

12.1 Überblick über die Analyse von Anforderungen 224

 12.1.1 Den Wald trotz vieler Bäume sehen. 225

 12.1.2 Der Ablauf bei der Anforderungsanalyse 226

12.2 Die Aufgaben im Detail 228

 12.2.1 Anforderungen separieren 228

 12.2.2 Notwendige Anforderungen extrahieren 230

 12.2.3 Anforderungen abstrahieren 232

 12.2.4 Fehlende Anforderungen ergänzen 233

 12.2.5 Anforderungen verfeinern 235

 12.2.6 Anforderungen verbessern. 237

12.3 Angemessener Einsatz der Tätigkeiten. 238

 12.3.1 Die richtige Qualität erzeugen 239

 12.3.2 Was wirklich benötigt wird. 240

13 Nicht-funktionale Anforderungen – die heimlichen Stars 243

13.1 Definition, Bedeutung und Chancen 244

13.2 Erhebungsprozess für NFAs 245

 13.2.1 Vorbereitung 245

 13.2.2 Ermitteln 246

 13.2.3 Dokumentieren 248

 13.2.4 Evaluierung. 249

 13.2.5 Best Practices. 249

13.3 Steckbrief „Anforderungen an die Technologie“ 250

13.4 Steckbrief „Qualitätsanforderungen“ 252

13.5 Steckbrief „Anforderungen an die Benutzungsoberfläche“ 256

13.6 Steckbrief „Anforderungen an sonstige Lieferbestandteile“ 258

13.7	Steckbrief „Anforderungen an durchzuführende Tätigkeiten“	259
13.8	Steckbrief „Rechtlich-vertragliche Anforderungen“	260
13.9	Fazit	262
14	Prüftechniken für Anforderungen – ungeahntes Verbesserungspotenzial	263
14.1	Reviews	264
14.1.1	Stellungnahme	264
14.1.2	Walkthrough	265
14.1.3	Inspektion	267
14.2	Prototyp	268
14.3	Reverse Presentation	268
14.4	Metriken	269
14.5	Testfälle	270
14.6	Analysemodell	272
14.7	Hilfsmittel bei der Prüfung	274
14.7.1	Lesetechniken	274
14.7.2	Checklisten	274
14.7.3	SOPHIST-REgelwerk	275
14.7.4	Anforderungsschablone	275
14.8	Vom Durchblick im Dschungel der Prüftechniken: Die Auswahl geeigneter Prüftechniken	275
15	Anforderungskonflikte – Gehasst? Geliebt? Gelöst!	277
15.1	Was ist ein Konflikt?	278
15.2	Konfliktidentifikation	279
15.2.1	Konfliktindikatoren in der Kommunikation	279
15.2.2	Konfliktindikatoren in der Dokumentation	280
15.3	Konfliktanalyse	280
15.3.1	Konfliktursachen	281
15.3.2	Konfliktentwicklung	282
15.3.3	Konfliktgegenstand/betroffene Anforderungen	282
15.3.4	Beteiligte Stakeholder	282
15.3.5	Konfliktpositionen	282
15.3.6	Konfliktarten	283
15.3.7	Konfliktfolgen	285
15.3.8	Konfliktrisiken	285
15.4	Konfliktauflösung	286
15.4.1	Konsolidierungstechniken	287
15.4.2	Auswahl der Konsolidierungstechniken	289
15.5	Dokumentation der Anforderungskonsolidierung	291

Teil IV – Anforderungen dokumentieren und vermitteln 293

16	Wegweiser: Anforderungen dokumentieren und vermitteln	295
16.1	Anforderungen vermitteln	296
16.2	Wie plane ich die Vermittlung?	297
16.2.1	Vorbereitung	298
16.2.2	Durchführung	300
16.2.3	Evaluierung	300
16.3	Einflussfaktoren für die Vermittlung	301
16.3.1	Einflussfaktor: Ziel der Vermittlung/Inhalt	301
16.3.2	Einflussfaktor: Umfang des zu vermittelnden Betrachtungsgegenstandes	302
16.3.3	Einflussfaktor: Komplexität des Betrachtungsgegenstands	302
16.3.4	Einflussfaktor: Qualitätskriterien	303
16.3.5	Einflussfaktor: Vergessen und Wissensstabilität	303
16.3.6	Einflussfaktor: Wiederverwendung von Anforderungen	304
16.3.7	Einflussfaktor: Verfügbarkeit	305
16.3.8	Einflussfaktor: Normen, Standards und Vorgaben	305
16.3.9	Einflussfaktor: Sprache	305
16.3.10	Fazit	306
16.4	Anforderungen dokumentieren	306
16.4.1	Der Klassiker – die Anforderungsspezifikation	306
16.4.2	Die agile Welt – das Product-Backlog	307
17	Storytelling, User-Storys und Co. – verschiedene Arten, Anforderungen zu vermitteln	309
17.1	Storytelling – Grimms Märchen der Anforderungsvermittlung	310
17.1.1	Arten des Storytellings	311
17.1.2	Was macht gutes Storytelling aus?	312
17.1.3	Die irrelevanten Teile einer Story	313
17.1.4	Gute Geschichten für eine gute Vermittlung	314
17.2	User-Storys und Story Mapping	315
17.2.1	Verschiedene Detaillierungsebenen von User-Story – von Epics bis zu detaillierten User-Storys	316
17.2.2	Vermitteln mit User-Storys	316
17.2.3	Formulieren einer User-Story	317
17.2.4	Das Gespräch zu einer User-Story	318
17.2.5	Story Mapping – das Gesamtbild betrachten	319
17.2.6	Gute User-Storys für eine gute Vermittlung	320
17.3	Prototypen – everybodys darling	320
17.3.1	Wireframe – das Drahtmodell für den Bildschirm	320
17.3.2	Funktionaler Prototyp – erlebte Funktion	322
17.3.3	Mock-up der Oberfläche – das Designmodell	322
17.3.4	Gute Prototypen für eine gute Vermittlung	323

