

Albin Meyer

# Software- entwicklung



Ein Kompass für die Praxis

**DE GRUYTER**  
OLDENBOURG

# Inhalt

## Vorwort — V

## 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 — 4
- 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 Jedes Projekt besitzt ein eigenes Vorgehensmodell — 12
- 2.3 Das Projektteam — 13
  - 2.3.1 Organisationsformen — 13
  - 2.3.2 Zusammenstellung des Projektteams — 13
  - 2.3.3 Verteilte Teams — 14
- 2.4 Aufgaben der Projektleitung — 15
  - 2.4.1 Projektstart — 15
  - 2.4.2 Planung und Tracking — 16
  - 2.4.3 Berichterstattung — 19
  - 2.4.4 Risk Management — 20
  - 2.4.5 Förderung der Zusammenarbeit — 20
- 2.5 Artefakte im Projekt — 21
- 2.6 Das magische Dreieck — 22

<b>3</b>	<b>Requirements Engineering — 23</b>
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
<b>4</b>	<b>Aufwandschätzung — 33</b>
4.1	Vorgehensweise beim Schätzen — 33
4.2	Schätzeinheit — 34
4.3	Bestandteile von Aufwandschätzungen — 36
<b>5</b>	<b>Hardware — 37</b>
5.1	Grundlagen — 37
5.2	Prozessoren — 38
5.2.1	Aufbau von Prozessoren — 38
5.2.2	CPU, MCU und GPU — 40
5.3	Hardware-Schnittstellen — 41
5.3.1	Bus — 41
5.3.2	Serielle Schnittstellen — 43
5.3.3	Anbindung von Sensoren und Aktuatoren — 44
5.4	RAM — 44
5.5	Disks — 45
5.5.1	Disk-Typen — 45
5.5.2	Booting und Partitionierung — 46
5.5.3	Dateisysteme — 47
5.6	Peripherie — 47
5.7	Geschichte der Computer — 49
5.8	Virtual Machines — 50
<b>6</b>	<b>Betriebssysteme — 52</b>
6.1	Konzepte — 52
6.2	Unix/Linux — 52

- 6.2.1 Unix — 52
- 6.2.2 Linux — 52
- 6.2.3 Grundlagen — 53
- 6.3 Windows — 55
- 6.4 Apple macOS — 57
- 6.5 IBM-Betriebssysteme — 57
- 6.6 Embedded OS — 58
- 6.7 Mobile OS — 59
  - 6.7.1 Android — 59
  - 6.7.2 iOS — 59
  - 6.7.3 Andere mobile Systeme — 60
- 6.8 Andere OS — 60
  
- 7 Plattformen — 62**
  - 7.1 Der abstrakte Begriff der Plattform — 62
  - 7.2 Non-OS-Systeme — 62
  - 7.3 Native Programme — 62
  - 7.4 Container — 64
  - 7.5 Die Java-Plattform — 64
    - 7.5.1 Grundlagen — 64
    - 7.5.2 Java Virtual Machine (JVM) — 65
    - 7.5.3 Java SE — 66
    - 7.5.4 Java EE — 68
    - 7.5.5 Java ME — 69
    - 7.5.6 Librarys und Frameworks — 69
    - 7.5.7 Webapplikationen mit Java-Frameworks — 71
    - 7.5.8 Webapplikationen mit REST — 72
  - 7.6 Microsoft .NET — 73
  - 7.7 Webbrowser — 74
    - 7.7.1 Webanwendungen — 74
    - 7.7.2 Produkte — 75
    - 7.7.3 Ein „Hello World“-Beispiel — 76
    - 7.7.4 HTML5 — 78
    - 7.7.5 CSS3 (Cascading Style Sheets) — 79
    - 7.7.6 Grafik und Animation im Browser — 80
    - 7.7.6 Moderne Frameworks/Librarys für Webapplikationen — 81
  - 7.8 Mobile Plattformen — 83
  - 7.9 Spielplattformen — 84
    - 7.9.1 Konsolen — 84
    - 7.9.2 Gaming — 84

