

Inhaltsverzeichnis

1	Software-Prozessmodelle	1
1.1	Ein Prozessmodell – was ist das?	1
1.1.1	Einteilung in Projekttypen	2
1.1.2	Schwergewichtige und leichtgewichtige Prozessmodelle	3
1.1.3	Prozessmodell vs. Vorgehensmodell	3
1.2	Das Phasenmodell	4
1.3	Das Spiralmodell	4
1.4	Moderne iterativ-inkrementelle Prozessmodelle	6
1.4.1	V-Modell	6
1.4.2	Unified Software Development Process (UP)	6
1.4.3	Agile Prozessmodelle	7
1.5	Die HELENA-Studien	7
	Literatur	8
2	Das agile Manifest	11
	Literatur	13
3	Extreme Programming (XP)	15
3.1	Die fünf Werte	16
3.1.1	Kommunikation	16
3.1.2	Einfachheit	16
3.1.3	Feedback	17
3.1.4	Mut	18
3.1.5	Respekt	19
3.2	Die 14 Prinzipien	19
3.2.1	Menschlichkeit	19
3.2.2	Wirtschaftlichkeit	20
3.2.3	Wechselseitiger Vorteil	20
3.2.4	Selbstähnlichkeit	20
3.2.5	Verbesserung	20
3.2.6	Vielfältigkeit	21

3.2.7	Reflexion	21
3.2.8	Fluss	21
3.2.9	Gelegenheit	21
3.2.10	Redundanz	22
3.2.11	Fehlschlag	22
3.2.12	Qualität	22
3.2.13	Kleine Schritte	22
3.2.14	Akzeptierte Verantwortung	22
3.3	Prozessschritte und traditionelle XP-Praktiken	23
3.3.1	Planung	23
3.3.2	Design der Software	29
3.3.3	Kodieren	32
3.3.4	Testen der Software	38
3.4	Erweiterte XP-Praktiken	40
3.4.1	Primärpraktiken	40
3.4.2	Folgepraktiken	43
3.4.3	Unterschiede zwischen erweiterten und traditionellen XP-Praktiken	45
3.5	Berühmte XP-Praktiken	46
3.5.1	Erstellen von User Stories	46
3.5.2	Paarweises Programmieren	46
3.5.3	Collective Code Ownership, Shared Code	46
3.5.4	Kontinuierliche Code-Integration, eine Codebasis	47
3.5.5	Kunde im Team, Einbeziehung des Kunden	47
	Literatur	47
4	Crystal und Crystal Clear	49
4.1	Teamgröße und Risiko – die Crystal-Familie	50
4.2	Die sieben Crystal-Eigenschaften	50
4.2.1	Regelmäßige Lieferung	51
4.2.2	Reflektierte Verbesserung	51
4.2.3	Verdichtete oder osmotische Kommunikation	52
4.2.4	Persönliche Sicherheit	53
4.2.5	Schwerpunkte bilden	53
4.2.6	Einfache Kontaktaufnahme mit Endanwendern	54
4.2.7	Technische Umgebung mit automatisierten Tests, Konfigurationsmanagement und regelmäßige Integrationen	55
4.3	Crystal Clear	55
4.3.1	Eigenschaften und Praktiken	55
4.3.2	Rollen im Projektteam	58
	Literatur	61

5	Scrum	63
5.1	Scrum – die Projektrollen	64
5.1.1	Product Owner	64
5.1.2	Team	65
5.1.3	ScrumMaster	67
5.1.4	Weitere Scrum-Rollen	69
5.1.5	Gefahr durch Rollenmissbrauch	70
5.2	Scrum – der Prozess	70
5.2.1	Scrum-Flow – Überblick	70
5.2.2	Sprint – Details	72
5.3	Scrum-Artefakte	76
5.3.1	Product Backlog	77
5.3.2	Sprint Backlog	77
5.3.3	Releaseplan und Burndown Chart	79
	Literatur	80
6	Kanban	81
6.1	Software-Kanban	82
6.2	Das Kanban-Board	82
6.3	Kanban – die vier Prinzipien	84
6.4	Die sechs Kanban-Kernpraktiken	85
6.5	Praktiken zur Prozessverbesserung	87
6.5.1	Der PDCA-Zyklus	87
6.5.2	Die sieben Muda	88
6.6	Kanban im Projekt	90
6.6.1	Ein Kanban-Board aus der Praxis	90
6.6.2	Die Kunst der Limitierung der Work in Progress (WIP)	93
6.6.3	Kanban-Metriken	95
	Literatur	98
7	DevOps	99
7.1	CALMS – Die Grundprinzipien von DevOps	100
7.1.1	Kultur (Culture)	100
7.1.2	Automatisierung (Automation)	101
7.1.3	Lean	101
7.1.4	Messung (Measurement)	101
7.1.5	Teilen (Sharing)	101
7.2	DevOps Life Cycle	102
7.3	„Die drei Wege“ – The Three Ways	103
7.4	Eine kritische Betrachtung	104
	Literatur	104

8	Experimentelles Software-Engineering im studentischen Labor	107
8.1	Die Projekte im studentischen Labor	108
8.2	Randbedingungen der studentischen Sessions	110
8.3	Veränderung der klassischen Projektrollen in agilen Projekten	112
8.3.1	Der Projektmanager im agilen Projekt	112
8.3.2	Der Qualitätsmanager im agilen Projekt	116
8.4	Neue Teamrolle – der Integrationsingenieur	120
8.5	Veränderung agiler Praktiken und Prozesse in der Praxis	121
8.5.1	Design von User Stories	121
8.5.2	Collective Code Ownership	122
8.5.3	Mini-Team-Größe: Sind XP-Paare erfolgreich?	122
8.5.4	Der Weg zur erfolgreichen Software-Integration	129
8.5.5	Crystal und die reflektierte Verbesserung	130
8.5.6	Scrum im studentischen Labor	134
8.5.7	Hierarchische Prozesse unter dem agilen Deckmantel	142
	Literatur	144
9	MAP – Meta Agile Process Model	145
9.1	Team-Psychologie –Landkarte der Verhaltensweisen im Team	145
9.2	Existenz aller MAP-Typen im Team	148
9.2.1	Das Super-Team im studentischen Labor	148
9.2.2	Vergleich mit den Team-Rollen nach Belbin	149
9.3	Was ist MAP?	150
9.4	Das MAP-Team als System – Integration in das Viable System Model	151
9.5	MAP – die Projektrollen im Team	156
9.5.1	Kunde	156
9.5.2	Kommunikationsmanager	156
9.5.3	Integrationsingenieur	158
9.5.4	Team	158
9.5.5	Prozessverantwortlicher – der MAP-Beobachter	160
9.5.6	Projektrollen in der Landkarte der Verhaltensweisen	161
9.6	MAP Cycle – der Referenzprozess	162
9.7	MAP in der Praxis	164
9.8	Konkurrierende Teams – Vorhersage des Teamerfolgs mit MAP	166
9.9	MAP und Scrum	168
9.9.1	Vergleich der Rollen	169
9.9.2	Bestimmung des Product Owners	172
9.9.3	Prozess	174
9.9.4	Artefakte	176
9.9.5	MAP und Scrum – geht das?	177
9.9.6	Vorteile für Scrum-Anwender	178
9.10	MAP im regulierten Umfeld	179
9.11	MAP-Projekte in der chemischen und der pharmazeutischen Industrie	181
	Literatur	182