17.4	Bilder zur Vermittlung von Wissen	323
17.4.1	Definition der eigenen Bildsprache	325
17.4.2	Verbindliches von nicht Verbindlichem trennen	325
17.4.3	Kombination Bild mit anderen Techniken der Wissensvermittlung	326
17.4.4	Bilder für eine gute Vermittlung	327
17.5	Gemeinsam Artefakte erstellen	327
17.5.1	Vorbereitung	328
17.5.2	Überblick geben	328
17.5.3	Erstellen der Testfälle	329
17.5.4	Gemeinsam erstellte Artefakte für eine gute Vermittlung	330
18	Anforderungen modellieren – malen statt schreiben	331
18.1	Modelle geben Struktur	332
18.2	Use-Case-basierte vs. zustandsbasierte Analyse	333
18.2.1	Use-Case-basierte Analyse	335
18.2.2	Zustandsbasierte Analyse	336
18.3	Use-Cases des Systems beschreiben	338
18.3.1	Das Use-Case-Diagramm	339
18.3.2	Die Use-Case-Beschreibung	341
18.4	Systemabläufe beschreiben	343
18.4.1	Systemabläufe in Aktivitäten beschreiben	344
18.4.2	Systemabläufe in Sequenzen beschreiben	346
18.5	System- und Objektzustände beschreiben	349
18.6	Begriffe und Informationsstrukturen beschreiben	351
18.6.1	Das Glossar	352
18.6.2	Das Informationsmodell – Zusammenhänge von Fachbegriffen	353
19	Schablonen für Anforderungen und User-Storys – MASTER und andere Templates	357
19.1	Linguistische und philosophische Grundlagen	358
19.2	Der schablonenbasierte Ansatz	359
19.2.1	Der Bauplan einer Anforderung	359
19.2.2	Anwendungsgebiete	360
19.3	Schritt für Schritt zur Anforderung	361
19.3.1	Schritt 1: Betrachtungsgegenstand identifizieren	362
19.3.2	Schritt 2: Wichtigkeit festlegen	362
19.3.3	Schritt 3: Funktionalität identifizieren	362
19.3.4	Schritt 4: Art der Funktionalität festlegen	363
19.3.5	Schritt 5: Objekt identifizieren	365
19.3.6	Schritt 6: Bedingungen formulieren	365
19.3.7	Schritt 7: SOPHIST-REgelwerk anwenden	367

19.4	Semantische Präzisierung	367
19.4.1	Wichtigkeit definieren.	368
19.4.2	Verben definieren	369
19.4.3	Substantive definieren	371
19.5	Details für die Konstruktion.	373
19.5.1	Präzisierung des Objekts.	374
19.5.2	Präzisierung des Verbs	374
19.6	Schnell und einfach zur User-Story	375
19.6.1	Aufbau und Inhalt einer User-Story	375
19.6.2	Aufbau und Inhalt von Akzeptanzkriterien für User-Stories.	376
19.7	Nicht-funktionale Aspekte	377
19.7.1	Eigenschaften	378
19.7.2	Umgebungen und Kontext	379
19.7.3	Prozesse	382
19.8	Bedingungen	383
19.9	Schablonen innerhalb der Szenarien	386
19.9.1	Szenario 1 „Kundenanfrage bearbeiten“	386
19.9.2	Szenario 2 „Innovative Eigenentwicklung durchführen“	387
19.9.3	Szenario 3 „Subunternehmen beauftragen“	387
19.10	Auf die Sätze, fertig, los!	387

Teil V – Anforderungen verwalten 389

20	Wegweiser: Anforderungen verwalten	391
20.1	Was ist Requirements-Management?.	392
20.2	Grundannahmen für professionelles Requirements-Management – die drei Gebote	393
20.2.1	1. Grundannahme: Anforderungen ändern sich	394
20.2.2	2. Grundannahme: Anforderungen werden weiterverwendet	394
20.2.3	3. Grundannahme: Anforderungen sind nicht die einzige relevante Informationsart für erfolgreiches Requirements-Engineering	394
20.3	Die Aufgaben professionellen Requirements-Managements.	395
20.3.1	Informationsaustausch – wer gibt wann wem was?	396
20.3.2	Ablaufsteuerung – wer darf wann was?	396
20.3.3	Verwaltung von Abhängigkeiten und Nachvollziehbarkeit – was hängt wie womit zusammen?	397
20.3.4	Auswertung und Projektsteuerung – wie läuft’s?	397
20.4	Wie gestalte ich mein Requirements-Management? – Rahmenbedingungen, Einschränkungen und Einflussfaktoren.	398
20.4.1	Wann ist wie viel Requirements-Management sinnvoll? – Rahmenbedingungen identifizieren	400
20.4.2	Das einzig Beständige ist der Wandel – Handlungsspielraum und -felder identifizieren	405

21	Strukturen und Zustände – wider die Unordnung	407
21.1	Informationsarten definieren – was genau soll verwaltet werden?	408
21.2	Dokumentenlandschaft definieren	411
21.3	Anforderungssammlung strukturieren	414
21.3.1	Gliederungsstrukturen – das Skelett des Requirements-Managements	414
21.3.2	Standardgliederungen – das Rad nicht neu erfinden	415
21.3.3	Story Mapping – ein Product-Backlog strukturieren	418
21.4	Anforderungen strukturieren	419
21.4.1	Nicht-funktionale Anforderungen strukturieren	420
21.4.2	Funktionalitäten strukturieren	421
21.5	Zustände, Rechte und Rollen	423
21.5.1	Zustände einer Anforderung	423
21.5.2	Der Zustandsautomat einer Anforderung	424
21.5.3	Rollen identifizieren	431
21.5.4	Rechte vergeben	432
22	Attribute, Traces, Historie – das Chaos verhindern	435
22.1	Attribuierung – Verwaltungsinformationen ergänzen	436
22.1.1	Attributtypen definieren	437
22.1.2	Attribuierungsschema definieren	441
22.1.3	Die Objekt-ID – Anforderungen eindeutig identifizieren	443
22.2	Sichten bilden	444
22.2.1	Selektive Sichten – Informationen filtern, sortieren und gruppieren	444
22.2.2	Reporting – verdichtende Sichten	445
22.3	Anforderungen historisieren und versionieren	447
22.3.1	Anforderungen historisieren	447
22.3.2	Anforderungen versionieren	448
22.3.3	Konfigurationen und Basislinien	449
22.4	Verfolgbarkeit/Traceability herstellen	450
22.4.1	Die Eltern-Kind-Verbindung – Verfeinerungs- und Ableitungshierarchien abbilden	453
22.4.2	Verbindung von Informationen in gleichem Verfeinerungsgrad	454
22.4.3	Ein Verfolgbarkeitsmodell definieren	456
22.4.4	Umsetzung der Verfolgbarkeit	458
22.5	Change-Management – Anforderungsänderungen bearbeiten	459
22.5.1	Vom Änderungswunsch zur Umsetzung	461
22.5.2	Der Change-Management-Prozess	462

