

Inhaltsverzeichnis

Teil I Digitale Transformation mit Large-Scale Agile Frameworks

1	Einleitung	3
	Literatur	8
2	Digitale Transformation & Agile Priorisierung	9
2.1	Agile Modelle zur Organisation digitaler Transformation	9
2.1.1	Digitale Transformation – Herausforderung mit vielen Chancen	10
2.1.2	Welche Faktoren beeinflussen digitale Transformation?	12
2.1.3	Ziele digitaler Transformation	14
2.1.4	Digitale Transformation und Large-Scale Agile Frameworks	16
2.1.5	Die Bedeutung agiler Prozesse und Large-Scale Agile Frameworks	18
2.2	Wie bewährt sind agile Vorgehensmodelle?	20
2.3	Agile Konzepte und grundlegende Begriffe	21
2.3.1	Multi-Projekt-Management	22
2.3.2	Portfolio-Management	24
2.3.3	Software-Lizenzprodukte und Individualsoftware	25
2.3.4	Software-Plattform, Software-Produktfamilie und Software-Produktlinie	28
2.3.5	Produkt-Lebenszyklus von Software	30
2.3.6	Software Releases und Release Management	32
2.3.7	Software-Architektur und Wissensmanagement	34
2.3.8	Requirement Management/Priorisierung	35
2.3.9	DevOps und DevOps-Modell	37
2.3.10	Agilität und Agiles Projekt-Modell	38
2.3.11	Scaled Agile/Large-Scale Agile Development	44
	Literatur	45
3	Large-Scale Agile Frameworks	49
3.1	Bewertungskriterien für Large-Scale Agile Frameworks	49
3.2	Ausgewählte Scaled Agile Frameworks	51

3.3	Domänenorientiertes Modell/Domain-Driven Design	52
3.4	Spotify Engineering Model	54
3.5	Scaled Agile Framework (SAFe)	60
3.6	Vergleich der drei ausgewählten Large-Scale Agile Frameworks	63
	Literatur	65
4	Wie Sie ein Large-Scale Agile Framework anpassen und in Ihrer Organisation einführen	67
4.1	Action Design Research	68
4.1.1	Individuelle Anpassungen für ein Large-Scale Agile Framework definieren	72
4.1.2	Ist-Analyse im Transformationsprozess mittels Evaluation	73
4.1.3	Befragungsdesign	76
4.1.4	Auswertung und Ergebnisse der formativen Evaluation	78
4.1.5	Summative Evaluation	78
4.1.6	Experteninterview	79
4.2	Einflussfaktoren des Cloud-Trends und der Virtualisierung im Fokus eines agilen Frameworks berücksichtigen	79
4.2.1	Cloud Computing	80
4.2.2	Cloud-Eigenschaften	81
4.2.3	Cloud-Service-Modelle	81
4.2.4	Cloud-Modelle	82
4.2.5	Virtualisierung & Containerisierung	85
4.2.6	Relevante Gremien zur Etablierung von Internet- und Cloud-Standards	89
4.3	Software-Architektur & IT-Security als integraler Bestandteil eines agilen Frameworks	92
4.3.1	IT-Security-by-Design: Software-Architektur & IT-Security	93
4.3.2	Zero-Trust-Strategie	94
4.3.3	Schutzprinzipien und deren technische Umsetzung auf Basis von Zero Trust	96
4.3.4	Secret-Management	100
4.3.5	Erweiterter Schutzbedarf für virtuelle Container-Umgebungen	100
4.3.6	Key-Management & kryptografische Schutzmaßnahmen	101
4.3.7	Public-Key-Infrastrukturen	104
4.3.8	Microservice-Architekturen	105
4.3.9	APIs, Ressourcen und dynamische IP-Adressen in Cloud-Netzwerken	108
4.3.10	APIs und REST	109
4.3.11	Qualitätsmerkmale von Microservices und Web-APIs	109
4.3.12	RESTful API	113

