

Inhaltsverzeichnis

Kapitel 1	Einleitung	13
1.1	Ziel dieses Buches: Moderne Software mit Qualität	15
1.2	Aufbau des Buches und Lesepfade	19
1.3	Werkentstehung und Companion Website.	23
Kapitel 2	Ingenieurdisziplin Softwaretechnik und die Bausteine eines Projektes	27
2.1	Wie reif ist die Softwaretechnik als Ingenieurdisziplin?	28
2.2	Industrielle Konsequenzen aus dem Reifegrad	33
2.3	Softwaretechnik als Verfahren der Übersetzung, Entscheidung und Schlichtung zwischen unterschiedlichen Denkweisen.	42
2.4	Definition eines (Software-)Projektes.	53
2.5	Frequently Asked Questions (FAQ) für Novizen.	62
2.6	Die charakteristischen Merkmale von Softwareprojekten.	73
2.7	Software = Peopleware	93
Kapitel 3	Fallbeispiele von Projekten	101
3.1	LiSaMe.	102
3.1.1	Organisatorische Aspekte	104
3.1.2	Aspekte der technischen Konzeption.	106
3.2	Stadt21.	115
3.2.1	Das Projekt Stadt21	116
3.2.2	Allgemeiner Aufbau	117
3.2.3	Prozesserstellung.	118
3.2.4	Aktenbearbeitung	121
3.2.5	Administration	122
3.2.6	Entwicklungsumgebung	122
3.3	Touch&Go	123
3.3.1	Technische Anforderungen.	123
3.3.2	Architektur des Touch&Go Services.	126
3.3.3	Integration unterschiedlicher Technologien	128
3.3.4	Schnittstellen und Integration mit Bestandssystemen	129
3.3.5	Organisatorische Aspekte	130
3.3.6	Entwicklungsumgebung	131
3.4	HISS.	131
3.4.1	Projektentstehung	132
3.4.2	Ziele und Umfang von HISS	132
3.4.3	Phasen und grober Aufbau	133
3.4.4	Motivation und Ziele des Fachkonzeptes	134
3.4.5	Vorgehensweise bei der fachlichen Analyse	135
3.4.6	Die Softwareentwicklung von HISS	136

Kapitel 4	Phase Analyse – Was wird gebaut?	139
4.1	Grundlagen der Analyse	142
4.1.1	Vision und Scope	142
4.1.2	Stakeholder und Anforderungsquellen.	143
4.1.3	Machbarkeit und Machbarkeitsstudie.	146
4.1.4	Make or Buy	146
4.1.5	Grobarchitektur	148
4.1.6	Anforderungsvolatilität	149
4.1.7	Klassifikation von Anforderungen	151
4.1.8	Was sind gute Anforderungen?	153
4.2	Anforderungsmodellierung	155
4.2.1	Unified Modeling Language (UML).	155
4.2.2	DSLs	160
4.2.3	Datenmodellierung.	161
4.3	Anforderungsanalyse.	163
4.3.1	Anwendungsfälle (Use Cases)	163
4.3.2	User Stories/Kunde vor Ort	166
4.3.3	Geschäftsprozessmodellierung	168
4.3.4	Entwurf und Prototyping der Anwenderschnittstelle	172
4.3.5	End User Programming	173
4.4	Anforderungsdokumentation	175
4.4.1	Projektglossar	177
4.4.2	Anforderungsverwaltung.	179
4.4.3	Nachverfolgbarkeit von Anforderungen (Traceability)	180
4.5	Organisatorische Aspekte der Analyse.	181
4.5.1	Die Rolle des Analytikers	182
4.5.2	Stakeholder identifizieren	184
4.5.3	Ziele definieren	188
4.5.4	Anforderungen ermitteln.	188
4.6	Glossar	195
Kapitel 5	Phase Entwurf – Wie wird technisch gebaut?	197
5.1	Grundlagen der Softwarearchitekturen	199
5.1.1	Definitionen zu unterschiedlichen Begriffen	199
5.1.2	Von der Anforderung zur Architektur	203
5.1.3	Softwarearchitektur im Wandel der Zeit	204
5.1.4	Grundlegende Architekturmuster	207
5.1.5	Architekturbausteine und deren Kommunikation	214
5.1.6	Architektur-Frameworks	217
5.2	Entwurfsparadigmen	219
5.2.1	Objektorientiertes Design.	219
5.2.2	Serviceorientiertes Design	223
5.3	Architekturmuster	229
5.3.1	Architektur-Patterns.	230
5.3.2	Design-Patterns.	232

5.4	State-of-Art Softwarearchitektur: serviceorientierte Architektur und Design	233
5.4.1	SOA-Referenzarchitekturen	233
5.4.2	Granularität	236
5.4.3	Service-Design-Paradigmen und Benefits	238
5.4.4	Service-Modellierung	239
5.5	Glossar	242