Teil VI – Weitere RE-Aspekte 465

23 Systems-Engineering – Systemdenken und RE467

- 23.1 Warum ein schnelleres Pferd noch kein Einhorn ist! 468
- 23.2 Das Twin-Peaks-Modell 470
- 23.3 Architektur im Systems-Engineering 471
 - 23.3.1 Tätigkeiten in der Architektur 472
 - 23.3.2 Black-Box-Sicht – technischer Kontext 472
 - 23.3.3 White-Box-Sicht mit Blockdefinitionsdiagrammen 476
 - 23.3.4 White-Box-Sicht mit dem Internen Blockdiagramm. 476
- 23.4 Anforderung und Realisierung verbinden 478
 - 23.4.1 Allokationssicht 478
 - 23.4.2 Schnittstellen im Systems-Engineering 480
- 23.5 Mountain-View-Modell – Sichten im SE 481
 - 23.5.1 Organisations-Peak 481
 - 23.5.2 Testfall-Peak. 482
 - 23.5.3 Feature-Peak 482
 - 23.5.4 Funktionale Wirkketten und weitere Sichten 483
- 23.6 Analysen und weitere Methoden 484
 - 23.6.1 Quality Function Deployment 484
 - 23.6.2 Hazard Analysis and Risk Assessment 486
 - 23.6.3 Failure Mode and Effects Analysis 487

24 Die digitale REvolution – Anforderungen an Smart Ecosystems und Industrie 4.0 489

- 24.1 Definition und Begriffsabgrenzung – „Smart Eco... was?“ 490
 - 24.1.1 Informations-, eingebettete und mobile Systeme – die Grundsystemarten 490
 - 24.1.2 Emergente Systeme 491
 - 24.1.3 Cyber-physische Systeme 492
 - 24.1.4 Smarte Ökosysteme/Smart Ecosystems 492
- 24.2 Die digitale Transformation bzw. der digitale Wandel 494
- 24.3 Herausforderungen für die Entwicklung von Systemen innerhalb eines Smart Ecosystems. 495
 - 24.3.1 Autonomie – jeder ist sich selbst der Nächste 495
 - 24.3.2 Diversität – es lebe die Vielfalt. 495
 - 24.3.3 Komplexität – höher, schneller, weiter, größer. 496
 - 24.3.4 Selbstadaption – Maschinen als TÜV-Prüfer. 496
 - 24.3.5 Vernetzung – alles mit allem, jeder mit jedem. 497

24.4	Einfluss der digitalen Transformation und Smart Ecosystems auf das Requirements-Engineering	497
24.4.1	Auswirkungen der digitalen Transformation auf die Tätigkeiten des Requirements-Engineerings	497
24.4.2	Auswirkungen von Smart Ecosystems auf das Requirements-Engineering	499
24.5	Die Komplexität beherrschen – mögliche Lösungsansätze zur Spezifikation im Rahmen von Smart Ecosystems	502
24.5.1	Model-based Systems-Engineering	502
24.5.2	Künstliche Intelligenz	504
25	RE für Produktlinien und -familien – auf dem Weg zum individuellen Massenprodukt.	505
25.1	Von der Individualität der Masse	506
25.2	Grundlagen	506
25.3	Referenzprodukt.	508
25.4	Die Variante.	512
25.4.1	Ermittlung einer konkreten Variante	512
25.4.2	Ausleitung der Anforderungen für eine Variante	516
25.5	Erweiterungen und Änderung des Referenzprodukts.	518
25.6	Weiterbearbeitung in der Architektur	519
25.6.1	Übernahme der Features	519
25.6.2	Transformation der Features	520
25.6.3	Definition neuer Features	520
25.7	Herausforderungen in der Praxis	521
25.7.1	Definition komplizierter Abhängigkeiten	521
25.7.2	Tools	522
26	Einführungsstrategien – ein Ratgeber für die organisierte REorganisation . . .	523
26.1	Gründe für eine gute Strategie	524
26.1.1	Warum sollte ich mich ändern?	524
26.1.2	Und warum ist das nicht so einfach?	525
26.2	Eine Einführung ist ein Projekt!	526
26.3	Alle Wege führen nach	527
26.3.1	Top-down-Einführung – alles Gute kommt von oben – Beschreibung der Enterprise-Transition-Community	528
26.3.2	Middle-out – Scrum-Software-Studio als Mittler zwischen den Welten	530
26.3.3	Bottom-up – teamweise, partiell oder unter der Tarnkappe	533
26.3.4	Best in Show – agiles Change-Management	535

26.4	Arbeitspakete einer Einführung	541
26.4.1	Marketingkonzept	541
26.4.2	Konzept zur Wissensvermittlung	542
26.4.3	Pilotierungskonzept	546
26.4.4	Migrationskonzept	548
27	Videos im RE – Hollywood für Anforderungen	549
27.1	Warum Videos im RE?	550
27.2	Ein PILZ stellt sich vor	550
27.2.1	Phase	552
27.2.2	Inhalt	553
27.2.3	Lösungsbezug	553
27.2.4	Zeitbezug	554
27.2.5	PILZe sammeln in den Szenarien	554
27.2.6	Allgemeine Handlungsempfehlungen	556
27.2.7	Handlungsempfehlung Phase	556
27.2.8	Handlungsempfehlungen Inhalt	557
27.2.9	Handlungsempfehlungen Lösungsbezug	558
27.2.10	Handlungsempfehlungen Zeitbezug	559
27.3	Der Videoworkshop	560
27.4	Toll, ein Video ... und jetzt?	561
	Literaturverzeichnis	563
	Videoverzeichnis	573
	Animationsverzeichnis	575
	Index	577

Einleitung



Liebe Leserin, lieber Leser

Wir freuen uns sehr, Ihnen die neue Auflage unseres Buches Requirements-Engineering und -Management präsentieren zu können, und möchten uns bei Ihnen ganz herzlich bedanken, dass Sie dieses Buch zur Hand genommen haben.

