

Inhalt

Vorwort — V

Danke — IX

1 Offerte und Vertrag — 1

- 1.1 Der Weg zum Vertragsabschluss — 1
- 1.2 Finanzielle Aspekte — 1
 - 1.2.1 Preismodelle — 1
 - 1.2.2 Rechnungsstellung — 3
- 1.3 Vertrag — 3
 - 1.3.1 Leistungsbeschreibung — 3
 - 1.3.2 Lizenzen — 4
 - 1.3.3 Support und Wartung — 5
- 1.4 Vermeidung von Klagen — 5

2 Projekte — 6

- 2.1 Lebenszyklus von Software — 6
 - 2.1.1 Zeitlicher Verlauf — 6
 - 2.1.2 Stakeholder — 6
- 2.2 Vorgehensmodelle — 7
 - 2.2.1 Entstehung — 7
 - 2.2.2 RUP — 7
 - 2.2.3 XP — 8
 - 2.2.4 Agiles Manifest — 8
 - 2.2.5 Scrum — 9
 - 2.2.6 Kontinuierliche Prozesse — 11
 - 2.2.7 SAFe — 12
- 2.2.8 Jedes Projekt besitzt ein eigenes Vorgehensmodell — 12
- 2.3 Das Projektteam — 13
 - 2.3.1 Organisationsformen — 13
 - 2.3.2 Zusammenstellung des Projektteams — 14
 - 2.3.3 Verteilte Teams — 15
- 2.4 Aufgaben der Projektleitung — 16
 - 2.4.1 Projektstart — 16
 - 2.4.2 Planung und Tracking — 16
 - 2.4.3 Berichterstattung — 19
 - 2.4.4 Risk Management — 19
 - 2.4.5 Förderung der Zusammenarbeit — 19
- 2.5 Artefakte im Projekt — 21
- 2.6 Das magische Dreieck — 22

3	Requirements Engineering — 23
3.1	Vorgehensweise — 23
3.1.1	Business Modeling und Prototypen — 23
3.1.2	Anforderungskatalog — 23
3.1.3	Iterative Verfeinerung der Spezifikationen — 25
3.2	Arten von Anforderungen — 25
3.2.1	Funktionale Anforderungen — 25
3.2.2	Nicht-funktionale Anforderungen — 25
3.3	Schnittstelle zum Benutzer — 26
3.3.1	Grafische Oberfläche — 26
3.3.2	Usability — 27
3.3.3	Alternativen zur grafischen Oberfläche — 28
3.4	UML — 29
3.4.1	Grundlagen — 29
3.4.2	UML in der Praxis — 31
3.4.3	Alternativen zu UML — 31
4	Aufwandschätzung — 32
4.1	Vorgehensweise beim Schätzen — 32
4.2	Schätzeinheit — 33
4.3	Bestandteile von Aufwandschätzungen — 35
5	Hardware — 36
5.1	Grundlagen — 36
5.2	Prozessoren — 37
5.2.1	Aufbau von Prozessoren — 37
5.2.2	CPU, MCU und GPU — 39
5.3	Hardware-Schnittstellen — 40
5.3.1	Bus — 40
5.3.2	Serielle Schnittstellen — 42
5.3.3	Anbindung von Sensoren und Aktuatoren — 43
5.4	RAM — 43
5.5	Disks — 44
5.5.1	Disk-Typen — 44
5.5.2	Bootling und Partitionierung — 45
5.5.3	Dateisysteme — 46
5.6	Peripherie — 47
5.7	Geschichte der Computer — 48
5.8	Virtual Machines — 49
6	Betriebssysteme — 51
6.1	Konzepte — 51