Kapitel 6 Phase Implementierung – Wie wird codiert? 245

6.1	Grundlagen der Implementierung	247
6.1.1	Die Wahl der Programmiersprache	248
6.2	Frameworks als technische Basis der Implementierung	249
6.2.1	Warum werden Frameworks eingesetzt?	250
6.2.2	Abgrenzungen zwischen Framework und Bibliothek	251
6.2.3	Designentscheidung: Welches Framework soll eingesetzt werden?	253
6.2.4	Eigenschaften von Frameworks	254
6.3	Die Entwicklungsumgebung (IDE) als Cockpit des Entwicklers	258
6.4	Konzepte für eine wartungsfreundliche Implementierung	262
6.4.1	Best Practices für die Strukturierung des Quellcodes	262
6.4.2	Kollektiver Codebesitz (Collective Code Ownership)	266
6.4.3	Automatisierte Entwicklungstests	267
6.4.4	Refaktorisierung	269
6.4.5	Dokumentation der Entwicklungsartefakte	273
6.4.6	Implementierungsrichtlinien	276
6.4.7	Internationalisierung in der Softwareentwicklung	277
6.4.8	Fehlerbehandlung (Exception Management)	280
6.4.9	Laufzeitprotokollierung (Logging)	282
6.5	Konfigurationsmanagement in der Implementierung	283
6.5.1	Build Management	284
6.5.2	Source Code-Management	285
6.5.3	Vergabe von Programmversionsnummern	293
6.6	Glossar	294

Kapitel 7 Phase Integration/Test – Wie wird zusammengefügt/geprüft 297

7.1	Bedeutung von Softwaretests	299
7.2	Grundlagen der Integration und des Tests	301
7.2.1	Softwaretests – Begriffserklärung	301
7.2.2	Ziele und Grundsätze des Softwaretests	302
7.3	Test- und Integrationsstufen	305
7.3.1	Komponententest	307
7.3.2	Integrationstest	308
7.3.3	Systemtest	312
7.3.4	Akzeptanztest	313

7.4	Funktionale Softwaretests	315
7.4.1	Strukturelle Methoden	315
7.4.2	Funktionale Methoden	320
7.5	Nichtfunktionale Softwaretests	327
7.5.1	Performance-Tests	327
7.6	Testautomatisierung	332
7.6.1	Automatisierte Komponententests	332
7.6.2	Automatisierte GUI-Tests	333
7.6.3	Regressionstest	334
7.6.4	Testdaten	335
7.7	Testprozess	337
7.7.1	Fehlermanagement	342
7.7.2	Software Test Outsourcing	344
7.7.3	Standards und Normen im Softwaretest	345
7.8	Glossar	347

Kapitel 8 Projektphasen: Inbetriebnahme, Rollout und Wartung 349

8.1	Integration der Systemanwender	351
8.1.1	Anforderungen an die Anwenderdokumentation	351
8.1.2	Planung und Durchführung von Schulungsmaßnahmen	352
8.1.3	Szenario: Applikations-Rollout	353
8.2	Inbetriebsetzung und laufender Betrieb	354
8.2.1	Planung, Installation und Integration	355
8.2.2	Abnahme komplexer Systeme	356
8.2.3	Überwachung und Optimierung	357
8.2.4	Checkliste: Betrieb „light“	358
8.3	Migration von existierenden Systemteilen und Daten	358
8.3.1	Migration vs. Integration	358
8.3.2	Migrations- und Integrationsstrategien	359
8.4	Softwarewartung	361
8.4.1	Wartungstypen	362
8.4.2	Wartungsprozesse	363
8.4.3	Softwarewartung als Investitionsschutz	364
8.4.4	Maßnahmen während der Wartung	365
8.5	Glossar	367

Kapitel 9 Vorgehensmodelle im Software Engineering 369

9.1	Grundlagen und Bedeutung von Vorgehensmodellen	371
9.1.1	Die Entstehungsgeschichte der Prozessmodelle	372
9.2	Fundamentale Prozessmodelle	373
9.2.1	Wasserfallmodell	374
9.2.2	V-Modell	375
9.2.3	Spiralmodell	376
9.3	Industriell geprägte Prozessmodelle	379
9.3.1	Rational Unified Process (RUP)	379
9.3.2	Microsoft Solutions Framework (MSF)	384

9.4	Von Personen gepragte Prozessmodelle	387
9.4.1	Scrum	387
9.4.2	eXtreme Programming (XP)	391
9.5	Von Gremien/Konsortien gepragte Prozessmodelle	398
9.5.1	V-Modell 97	398
9.5.2	V-Modell XT	402
9.5.3	ISO/IEC 12207	406
9.6	Diskussion: agile versus traditionelle Vorgehensmodelle	409
9.7	Glossar	412