Viel Arbeit ist in dieses Buch geflossen. Viel wurde diskutiert, verbessert, verändert, verworfen. Viele neue Ideen, neue Methoden und eine Menge Praxiserfahrung finden sich auf den kommenden Seiten. Die Tatsache, dass viele Menschen an diesem Buch mitgewirkt haben, hat zum einen den Vorteil, dass viele verschiedene Erfahrungen, Ideen und auch Sichtweisen in das Buch einfließen konnten. Zum anderen wirkt die große Beteiligung auch Probleme auf, denn die verschiedenen Meinungen müssen in eine im Buch abgedruckte münden. Das kann durchaus auch mal sehr lustig sein. Zur Veranschaulichung zeigen wir Ihnen an dieser Stelle ein paar Ideen für den Untertitel des Buches, die es leider nicht auf das Cover geschafft haben:

- Der Watzlawicksche Universalhammer für Requirements-Engineering
- Don't believe everything you hear about Smart Ecosystems, agile and Systems-Engineering
- Willst du viel, mach's klassisch oder agil
- Kann immer und überall
- Alles kann, nichts muss
- Die Wunderwaffe für agile, klassische, smarte und technische Projekte
- Und jetzt auch mit Smart Ecosystems und Systems-Engineering
- Vorsicht: Kann Spuren von ... enthalten
- Der praktische Ratgeber für die täglichen REalitäten
- Ein Buch für alle Fälle

Sie sehen schon, bei manchen Vorschlägen können wir froh sein, dass sie sich nicht durchgesetzt haben.

Die SOPHISTen: Alt und Neu

Die Sophisten, eine Gruppe von Philosophen, lebten in der Zeit um 450 vor Christus in Athen. Sie galten als die Ersten, die auf die von den Vorsokratikern propagierte Naturphilosophie eine *menschenbezogene* Antwort gaben. Protagoras (481 – 411) postulierte: „Der Mensch ist das Maß aller Dinge“. Sie gaben auch die entscheidenden Impulse für die Entwicklung vom Mythos zum Logos, das heißt zur Idee eines durch theoretische Vernunft begründeten Weltverständnisses.

Als SOPHISTen der Neuzeit bezeichnen sich die Mitarbeiter der SOPHIST GmbH, der Gesellschaft für innovatives Software-Engineering. Die Ideen und die Werte der alten

Sophisten haben wir aufgegriffen und sehen es als Teil unserer Mission, unsere Kunden dazu zu bringen, das Althergebrachte infrage zu stellen. Seit Jahren begleiten die SOPHISTen namhafte Kunden in unterschiedlichen Projekten mit Coaching, Training und Auditierung. Dadurch entstand ein umfassender Wissenspool in den Bereichen Requirements-Engineering und -Management und Architektur.

Das Team

Wieder waren sehr viele Personen an der Entstehung der neuen Auflage beteiligt. Es bedarf nicht nur der Personen, die das Buch schreiben, sondern auch Menschen, die sich um Layout, Review, Druck, Organisation und Planung kümmern. All jenen möchten wir ein großes Dankeschön sagen. Falls Sie die Autoren etwas näher kennenlernen möchten, finden Sie auf unserer Webseite unter www.sophist.de mehr Informationen.

Und da waren noch ...

Neben dem Team waren aber noch weitere Menschen an der Entstehung dieser Auflage beteiligt. Zum Beispiel all die Menschen, die uns mit Lob, Kritik, Verbesserungswünschen etc. viele Anregungen zu Inhalt und Gestaltung dieser Auflage gegeben haben. Auch Sie können gerne Ihre Meinung – egal ob gut oder schlecht, wir nehmen alle sehr ernst – zu dieser Auflage mit uns teilen. Schreiben Sie uns einfach an buch@sophist.de.

