

Inhaltsverzeichnis

Teil I Grundlagen

1	Einführung	3
1.1	VUCA-Welt?	3
1.2	Das Wasserfall-Modell	8
1.3	Der Gegenvorschlag	12
1.4	eXtreme Programming: Praktiken	20
1.5	Scrum als Produktmanagement Framework	23
1.6	Kanban zur Prozessverbesserung	30
1.7	DevOps: Integration des IT-Betriebs	33
1.8	Lean Startup: hypothesenbasiertes Vorgehen	35
1.9	Design Thinking	36
1.10	Einschätzungen und Kritik	39
1.11	Schnelles Lernen	42
	Literatur	42
2	Beginne bei dir	45
2.1	Kopfarbeit	45
2.2	Manage dich selbst!	52
2.3	Was ist Dein Beitrag zum Ganzen?	58
2.4	Agiles Mindset	59
2.5	Schnelles Lernen	64
	Literatur	71
3	Euer Team	75
3.1	Ein Team entsteht: Vom Ich zum Wir	75
3.2	Wie werdet ihr zu einem Team?	84
3.3	Teamkoordination	94

3.4	Alle sind informiert	102
3.5	Schnelles Lernen	108
	Literatur	115
4	Euer Produkt	119
4.1	Was genau ist das Problem?	119
4.2	Was ist ein Produkt?	121
4.3	Unser Beispielprodukt ist ein Projektvorschlagssystem (PVS)	122
4.4	Die Welt verändern	123
4.5	Produktentwicklung Überblick	124
4.6	Systemvision	125
4.7	Personas	129
4.8	Nutzungskontext	133
4.9	SMARTe Ziele	135
4.10	Rahmenbedingungen	140
4.11	Product Backlog	142
4.12	Minimum Viable Product (MVP)	147
4.13	Kontext	149
4.14	Produktlebenszyklus	150
4.15	Risiken	153
4.16	Produktrisiken	154
4.17	Schnelles Lernen	159
	Literatur	161
5	Euer Projekt	165
5.1	Wozu ein Projekt?	165
5.2	Was ist ein Projekt?	166
5.3	Genauere Prozessdefinition	172
5.4	Projektdefinition und -initialisierung	179
5.5	Stakeholder	181
5.6	Initiales Product Backlog	188
5.7	Strategische Planung: Grobstruktur des Projekts	197
5.8	Terminplan	204
5.9	Ressourcen	205
5.10	Projektrisiken	209
5.11	Projektdurchführung	214
5.12	Fortschrittskontrolle	221
5.13	Projektabschluss	225
5.14	Schnelles Lernen	227
	Literatur	228

6	Angebot und Vertrag	231
6.1	Ihr seid Urheber	231
6.2	Ihr braucht einen Vertrag	233
6.3	Vertrags- und Preismodelle	234
6.4	Angebot als Vertragsgrundlage	238
6.5	Vertrags- bzw. Angebotsinhalte	238
6.6	Datenschutz	243
6.7	Open-Source-Software	245
6.8	Schnelles Lernen	249
	Literatur	250

Teil II Handwerkszeug zur Projektdurchführung

7	Effektive Kommunikation	253
7.1	Kommunikation: Nicht nur die Sachebene zählt	253
7.2	Kommunikationspartner spiegeln euch	254
7.3	Schnelles Denken und langsames Denken	255
7.4	Missverständnisse vermeiden	257
7.5	Kommunikation	262
7.6	Kommunikationskanäle	263
7.7	Konflikte	275
7.8	Schnelles Lernen	277
	Literatur	277
8	Effektive Dokumente	279
8.1	Wozu überhaupt noch Dokumente?	279
8.2	Leserkreis und Nutzen definieren	282
8.3	Was ist ein gutes Dokument?	285
8.4	Grafiken	293
8.5	Arbeit mit Literatur- und Internetquellen	297
8.6	Qualitätssicherung	299
8.7	Schnelles Lernen	300
	Literatur	300
9	Effektive Präsentationen	301
9.1	Vorbereitung	301
9.2	Ausarbeitung	306
9.3	Generalprobe	308
9.4	Durchführung	309
9.5	Beispiele für Präsentationen	314
9.6	Schnelles Lernen	315
	Literatur	317