6.2	Unix/Linux —	51
6.2.1	Unix —	51
6.2.2	Linux —	51
6.2.3	Grundlagen —	52
6.3	Windows —	55
6.4	Apple macOS —	56
6.5	IBM-Betriebssysteme —	56
6.6	Embedded OS —	57
6.7	Mobile OS —	58
6.7.1	Android —	58
6.7.2	iOS —	59
6.7.3	Andere mobile Systeme —	59
6.8	Andere OS —	60
7	Plattformen —	61
7.1	Der abstrakte Begriff der Plattform —	61
7.2	Non-OS-Systeme —	61
7.3	Native Programme —	61
7.4	Containerisierung —	63
7.5	Laufzeitumgebungen —	64
7.5.1	Die Java-Plattform —	64
7.5.2	Microsoft .NET —	69
7.5.3	Weitere Laufzeitumgebungen —	70
7.6	Webbrowser —	70
7.6.1	Webanwendungen —	70
7.6.2	Produkte —	71
7.6.3	Ein „Hello World“-Beispiel —	72
7.6.4	HTML5 —	74
7.6.5	CSS3 (Cascading Style Sheets) —	75
7.6.6	Grafik und Animation im Browser —	76
7.6.7	Moderne Frameworks/Libraries für Webapplikationen —	77
7.7	Mobile Plattformen —	79
7.8	Spielplattformen —	81
7.8.1	Konsolen —	81
7.8.2	Gaming —	81
8	Softwarearchitektur —	82
8.1	Grundlagen —	82
8.1.1	Begriffsbestimmung —	82
8.1.2	Entstehung einer Softwarearchitektur —	83
8.1.3	Der Softwarearchitekt —	84

8.1.4	Dokumentation von Softwarearchitekturen und technischen Konzepten —	85
8.2	Statische Sicht —	85
8.2.1	Module und Schichten —	85
8.2.2	Verschiedene Arten von Abhängigkeiten —	86
8.2.3	Minimierung von Abhängigkeiten —	87
8.2.4	Fassaden —	88
8.3	Laufzeitsicht —	88
8.4	Best Practice —	90
9	Geschichte der Trends in der Softwareentwicklung —	91
9.1	Vom Host zum Monolithen —	91
9.2	Client/Server —	91
9.2.1	Komponenten mit Thin Clients —	91
9.2.2	Rich Clients —	92
9.2.3	Applikationsserver —	92
9.3	EAI (Enterprise Application Integration) —	93
9.3.1	Webservices (SOAP) —	93
9.3.2	Middleware —	93
9.3.3	Lokale Services —	94
9.4	DSL (Domain Specific Languages) —	95
9.4.1	MDA und MDSD —	95
9.4.2	Software Factory —	96
9.4.3	Tabellengesteuerte Programmierung —	96
9.5	IoT und Cloud —	97
9.5.1	RESTful Webservices —	97
9.5.2	Cloud Computing —	98
9.5.3	IoT und M2M —	100
9.6	Aktuelle Trends —	104
9.6.1	Digitalisierung —	104
9.6.2	Dezentrale Netze —	104
9.6.3	Künstliche Intelligenz, neuronale Netzwerke und Robotik —	105
9.6.4	Neue Mensch-Maschine-Schnittstellen —	107
10	Standards —	108
10.1	Organisationen für Standardisierungen —	108
10.2	Text —	108
10.2.1	Schriftarten —	108
10.2.2	Steuerzeichen —	109
10.3	Datentypen —	109
10.3.1	Zahlen —	109
10.3.2	Datum und Zeit —	110

10.4	Inhalte — 110
10.4.1	Dateitypen — 110
10.4.2	Streaming Audio und Video — 112
10.5	Mehrsprachigkeit — 112
10.5.1	Zeichensätze — 112
10.5.2	Internationalisierung und Lokalisierung — 113
10.6	XML und JSON — 114
10.6.1	Einführung in XML — 114
10.6.2	XML-basierte Standards und Produkte — 116
10.6.3	JSON — 117
10.6.4	Hypertext — 118
11	Standardsoftware — 119
11.1	Standard oder maßgeschneidert? — 119
11.1.1	Produktkauf — 119
11.1.2	Entwicklung eines eigenen Produkts (Standard, Framework oder Library) — 119
11.2	Dateien und Backup — 120
11.2.1	Filesharing — 120
11.2.2	Backup — 120
11.3	Produkte für private Endbenutzer — 121
11.3.1	Musik — 121
11.3.2	Online — 121
11.3.3	Instant Messaging — 122
11.4	E-Commerce — 123
11.4.1	Onlineshops und Marktplätze — 123
11.4.2	Kryptowährungen — 124
11.5	Businessspezifische Produkte — 125
11.5.1	Enterprise (CRM, ERP, SCM) — 125
11.5.2	Industrielle Automatisierung — 125
11.5.3	Mathematik und Ingenieurwesen — 126
11.5.4	CMS und DMS — 127
11.5.5	BPM (Business Process Management) — 127
11.6	Typische Produkte für den Büroarbeitsplatz — 128
11.7	Suchmaschinen — 129
11.8	Umgang mit einer heterogenen Systemlandschaft — 129
12	Programmieren — 131
12.1	Grundlagen — 131
12.1.1	Sprachenvielfalt — 131
12.1.2	Theorie — 131
12.1.3	Programmierrichtlinien — 132