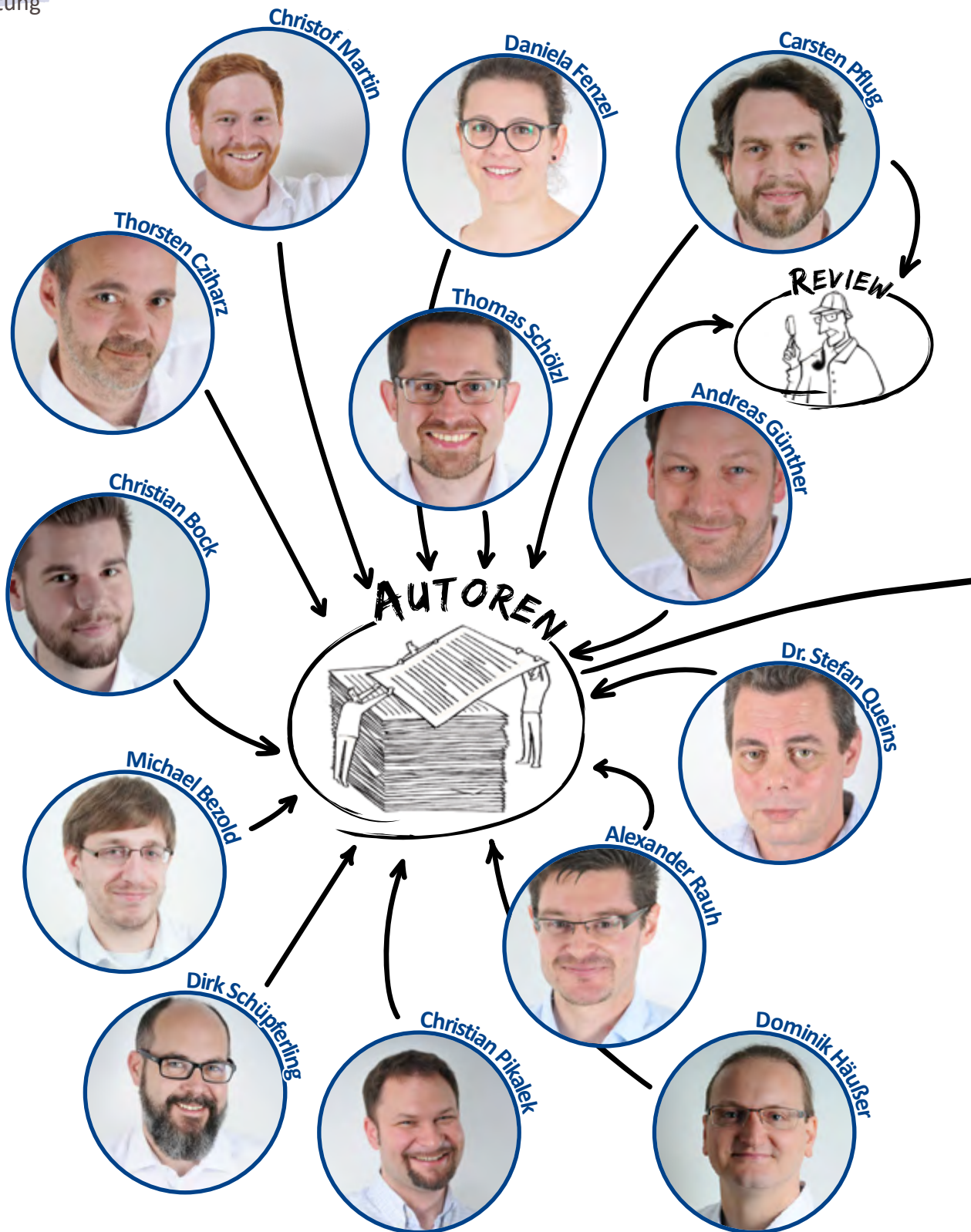
Zusätzlich haben wir für diese Auflage eine neue Technik ausprobiert, die wir in diesem Buch auch beschrieben haben: das CrowdRE. Wie diese Technik funktioniert, können Sie in [Kapitel 10 „CrowdRE“](#) nachlesen. Wir haben mit dieser Technik Wünsche und Anregungen für diese Auflage im Vorfeld eingesammelt und in dieses Buch einfließen lassen. In diesem Zuge möchten wir uns herzlich bedanken bei

Dr.-Ing. Adam Fiolka, Cecilia La Fuente, Anne Hoffmann, Prof. Dr. Isabel John, Florian Kopf, Achim Krzesinski, Christian Mies, Bonifaz Stuhr, Vincenz Vogler, Prof. Dr. Klemens Waldhör, Dr. Lennard Wasserthal, Jens Weber, Dipl.-Ing. Silvia Wirrer.

Ihre Anregungen haben zum Gelingen dieser Auflage beigetragen.