10	Effektive Besprechungen	319
10.1	Wozu Besprechungen?	319
10.2	Vorbereitung: Klare Ziele als Agenda	320
10.3	Durchführung	324
10.4	Nicht vergessen: Nachbereitung	329
10.5	Telefon- und Videokonferenzen	329
10.6	Sprint-Review-Meeting	331
10.7	Schnelles Lernen	334
	Literatur	335
11	Effektive Workshops	337
11.1	Was ist ein Workshop?	337
11.2	Allgemeiner Ablauf eines Workshops	339
11.3	Vorbereitung und Planung eines Workshops	341
11.4	Einstieg	347
11.5	Themen bearbeiten	348
11.6	Aufgaben und Ergebnisse protokollieren	355
11.7	Abschluss	356
11.8	Sonderform: Online-Workshop	356
11.9	Schnelles Lernen	357
	Literatur	358
 Teil III Werkzeugkasten Produktentwicklung		
12	Versions- und Konfigurationsmanagement	363
12.1	Was leistet eine Versionsverwaltung?	363
12.2	Was ist eine Versionsverwaltung?	365
12.3	Zentrale Versionsverwaltungen	367
12.4	Git als verteilte Versionsverwaltung	368
12.5	Was ist eine Konfiguration?	370
12.6	Du arbeitest mit Git	376
12.7	Teamarbeit mit Git und GitLab	381
12.8	GitLab, GitHub und der GitLab-Flow	386
12.9	Schnelles Lernen	388
	Literatur	388
13	Anforderungen sammeln und verwalten	391
13.1	Systematischer Umgang mit Anforderungen – wozu?	391
13.2	Was ist eine Anforderung?	392
13.3	Der PO macht das schon!	395
13.4	Wie gehen wir mit Anforderungen um?	396
13.5	Anforderungen ermitteln, Kontext verstehen	397

13.6	Anforderungen analysieren	405
13.7	Anforderungen spezifizieren	406
13.8	Anforderungen prüfen	414
13.9	Anforderungen verwalten.....	416
13.10	User Stories, Features und Epics.....	420
13.11	Akzeptanzkriterien (Confirmation)	423
13.12	Qualitätsanforderungen und Rahmenbedingungen	427
13.13	Zerlegen von User Stories.....	428
13.14	User Story Maps	430
13.15	Schnelles Lernen	434
	Literatur	436
14	Benutzerschnittstelle	439
14.1	Usability und User Experience	439
14.2	Euer Benutzer	442
14.3	Schnittstelle eures Produkts	448
14.4	Anforderungen an die Benutzerschnittstelle	452
14.5	Dialoge entwerfen	463
14.6	Nutzungskontext verstehen	465
14.7	Ablauf entwerfen: Taskflow	467
14.8	Wireframes und Mockups	468
14.9	Dynamik festlegen und Layout prüfen.....	473
14.10	Prototypen bei Konversation	474
14.11	Usability und User Experience prüfen	476
14.12	Schnelles Lernen	482
	Literatur	483
15	Domänenendesign	485
15.1	Wozu Domänenendesign?.....	485
15.2	Domain Driven Design	489
15.3	Das Domänenmodell entwerfen.....	494
15.4	Das logische Datenmodell entwerfen	497
15.5	Das physische Datenmodell entwerfen	505
15.6	Prozesse betrachten: Event Storming	507
15.7	Schnelles Lernen	509
	Literatur	510
16	Architektur	511
16.1	Was ist Architektur?	512
16.2	Überblick	513
16.3	Analyse: Entscheidungsgrundlagen	515

16.4	Entwurf: Entscheidungen treffen	526
16.5	Planung und Grobstruktur – Logische Architektur	533
16.6	Verteilungsarchitektur	537
16.7	Schichten, Frameworks, Komponenten, Schnittstellen – Technische Architektur	547
16.8	Code, Namensräume, Build-Skripte – Implementierungsarchitektur	554
16.9	Laufzeitarchitektur	557
16.10	Querschnittsthemen	560
16.11	Architektur kommunizieren	562
16.12	Bewertung: Entscheidungen überprüfen	564
16.13	Schnelles Lernen	567
	Literatur	567
17	Entwickeln	571
17.1	Ziel: Sauberer Code	571
17.2	Automatisierung und Werkzeuge	574
17.3	Architektur in Quelltexte umsetzen	580
17.4	Was genau ist guter Code?	584
17.5	Technische Schulden	588
17.6	Sauberen Code absichern: Statische und Dynamische Codeanalyse	590
17.7	Sauberen Code absichern: Code-Reviews	592
17.8	Test Driven Development (TDD)	594
17.9	Probleme beheben: Refaktorisierung	600
17.10	Emergentes Design	602
17.11	Definition of Done mit verbindlichen Qualitätskriterien	602
17.12	Schnelles Lernen	603
	Literatur	605
18	Testen	607
18.1	Vertrauen in die Software gewinnen	607
18.2	Testfälle finden	616
18.3	Test- und Demodaten finden	619
18.4	Testumgebungen definieren	622
18.5	Exploratives Testen	623
18.6	Testautomatisierung	627
18.7	Test der Qualitätseigenschaften	633
18.8	Fehlermanagement	635
18.9	Schnelles Lernen	640
	Literatur	642

19	Deployment und Betrieb	643
19.1	Was ist IT-Betrieb?	644
19.2	DevOps, SRE und ITIL®	644
19.3	Anforderungen an den IT-Betrieb	647
19.4	Basistechnik: Virtualisierung	651
19.5	Konfigurationsmanagement	657
19.6	Deployment	660
19.7	Automatisierung	663
19.8	Messen	670
19.9	Problemmanagement	676
19.10	Weitere Themen des IT-Betriebs	678
19.11	Schnelles Lernen	679
	Literatur	680
	Stichwortverzeichnis	683