- 4.3.13 Fazit und Bezug von APIs im Hinblick auf Large-Scale Agile Frameworks 115
 - 4.3.14 Service Mesh & Agile Microservice-Architekturen 116
 - 4.3.15 IT-Security auf Grundlage der OWASP Guidelines verbessern. . . 118
 - 4.3.16 Penetrationstests/Pentesting. 118
 - 4.3.17 Empfehlungen um IT-Security als festen Bestandteil in einem agilen Framework zu integrieren 122
 - 4.3.18 Threat Modeling. 130
 - 4.3.19 Frühestmöglich zum agilen Deployment & automatisierten Tests . . 134
 - 4.3.20 DevSecOps. 137
 - 4.3.21 Code Reviews 138
 - 4.3.22 Pair Programming 139
 - 4.3.23 Logging & Monitoring. 141
- 4.4 Agile Teams, Rollen, Aufgaben und Prozesse 143
 - 4.4.1 Agile Software-Entwicklungs-Teams 144
 - 4.4.2 IT-Security-Teams 146
 - 4.4.3 Umsetzung rechtlicher Rahmenbedingen und Datenschutz 147
 - 4.4.4 Software-Produktmanagement/Service-Management 148
 - 4.4.5 UX-Teams: Frontend Design, Usability und User Experience . . . 148
 - 4.4.6 Qualitätssicherung und Testverfahren 149
 - 4.4.7 Technische Redaktion 150
 - 4.4.8 Infrastruktur-Teams 151
 - 4.4.9 DevOps-Teams. 152
 - 4.4.10 Sales – Der Vertrieb von Produkten und Dienstleistungen Ihrer Organisation. 154
 - 4.4.11 Forschung, Innovation und PreSales 155
 - 4.4.12 Systemintegration 155
 - 4.4.13 Support-Teams 157
- 4.5 Mit Design Thinking & Prototyping durchstarten. 158
 - 4.5.1 Prototyping und Rapid Prototyping 160
 - 4.5.2 Prototyping-Phasenmodell. 162
- 4.6 Übergeordnete agile Prozessphasen auf eine prototypische Vorgehensweise ausrichten 163
 - 4.6.1 Phase 1: Anforderungserhebung und Ideenfindung 164
 - 4.6.2 Phase 2: Rapid Prototyping und Beratungsprozess 167
 - 4.6.3 Phase 3: Testmanagement und summative Evaluation 173
 - 4.6.4 Anmerkungen und Empfehlungen zum prototypischen Vorgehensmodell 175
- Literatur. 176
- 5 Agiles Priorisierungsmodell für Software-Hersteller 179**
 - 5.1 Inwieweit agiert Ihre Organisation in der Rolle eines Software-Herstellers?. 180

5.1.1	Beispiel 1: Ist LEGO ein Software-Hersteller?	182
5.1.2	Beispiel 2: Flaschenpost.de – innovativer, App-getriebener Getränkesservice	185
5.1.3	Beispiel 3: Moia – digitaler Shuttle Service	187
5.2	Agile Teams und Rollen bei Software-Herstellern	189
5.3	Prozesse und Aktivitäten	193
5.4	Fazit	201
5.5	Schlussfolgerungen	201
5.6	Ausblick.	205
5.6.1	Implikationen für die Praxis.	205
	Literatur.	207

6 Herausforderungen bei der Etablierung eines Large-Scale Agile

	Frameworks im Unternehmen.	209
6.1	Start-up-Wind in etablierte Organisationen bringen	209
6.1.1	Agile Software-Service-Entwicklung	211
6.1.2	Friendly User Tests (FUT).	211
6.1.3	Unterschiedliche Agilitätsgrade der Teams	212
6.1.4	Neuartige Entwicklungs- und Testumgebungen mit abstrahierter Hardwareebene etablieren	212
6.1.5	Transparente Entscheidungsprozesse und agiles Anforderungsmanagement.	213
6.2	Digital Leadership	215
6.2.1	Vision, Strategie und Product/Service Roadmap.	217
6.2.2	Chief Information Officer	218
6.2.3	Chief Digital Officer	220
6.3	Change-Management – Digital Leadership im Management	223
6.3.1	Perspektiven des Change-Management	223
6.3.2	Change-Request-/Release-Management	224
6.3.3	Agile Academy.	225
6.3.4	Agile Werte etablieren	225
6.4	Notfallmanagement: Kann Ihre Organisation agil in Notfallsituationen reagieren?.. . . .	226
6.4.1	Wie können Sie Notfällen proaktiv begegnen?	228
6.4.2	Wie gehen wir ein Notfallkonzept organisatorisch an?	229
	Literatur.	231