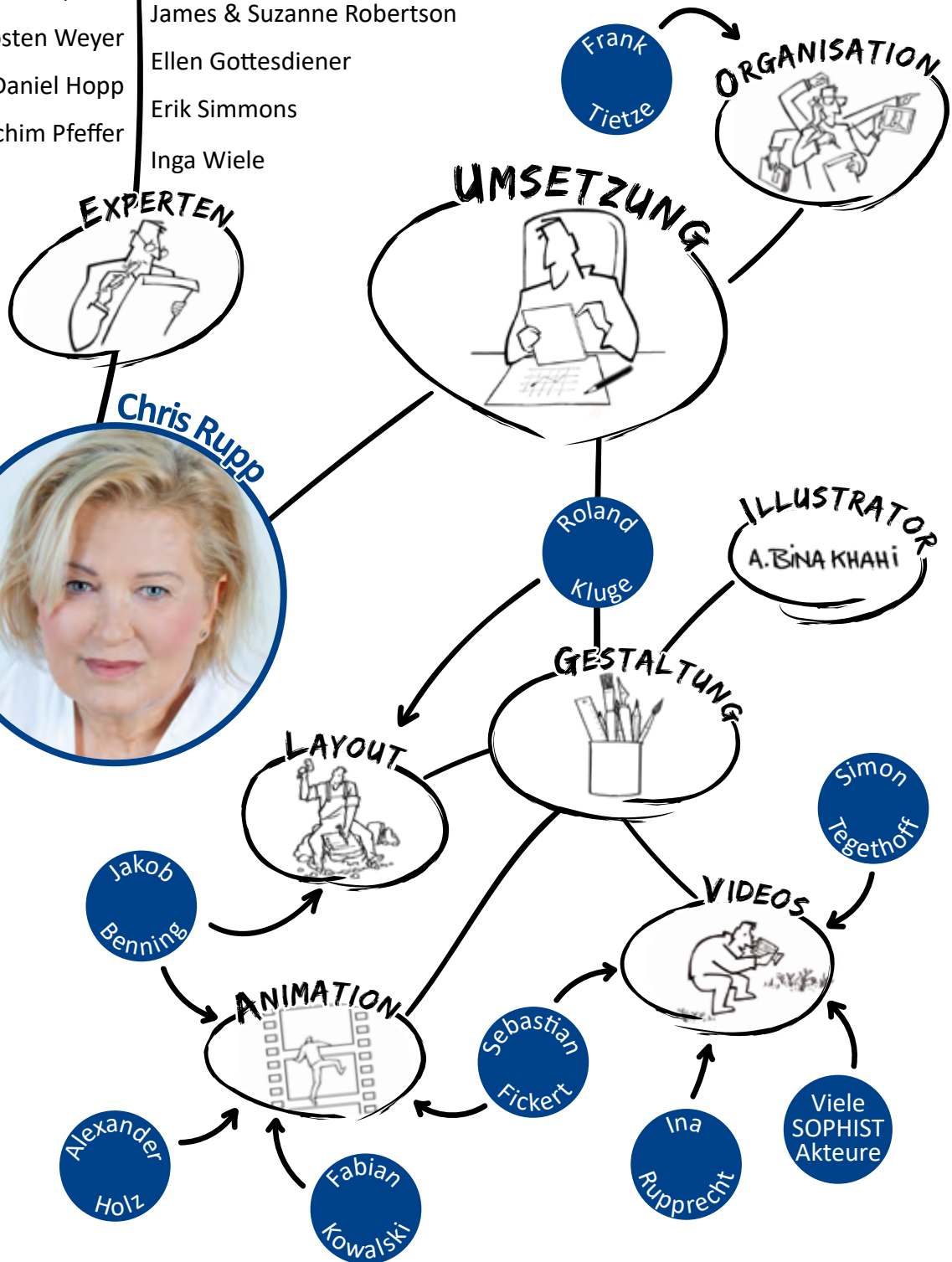
Der Zeichner

Vielen Dank auch an unseren Zeichner. Assad Bina Kahi, geboren 1960 in Masjedso-lyman/Iran, realisierte die Zeichnungen, die unser Buch zieren. Er schloss 1985 an der Filmakademie Teheran ab. Danach begann er ein Zweitstudium an der Universität Teheran mit Diplomabschluss im Fach Grafik. 1995 erlangte er den Magister im Fach Zeichentrickfilm. Anschließend arbeitete er als Dozent für Trickfilm an der Universität Teheran und nahm 1990 und 1992 am Internationalen Filmfestival in Stuttgart teil. Er war Zeichner und Cartoonist bei einer Kinderzeitung (Aftabgardan), einer Tageszeitung (Hamschahrie) und einer Cartoon-Fachzeitschrift (Keyhankarikatur) und produzierte mehrere Zeichentrickfilme für Werbeagenturen. Assad Bina Kahi lebt seit 1996 in Deutschland.



Dr. Stefan Queins
Dr. Throsten Weyer
Daniel Hopp
Joachim Pfeffer

Dr. Joerg Dörr & Eduard C. Groen
James & Suzanne Robertson
Ellen Gottesdiener
Erik Simmons
Inga Wiele



Was gibt es Neues?

Da wir auch zu dem stehen, was wir anbieten, haben wir ebenfalls Althergebrachtes hinterfragt und uns entschlossen, die in diesem Buch zu findenden Abbildungen an ausgewählten Passagen mit Videos und Animationen zu unterfüttern. Markiert sind die Passagen wie folgt:



Mittels des QR-Codes gelangen Sie direkt auf den enthaltenen Link, ohne diesen abtippen zu müssen. Ein Smartphone oder Tablet beim Lesen griffbereit zu haben, wird sich für den ein oder anderen durchaus lohnen.

Nun aber los!

Wir wollen Sie nun nicht länger auf die Folter spannen und lassen Sie in die bunte und spannende Welt des Requirements-Engineerings abtauchen. Falls Sie zu den Lesern gehören, welche zuerst die letzte Seite lesen, um sofort zu wissen, wie das Buch ausgeht, möchten wir Ihnen den kleinen Tipp geben: „Das Ende ist nicht auf der letzten Seite“.

Einführung

- **Kapitel 1:**
In medias RE – Grundlegendes zum Requirements-Engineering
- **Kapitel 2:**
Die Meyers und ihr Traum vom Smart Home.
- **Kapitel 3:**
Requirements-Engineering im Überblick – Von der Idee zur Anforderung
- **Kapitel 4:**
RE ist nicht gleich RE – das richtige Maß finden

In medias RE – Grundlegendes zum Requirements-Engineering



1.1 Motivation für ein erfolgreiches Requirements-Engineering

Unser Leben ist stark von IT-Systemen geprägt – hängt teilweise sogar von ihnen ab. Sie machen unser Dasein deutlich angenehmer, strukturieren und gestalten es mit. Sie liefern Informationen, unterstützen bei Entscheidungen oder Arbeitsvorgängen und automatisieren Vorgänge. Und sie ermöglichen uns Dinge, von denen wir vor wenigen Jahren noch nicht mal geträumt haben. Wir leben in einem Smart Home, unser Auto besteht aus intelligenten Komponenten, unsere Haushaltsgeräte kommunizieren mit uns und das Smartphone verbindet uns immer und überall mit dem Rest der Welt. In der Industrie werden Waren smart gefertigt, die Landwirtschaft, die Energieversorgung, das Gesundheitssystem ... leisten mehr, da Systeme die Wertschöpfung überwachen und optimieren.

Damit all das zu einem Traum für die Menschheit wird und nicht im Albtraum endet, ist es wichtig, dass die Systeme das tun, was sie tun sollen und was wir von ihnen wollen. Genau hier setzt Requirements-Engineering an. Diese immer komplexeren Leistungen, die Systeme übernehmen, müssen erfunden oder ermittelt, analysiert und vermittelt werden und das entstandene Wissen (die Anforderungen) muss dann oft auch dokumentiert und verwaltet werden. Vorher müssen Sie die vom System zu unterstützenden Geschäftsprozesse verstehen und skizzieren.

Die Kunst dabei ist es, auf die mannigfaltig vorliegenden Rahmenbedingungen einzugehen. Requirements-Engineering bedeutet hier die richtige Form für den richtigen Zweck zu finden. Wir haben uns für dieses Buch auf ein Beispiel – unser Smart Home – geeinigt und drei typische Szenarien beschrieben, in denen Requirements-Engineering in der Realität häufig stattfindet. Das Beispiel finden Sie in [Kapitel 2 „Die Meyers und ihr Traum vom Smart Home“](#), die Szenarien in [Kapitel 4 „RE ist nicht gleich RE“](#).

Und hier noch eine ganz persönliche Geschichte, die Sie zu unserem durchgehenden Beispiel in diesem Buch hinführt. Es handelt sich dabei um ein Beispiel für Storytelling, eine Vermittlungstechnik, die sie in [Kapitel 17 „Storytelling, User-Stories & Co.“](#) kennenlernen werden. Ganz nebenbei erläutert diese Geschichte, warum Requirements-Engineering wichtig ist.