<b>8</b>	<b>Softwarearchitektur — 86</b>
8.1	Grundlagen — 86
8.1.1	Begriffsbestimmung — 86
8.1.2	Entstehung einer Softwarearchitektur — 87
8.1.3	Der Softwarearchitekt — 88
8.1.4	Dokumentation von Softwarearchitekturen und technischen Konzepten — 89
8.2	Statische Sicht — 89
8.2.1	Module und Schichten — 89
8.2.2	Verschiedene Arten von Abhängigkeiten — 90
8.2.3	Minimierung von Abhängigkeiten — 91
8.2.4	Fassaden — 92
8.3	Laufzeitsicht — 93
8.4	Best Practice — 93
<b>9</b>	<b>Geschichte der Trends in der Softwareentwicklung — 94</b>
9.1	Vom Host zum Monolithen — 94
9.2	Client/Server — 94
9.2.1	Komponenten mit Thin Clients — 94
9.2.2	Rich Clients — 95
9.2.3	Applikationsserver — 95
9.3	EAI (Enterprise Application Integration) — 96
9.3.1	Webservices — 96
9.3.2	Middleware — 97
9.3.3	Lokale Services — 98
9.4	DSL (Domain Specific Languages) — 98
9.4.1	MDA und MDSD — 98
9.4.2	Software Factory — 99
9.4.3	Tabellengesteuerte Programmierung — 100
9.5	IoT und Cloud — 100
9.5.1	RESTful Webservices — 100
9.5.2	Cloud — 101
9.5.3	IoT und M2M — 102
9.6	Aktuelle Trends — 106
9.6.1	Digitalisierung — 106
9.6.2	Dezentrale Netze — 107
9.6.3	Künstliche Intelligenz, neuronale Netzwerke und Robotik — 107
9.6.4	Neue Mensch-Maschine-Schnittstellen — 109
<b>10</b>	<b>Standards — 110</b>
10.1	Organisationen für Standardisierungen — 110
10.2	Text — 110

- 10.2.1 **Schriftarten — 110**
- 10.2.2 **Steuerzeichen — 111**
- 10.3 **Datentypen — 111**
- 10.3.1 **Zahlen — 111**
- 10.3.2 **Datum und Zeit — 112**
- 10.4 **Inhalte — 112**
- 10.4.1 **Dateitypen — 112**
- 10.4.2 **Streaming Audio und Video — 114**
- 10.5 **Mehrsprachigkeit — 114**
- 10.5.1 **Zeichensätze — 114**
- 10.5.2 **Internationalisierung und Lokalisierung — 115**
- 10.6 **XML — 116**
- 10.6.1 **Einführung in XML — 116**
- 10.6.2 **XML-basierte Standards und Produkte — 118**
- 10.6.3 **Alternativen zu XML (ASN.1, CSV und JSON) — 119**
- 10.6.4 **Hypertext — 120**
  
- 11 Standardsoftware — 121**
- 11.1 **Standard oder maßgeschneidert? — 121**
- 11.1.1 **Produktkauf — 121**
- 11.1.2 **Entwicklung eines eigenen Produkts (Standard, Framework oder Library) — 121**
- 11.2 **Produkte für private Endbenutzer — 122**
- 11.2.1 **Filesharing — 122**
- 11.2.2 **Backup — 122**
- 11.2.3 **Musik — 123**
- 11.2.4 **Online — 123**
- 11.2.5 **Instant Messaging — 124**
- 11.3 **E-Commerce — 125**
- 11.3.1 **Onlineshops und Marktplätze — 125**
- 11.3.2 **Digitale Währungen — 125**
- 11.4 **Businessspezifische Produkte — 126**
- 11.4.1 **Enterprise (CRM, ERP, SCM) — 126**
- 11.4.2 **Industrielle Automatisierung — 126**
- 11.4.3 **Mathematik und Ingenieurwesen — 127**
- 11.4.4 **CMS und DMS — 128**
- 11.4.5 **BPM (Business Process Management) — 128**
- 11.5 **Typische Produkte für den Büroarbeitsplatz — 129**
- 11.6 **Suchmaschinen — 130**
- 11.7 **Umgang mit einer heterogenen Systemlandschaft — 131**