12.1.4	Objektorientierte Prinzipien —	133
12.1.5	Garbage Collection —	134
12.2	Sprachen —	134
12.2.1	Assembler (Maschinensprache) —	134
12.2.2	Skriptsprachen —	135
12.2.3	Prozedurale Sprachen —	137
12.2.4	Funktionale Sprachen —	137
12.2.5	Logische Sprachen —	138
12.2.6	Objektorientierte Sprachen —	138
12.2.7	Sprachen für Kinder —	139
12.2.8	Andere Sprachen —	140
12.3	Parallele Programmierung —	140
12.3.1	Nebenläufigkeit —	140
12.3.2	Parallelisierung —	141
12.3.3	MapReduce —	142
12.4	Refactoring vs. Redesign —	143
13	Tools für die Programmierung —	144
13.1	Compiler —	144
13.2	Build —	146
13.2.1	Tools —	146
13.2.2	Dependency-Management —	148
13.3	IDE (Integrated Development Environment) —	149
13.4	Zusätzliche Entwicklungs-Tools —	150
13.4.1	Profiling —	150
13.4.2	Debugging —	150
13.4.3	Weitere Tools —	152
14	Kommunikation und Netzwerke —	153
14.1	Grundlagen —	153
14.1.1	Modelle —	153
14.1.2	Standards —	153
14.1.3	Konzepte —	154
14.2	Kommunikations-Technologien —	156
14.2.1	Netzwerktypen —	156
14.2.2	Anschluss ans Internet —	157
14.2.3	Von RFID zu NFC —	157
14.2.4	Bluetooth und Alternativen —	159
14.2.5	Mobilfunknetz —	160
14.3	Netzwerkhardware —	160
14.4	IP —	161
14.4.1	Grundlagen —	161

14.4.2	Mapping von Domainnamen zu IP-Adressen —	163
14.4.3	Programmierung —	164
14.5	E-Mail —	165
14.6	HTTP —	165
14.6.1	Konzept —	165
14.6.2	Methoden —	166
15	Datenbanken —	168
15.1	Grundlagen —	168
15.1.1	Datenbanktypen —	168
15.1.2	Einbindung in die Applikation —	168
15.2	Relationale Datenbanken —	169
15.2.1	Konzepte relationaler Datenbanken —	169
15.2.2	Transaktionen —	170
15.2.3	Locking und Notifikationen —	171
15.2.4	SQL —	172
15.2.5	ERM-Diagramme —	174
15.3	Best Practice —	175
15.3.1	Mapping zwischen OO-Applikation und RDBMS —	175
15.3.2	Migration von Datenbanken —	175
15.3.3	Namenskonventionen —	176
15.3.4	Dokumentation von Datenmodellen —	176
15.3.5	Stammdaten —	177
15.3.6	Historisierung von Daten —	177
15.4	Produkte —	177
15.4.1	Klassische Datenbanken —	177
15.4.2	Relationale Tools —	179
15.4.3	Objektorientierte Datenbanken —	179
15.4.4	NoSQL —	179
15.5	Data-Warehouse —	180
15.5.1	Erstellung mittels ETL —	180
15.5.2	OLAP —	180
15.5.3	Alternativen zum Data-Warehouse —	181
15.6	Wissensbasierte Systeme —	181
15.7	Big Data Analytics —	182
15.7.1	Predictive Analytics —	182
15.7.2	Maschinelles Lernen —	183
16	Patterns —	185
16.1	Design Patterns —	185
16.2	Architekturpatterns —	186
16.2.1	Aufteilung in Module —	186