Ich lebe auf dem Land, und zwar wirklich auf dem Land – auf einem Bauernhof in einem abgelegenen Weiler. Hier guckt nicht ständig ein Nachbar über den Zaun, ob bei mir alles o. k. ist. So kam mir der Gedanke, dass etwas mehr Sicherheit und gerne auch Komfort durch mehr Digitalisierung meines Anwesens eine gute Idee wäre. Super wäre dann, wenn die zusätzliche Intelligenz im Haus auch noch für eine ökologische Nutzung der durch die vorhandene Solar- und Fotovoltaik-Anlage erzeugten Energie sorgen könnte. Wichtig waren mir aber erst mal ein paar Kameras und eine neue Haustüre mit sinnigen Features für Sicherheit und Komfort sowie eine Alarmanlage.

Auf die Idee kam ich, da gerade jeder in meinem Bekanntenkreis sein Haus smart machte, irgendwie begeistert davon erzählte und mir auf dem Smartphone zeigte, was gerade bei ihm so im Garten abging (meist nix). Da würde bei mir daheim schon mehr abgehen, denn ich habe zwei große Hunde und einige Katzen – und da würde mich interessieren, was die tagsüber so treiben.

Natürlich hatte ich überhaupt keine große Lust aus der Smartifizierung ein großes Ding zu machen.

Meine erste Vision war ein Standardprodukt von meinem Elektriker installieren zu lassen – und initial den Fokus nur auf die Eingangstüre und die Kameras zu legen. Da ich im ersten Stock schlafe, hätte ich gerne noch mittels einer Alarmanlage dafür gesorgt, dass ich rechtzeitig geweckt werde, wenn jemand gerade einbricht.

Auf die Idee Standardprodukt kam ich, nachdem ich mit einem Kollegen gesprochen hatte, der ein IT-Freak ist. Die Komplexität und der Ideenreichtum seines selbst konfigurierten Systems haben mich derart überfordert – ich fühlte mich nach einigen Minuten schon durch die Erzählung abgeschreckt. Somit besuchte ich einen weniger freakigen Kollegen, der mit seiner Frau und den Kindern in der Gegend wohnt. Sein System war wirklich Standard – vom Hauselektroniker eingebaut. Im Großen und Ganzen bestand es aus ein paar Kameras im Außenbereich, deren Bilder er über sein Handy ansehen konnte, und aus einer Alarmanlage für das Erdgeschoss, die er scharf schaltete, wenn er das Haus verließ oder wenn er zum Schlafen mit der Familie in das obere Stockwerk ging. Bei dem Besuch gingen wir dann alle nach oben und er zeigte mir, wie einfach es war, über das Smartphone jetzt die Alarmanlage zu aktivieren. Wir checkten auch gleich mal, was man am Abend noch so über die Außenkameras sehen kann – und das war leider etwas enttäuschend, denn die Nachtsicht war damals bei der Einrichtung anscheinend keine Anforderung gewesen. Mein Kollege erzählte, dass er aber ein Bild auf das Handy gesendet bekäme, sobald die Kamera eine Bewegung detektieren sollte. Bei der Idee wurde mir angst, denn in meinem Garten springen nachts Rehe, Katzen, Füchse und sonst noch so einiges herum. Das würde sehr viele Meldungen und Bilder auf meinem Handy bedeuten. Als wir nun alle so über das Smartphone gebeugt dastanden, ging plötzlich die Alarmanlage los. Irgendwie musste das System wohl doch nicht wie gewünscht funktionieren. Sofort fiel mir auf, dass Pirata und Nemo – meine beiden Hunde – nicht mehr neben uns waren. Sie hatten anscheinend die Erzählung langweilig gefunden und die Hauserkundung im unteren Stockwerk fortgesetzt – überwacht durch die Alarmanlage, die sie sofort für Einbrecher hielt. Frust machte sich langsam bei mir breit, denn anscheinend bin ich doch kein Standardkunde – nachts schlafen meine Hunde im Gang im Erdgeschoss und die Katzen kommen und gehen, wann sie wollen. Da wir dann schon mal bei den Problemen einer derartigen Anlage angekommen waren, packte mein Kollege gleich mal aus, was so alles nicht geht in dem Standard-Ding. Das Zufügen und Löschen weiterer Personen, die mittels Key Zutritt zum Haus haben, war sperrig und man konnte lediglich Zutritt erteilen oder nicht – also nix mit Profilen, wie „der Handwerker darf nur an einem bestimmten Tag zu einer gewissen Zeit rein“. Und eine offene Frage quälte meinen Kollegen: Was passiert, wenn es mal im Haus brennt und die Kinder noch drin sind. Man konnte mit dem System nicht Fernentriegeln – sodass die Feuerwehr bei der einbruchsicheren Türe und den sehr massiven Fenstern nie ins Haus käme, um die Kinder zu retten ...



Mir wurde klar: Wenn ich mich nur zehn Minuten lang hinsetzen würde, dann würden mir sicherlich einige Anforderungen einfallen, die nicht Standard sind, bei meiner Art zu leben aber unabdingbar. Mir war aber auch klar, dass mein Wissen zum Thema nicht ausreicht, um zu überblicken, was ich noch so alles beachten und fordern sollte. Ich bin eben Spezialist bezüglich meiner Lebensumstände, aber kein Smart-Home-Spezialist. Und klar – typisch, dass ich als Requirements-Engineer erst mal x Versuche unternehme dem Problem einer sinnvollen Analyse und sauberer Anforderungen zu entgehen –, der Schuster hat ja angeblich auch die schlechtesten Schuhe ...

Meine Rettung ist wohl doch nur ein etwas ausführlicheres Requirements-Engineering meiner Wünsche und Lebensgewohnheiten, um dann bei den richtigen Anforderungen zu enden. Damit ich weiß, was ich mir noch so alles wünschen kann, habe ich jetzt Kontakt zu einer Architektin und einem Ansprechpartner eines Smart-Home-Anbieters aufgenommen – und so habe ich vermutlich eine realistische Chance auch bei einem System zu enden, das für mich Sinn ergibt.