Teil II Agile Infrastruktur

7	Agile Tools: Werkzeugkasten für Produkt Owner & Agile Teams.	235
7.1	Agiles Mindset & die 12 Prinzipien des agilen Manifests.	237
7.2	Agile Ziele	242

- 7.2.1 Personas 243
 - 7.2.2 User Story 244
 - 7.2.3 Epic 248
 - 7.2.4 Task & Sub-Task 250
 - 7.2.5 Backlog 251
- 7.3 Collaboration Tools 252
 - 7.3.1 Confluence 254
 - 7.3.2 Jira 256
 - 7.3.3 Git-Repositories, GitHub und GitLab 256
 - 7.3.4 OpenAPI – Werkzeuge zum API-Design. 261
 - 7.3.5 Messenger und Chat-Systeme 265
- 7.4 Agile Architekturen – Fundament softwarebasierter Digitalisierung 267
 - 7.4.1 Einflussfaktoren für agile IT-Architekturen. 268
- 7.5 Pragmatische Software-Architektur-Dokumentation. 269
 - 7.5.1 Wie Sie visuelle Software-Architektur-Artefakte erstellen 270
 - 7.5.2 Standards, Kriterien und Normen für Software- Architektur-Artefakte 271
 - 7.5.3 Adressaten und Themenfelder der Software- Architektur-Dokumentation. 273
 - 7.5.4 Fragestellungen für Software-Architektur- Entscheidungen formulieren. 274
 - 7.5.5 Technische Schulden 277
 - 7.5.6 Arc42-Template zur Software-Architektur-Dokumentation 278
 - 7.5.7 ISO/IEC 25010 – Qualität von Software 280
- 7.6 DevOps-Methoden und DevOps-Tools 282
 - 7.6.1 Das DevOps-Periodensystem. 282
 - 7.6.2 DevOps ist mehr als = Software Engineering + IT-Management .. 282
 - 7.6.3 BizDevOps als Konsequenz für agile Unternehmen 285
- 7.7 Content. 288
 - 7.7.1 Ziele einer Content-Strategie 289
 - 7.7.2 Anforderungen an den Content 290
 - 7.7.3 Content Controlling 290
 - 7.7.4 Content-Prozess und Koordination 291
 - 7.7.5 Content-Leitfaden 291
 - 7.7.6 Design-System. 291
 - 7.7.7 Infografiken 294
 - 7.7.8 Dialogbilder 296
- 7.8 Monitoring & Controlling 298
 - 7.8.1 Key-Performance-Indikatoren (KPI) 299
 - 7.8.2 Monitoring 300
 - 7.8.3 Strategien und Lösungswege zu digitaler Transformation. 300
 - 7.8.4 Die optimale Architektur für die digitale Organisation 302

7.9	Methoden & Werkzeuge zur agilen Priorisierung	302
7.9.1	Agile Anforderungspriorisierung mit dem Feature Graph.	303
7.9.2	Agile Priorisierung mit der Einpunktabfrage.	304
7.9.3	Agile Priorisierung mit der Mehrpunktabfrage	304
7.9.4	Agile Priorisierung mit Liquid Democracy	306
7.9.5	Timeboxing	306
7.9.6	Feature-Driven Development.....	307
	Literatur.	309
8	Datenqualität – Lebenselixier der Digitalisierung.	311
8.1	Kundenindividuelle Produkte und Dienstleistungen auf Basis von Smart Data.	313
8.2	Datenqualität: Grundlagen und Voraussetzungen	314
8.2.1	Datenabhängigkeiten	315
8.2.2	Datenreparatur	318
8.2.3	Datendeduplikation	320
8.2.4	Informationsvollständigkeit.....	322
8.2.5	Datenaktualität	323
8.2.6	Datenpräzision	323
8.3	Techniken zur Datenbereinigung	323
8.3.1	Qualitätsregeln für Daten entdecken	324
8.3.2	Fehlererkennung für Daten	324
8.3.3	Datenreparatur	324
8.4	Kontrollmechanismen zur Optimierung der Datenqualität sind unverzichtbar	326
8.4.1	Mit Machine Learning zu Smart Data	326
8.4.2	Mit Smart Data Innovationen aus Daten gewinnen	327
8.4.3	Semantische Datenanalyse.	328
8.4.4	Framework für Data-Driven Design	329
8.4.5	Fazit	330
	Literatur.	331