16.2.2	Laufzeit —	187
16.2.3	Kommunikation —	187
16.3	Algorithmen und Datenstrukturen —	189
16.3.1	Datenstrukturen —	189
16.3.2	Algorithmen —	190
16.4	Webapplikationen —	191
16.4.1	Server —	191
16.4.2	Client —	192
16.4.3	Push —	193
16.5	Games —	194
16.6	Programmierung allgemein —	195
16.6.1	Konzeptionelle Patterns —	195
16.6.2	Regular Expressions —	196
16.6.3	Sonstige Patterns —	197
16.7	Anti-Patterns —	198
16.7.1	Die Idee —	198
16.7.2	Anti-Patterns beim Programmieren —	198
16.7.3	Anti-Patterns beim objektorientierten Design —	199
16.7.4	Architektonische Anti-Patterns —	200
16.7.5	Anti-Patterns im Management —	200
17	Sicherheit —	202
17.1	Grundlagen —	202
17.2	Symmetrische Kryptosysteme —	202
17.2.1	Symmetrische Verschlüsselung —	202
17.2.2	Signaturen mit symmetrischen Schlüsseln —	203
17.2.3	Schlüsselaustausch —	203
17.3	Public-Key-Kryptosysteme —	204
17.3.1	Asymmetrische Verschlüsselung —	204
17.3.2	Signaturen mit Private-Key —	204
17.3.3	Zertifikate und PKI —	204
17.3.4	KSI —	207
17.4	Hacking —	207
17.4.1	Allgemeine Methoden —	207
17.4.2	Malware —	208
17.4.3	Informationen über das Opfer einholen —	209
17.4.4	Kategorien von Attacken —	210
17.4.5	Angriffe über das Web —	211
17.4.6	Wireless Hacking —	213
17.5	Maßnahmen für die Sicherheit —	214
17.5.1	IT-Infrastruktur allgemein —	214
17.5.2	Schutz des Servers —	215

17.5.3	Schutz des Clients —	215
17.5.4	Anonymisierung —	216
17.6	Authentifizierung und Autorisierung —	217
17.6.1	Authentifizierung des Clients —	217
17.6.2	Single Sign-On —	219
17.6.3	Zusammenspiel WAF und IAM —	220
17.6.4	Passwörter —	222
17.7	Applikatorische Sicherheit —	223
17.7.1	Validierungen —	223
17.7.2	Transaktionssignierung —	224
17.8	Technologien und Methoden —	225
17.8.1	Technologien und Standards —	225
17.8.2	SSL/TLS —	225
17.8.3	Penetration-Tests —	226
17.9	Cracking Software —	227
17.9.1	Reverse Engineering —	227
17.9.2	Technischer Schutz —	228
17.10	Sicherheit bei der Entwicklung —	228
18	Konfigurations- und Releasemanagement —	229
18.1	Releases —	229
18.1.1	Motivation —	229
18.1.2	Versionierung —	229
18.1.3	Erstellung eines Releases —	230
18.1.4	Schnittstellenversionierung —	231
18.2	Versionsverwaltung (Repository) —	231
18.2.1	Sourcecode —	231
18.2.2	Binäre Artefakte —	232
18.2.3	Produkte —	233
18.2.4	Git —	234
18.3	Umgebungen —	235
18.3.1	Von der Entwicklung bis zur Produktion —	235
18.3.2	Fallbeispiel Konfigurations- und Releasemanagement —	236
18.4	DevOps —	236
18.5	Klassische Installationen —	238
18.5.1	Installation auf Unix, Linux, macOS und Windows —	238
18.5.2	Installation von Java-Applikationen —	238
18.5.3	Installation auf eingebetteten Systemen —	239
19	Qualitätssicherung —	240
19.1	Testarten —	240
19.2	Testen —	240

19.2.1	Testdrehbuch —	240
19.2.2	Unit-Tests —	241
19.2.3	Automatisierte Systemtests —	242
19.2.4	Testwerkzeuge —	242
19.3	Issue-Tracking —	243
19.3.1	Umgang mit Issues —	243
19.3.2	Change Control Board (CCB) —	245
19.3.3	Produkte —	245
19.4	Continuous Integration —	245
19.5	Qualitätsbeurteilung —	246
19.5.1	Softwaremetrik —	246
19.5.2	Code-Reviews —	247
20	Dokumentation —	248
20.1	Vorgehensweise —	248
20.2	Tools —	249
20.3	Schulung —	249
21	Aspekte im Betrieb —	250
21.1	Monitoring und Reporting —	250
21.2	Organisation des Supports —	250
21.3	Update im laufenden Betrieb —	251
22	Der Beruf des Softwareentwicklers —	252
22.1	Der Arbeitsplatz —	252
22.2	Ethik —	252
22.3	Ausbildung und Weiterbildung —	253
22.4	Networking und Bewerbungen —	254
22.5	Zehn zeitlose Leitlinien —	254
Literatur und Links —		257
Register —		259