Der Requirements-Engineer – Mittler zwischen den Welten

Konzentrieren wir uns nun auf die Person, die das Requirements-Engineering durchführen soll. Der Requirements-Engineer (auch als Systemanalytiker, Anforderungsanalytiker oder Business Analyst bezeichnet) ist ein Mittler zwischen den Welten. In der agilen Entwicklung übernimmt diese Person eine Mittlerrolle – und damit gelten für ihn all die Anforderungen, die wir uns hier für den Requirements-Engineer wünschen. Oftmals wird ein Product-Owner durch eine weitere Rolle – den Product-Owner-Support – unterstützt, für den dann natürlich ebenfalls die folgenden Anforderungen gelten. Wenn Sie in Ihrer Entwicklung (egal ob agil oder eher klassisch) eine Rolle wahrnehmen, die die Bedürfnisse von KundInnen, BenutzerInnen, KäuferInnen ... an die Menschen weiterleitet, die das Produkt designen, erstellen ..., dann sollten Sie sich im Folgenden durch die Bezeichnung Requirements-Engineer angesprochen fühlen.

Als Requirements-Engineer interagieren Sie mit vielen Menschen, vor allem mit den späteren Anwendern des Systems, aber auch mit den SystemarchitektInnen, den EntwicklerInnen, dem Testteam und dem Projektmanagement, und haben großen fachlichen Einfluss auf die Systementwicklung. Ein Requirements-Engineer erhebt und dokumentiert die Wünsche und Anforderungen der Stakeholder an das System oder Produkt, moderiert und vermittelt zwischen Stakeholdern und allen Beteiligten, arbeitet als Katalysator für Entscheidungen der Stakeholder und muss daher großes Fingerspitzengefühl für die Bedürfnisse der Beteiligten besitzen.

Die Rolle des Requirements-Engineers muss nicht immer von einer Person mit Informatikausbildung wahrgenommen werden, um die Bedürfnisse aller Stakeholder erfolgreich ermitteln, analysieren und abgleichen zu können. Es ist also nicht ungewöhnlich, dass neben Entwicklern auch Mitarbeitende des Fachbereichs diese Rolle besetzen.

Wichtig ist nicht die berufliche Herkunft der Person, sondern es sind ihre persönlichen fachlichen und methodischen Kompetenzen, die zählen.

1.1.1 Anforderungen an einen Requirements-Engineer

Für eine derartige Tausendsassa-Rolle braucht man gewisse Eigenschaften oder muss in sie hineinwachsen. Das IREB e.V. (International Requirements Engineering Board e.V.) hält für einen Requirements-Engineer neben der fachlichen und methodischen Kompetenz die folgenden Eigenschaften [CPRE11 und CPRE19] für relevant:

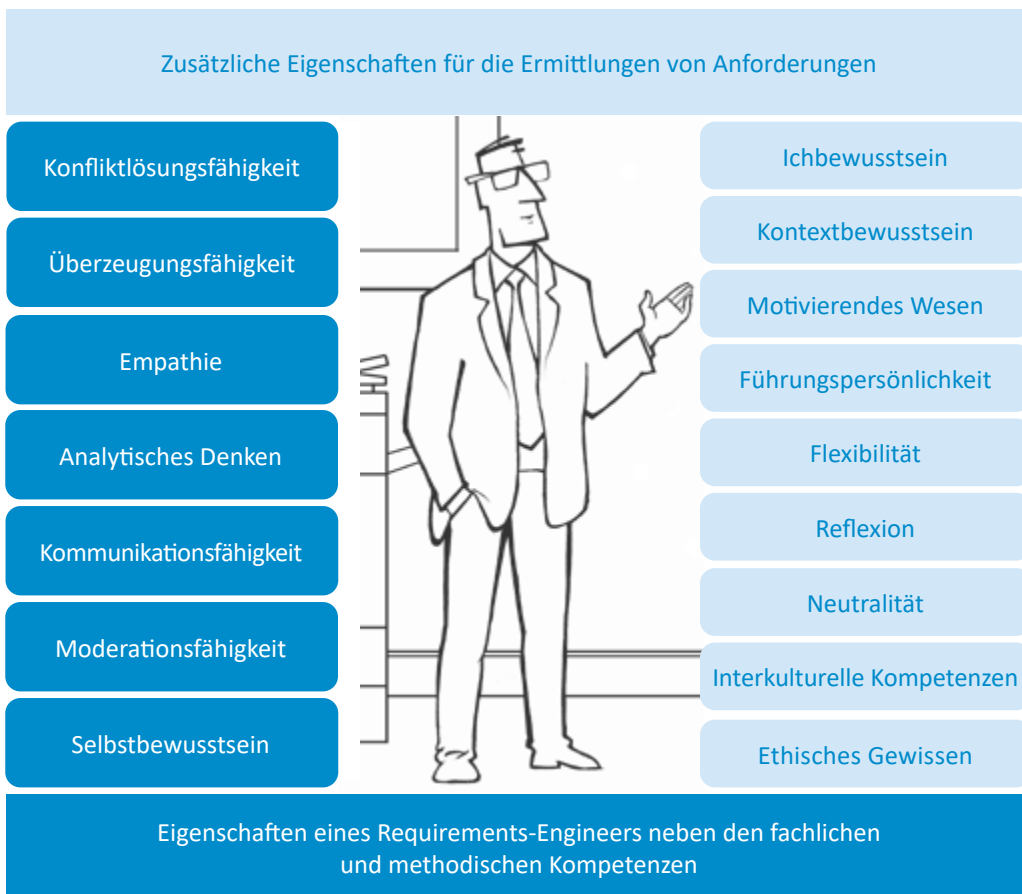


Abbildung 1.1: Eigenschaften eines Requirements-Engineers laut IREB

Mit den vorgestellten Eigenschaften (die einen werden bei Ihnen etwas mehr ausgeprägt sein, andere etwas weniger) und Ihrer fachlichen Kompetenz sind Sie gut ausgestattet. Mit diesem Buch helfen wir Ihnen jetzt noch, Ihre methodischen Kompetenzen zu perfektionieren – damit Sie in Zukunft auf dem Projektparkett glänzen können.