Kapitel 10 Projektmanagement und Risikomanagement 413

10.1	Schatzung von Entwicklungsaufwanden	415
10.1.1	Einflussfaktoren	416
10.1.2	Empirische Verfahren	418
10.1.3	Komparative Verfahren	419
10.1.4	Algorithmische Verfahren	420
10.1.5	Schatzproblematik	426
10.2	Planung von Ergebnissen und Projektaktivitaten	427
10.2.1	Ablauf und Bestandteile der Projektplanung	428
10.2.2	Fortschreibung der Projektplanung	439
10.3	Berichtswesen	440
10.4	Projekt-Controlling	441
10.4.1	Qualitatskontrolle	443
10.4.2	Terminkontrolle	444
10.4.3	Kostenkontrolle	444
10.4.4	Risikokontrolle	445
10.4.5	Earned Value Management	445
10.4.6	Controlling-Arten	448
10.4.7	Typische Probleme beim Projekt-Controlling	448
10.5	Risikomanagement	449
10.5.1	Was ist Risikomanagement?	449
10.5.2	Typische Risiken in der Softwareentwicklung	452
10.6	Glossar	454

Kapitel 11 Qualitatssicherung und Qualitatsmanagement 457

11.1	Grundlagen der Softwarequalitatssicherung	460
11.1.1	Softwarequalitatsfaktoren	463
11.1.2	Softwarefehler, -fehlerfall und -fehlerverhalten	465
11.1.3	Klassifikation der Ursachen von Softwarefehlern	466
11.1.4	Fehlerentstehung und Fehlerentdeckung	467
11.1.5	Kosten der Qualitatssicherung	468
11.2	Statische Qualitatssicherung	470
11.2.1	Statische Analyse	470
11.2.2	Softwaremetriken	475
11.2.3	Reviews	481

11.3	Dynamische Qualitätssicherung	499
11.3.1	Testen	499
11.3.2	Dynamische Analyse	500
11.4	Organisatorische Qualitätssicherung	500
11.4.1	Wissensmanagement	501
11.4.2	Konfigurationsmanagement.	502
11.4.3	Templates und Checklisten	503
11.5	Qualitätsmanagementstandards	503
11.5.1	ISO 9001 und ISO 9000-3	505
11.5.2	Capability Maturity Models – CMM and CMMI.	508
11.5.3	SPICE – ISO/IEC 15504	512
11.6	Glossar	517
Kapitel 12 Usability Engineering in der Softwareentwicklung		519
12.1	Usability Engineering als zentraler Bestandteil des Softwareentwicklungsprozesses	521
12.1.1	Usability Engineering.	521
12.1.2	User-Centered Designprozess	524
12.1.3	Erfolgsfaktor Usability Engineering.	526
12.2	Anwendbare Methoden im User-Centered Design.	528
12.2.1	Anforderungen	529
12.2.2	User Interface und Interaktionsdesign	537
12.2.3	Prototyping.	541
12.2.4	Evaluierung	549
12.3	Web Usability.	568
12.3.1	Ausgewählte Gestaltungshinweise für Web Usability	570
12.3.2	Accessibility.	578
12.4	Usability Engineering im realen Projekt.	581
12.4.1	Stadt21	582
12.4.2	HISS	584
12.5	Glossar	587
Kapitel 13 Sicherheit in der Softwareentwicklung		589
13.1	Grundlagen der IT-Sicherheit	591
13.2	Vorgehensmodelle zur Entwicklung sicherer Software	595
13.3	Sicherheitsanforderungen in der Analyse	596
13.3.1	Erfassung Sicherheitsanforderungen.	596
13.3.2	Bedrohungs- und Risikoanalyse	598
13.3.3	Sicherheit als Prozess	604
13.4	Sicherheitsaspekte beim Entwurf.	604
13.4.1	Designprinzipien für sichere Software	609
13.4.2	Sicherheitsmuster	613
13.4.3	Angriffsmuster	613
13.4.4	Modellierung mittels UML	614

13.5	Sichere Implementierung	614
13.5.1	Programmierrichtlinien.	614
13.5.2	Sicherheitskonzepte von Programmiersprachen	615
13.5.3	Ausgewählte Angriffsmöglichkeiten	616
13.6	Sicherheitstest	627
13.6.1	Sicherheitsfunktionalität vs. sichere Funktionalität	627
13.6.2	Sicherheitstests im Softwarelebenszyklus	629
13.6.3	Analyse des Testobjekts	629
13.6.4	Testtechniken zur Überprüfung von Sicherheitsfehlern	630
13.7	IT-Sicherheit bei Betriebsübergabe und Betrieb	636
13.8	Standards, Normen und Leitfäden	643
13.9	Glossar	645
 Kapitel 14 Exemplarische Vertiefungen		 647
14.1	Mobile Usability	649
14.1.1	Zielgruppe	649
14.1.2	Informationsdarstellung	651
14.1.3	Interaktionsformen	652
14.1.4	Kontext.	654
14.1.5	Aktuelle Entwicklungen	655
14.2	Cloud Computing	656
14.3	Scientific Computing	658
14.4	IT-Sicherheit für Software im VoIP-Umfeld	659
14.4.1	Sicherheitsanforderungen an VoIP	661
14.4.2	Beispiele für Bedrohungen der IT-Sicherheit bei VoIP	661
14.4.3	Kernpunkte der Sicherheit bei VoIP	662
 Literatur		 665
 Register		 681