<b>12</b>	<b>Programmieren — 133</b>
12.1	Grundlagen — 133
12.1.1	Sprachenvielfalt — 133
12.1.2	Theorie — 133
12.1.3	Programmierrichtlinien — 134
12.1.4	Objektorientierte Prinzipien — 135
12.1.5	Garbage Collection — 136
12.2	Sprachen — 136
12.2.1	Assembler (Maschinensprache) — 136
12.2.2	Skriptsprachen — 137
12.2.3	Prozedurale Sprachen — 139
12.2.4	Funktionale Sprachen — 139
12.2.5	Logische Sprachen — 140
12.2.6	Objektorientierte Sprachen — 140
12.2.7	Sprachen für Kinder — 142
12.2.8	Andere Sprachen — 142
12.3	Parallele Programmierung — 143
12.3.1	Nebenläufigkeit — 143
12.3.2	Parallelisierung — 144
12.3.3	MapReduce — 145
12.4	Refactoring vs. Redesign — 145
<b>13</b>	<b>Tools für die Programmierung — 147</b>
13.1	Compiler — 147
13.2	Build — 149
13.2.1	Tools — 149
13.2.2	Dependency-Management — 151
13.3	IDE (Integrated Development Environment) — 152
13.4	Zusätzliche Entwicklungs-Tools — 153
13.4.1	Profiling — 153
13.4.2	Debugging — 153
13.4.3	Weitere Tools — 155
<b>14</b>	<b>Kommunikation und Netzwerke — 156</b>
14.1	Grundlagen — 156
14.1.1	Modelle — 156
14.1.2	Standards — 156
14.1.3	Konzepte — 157
14.2	Kommunikations-Technologien — 158
14.2.1	Netzwerktypen — 158
14.2.2	Anschluss ans Internet — 159
14.2.3	Von RFID zu NFC — 160

- 14.2.4 Bluetooth und Alternativen — 161
- 14.2.5 Mobilfunknetz — 162
- 14.3 Netzwerkhardware — 163
- 14.4 IP — 164
  - 14.4.1 Grundlagen — 164
  - 14.4.2 Mapping von Domainnamen zu IP-Adressen — 165
  - 14.4.3 Programmierung — 166
- 14.5 E-Mail — 167
- 14.6 HTTP — 168
  - 14.6.1 Konzept — 168
  - 14.6.2 Methoden — 169
  
- 15 Datenbanken — 171**
  - 15.1 Grundlagen — 171
    - 15.1.1 Datenbanktypen — 171
    - 15.1.2 Einbindung in die Applikation — 171
  - 15.2 Relationale Datenbanken — 172
    - 15.2.1 Konzepte relationaler Datenbanken — 172
    - 15.2.2 Transaktionen — 173
    - 15.2.3 Locking und Notifikationen — 174
    - 15.2.4 SQL — 175
    - 15.2.5 ERM-Diagramme — 177
  - 15.3 Best Practice — 178
    - 15.3.1 Mapping zwischen OO-Applikation und RDBMS — 178
    - 15.3.2 Migration von Datenbanken — 179
    - 15.3.3 Namenskonventionen — 179
    - 15.3.4 Dokumentation von Datenmodellen — 180
    - 15.3.5 Stammdaten — 180
    - 15.3.6 Historisierung von Daten — 180
  - 15.4 Produkte — 181
    - 15.4.1 Klassische Datenbanken — 181
    - 15.4.2 Relationale Tools — 182
    - 15.4.3 Objektorientierte Datenbanken — 182
    - 15.4.4 NoSQL — 182
  - 15.5 Data-Warehouse — 183
    - 15.5.1 Erstellung mittels ETL — 183
    - 15.5.2 OLAP — 183
    - 15.5.3 Alternativen zum Data-Warehouse — 184
  - 15.6 Wissensbasierte Systeme — 184
  - 15.7 Big Data — 185
    - 15.7.1 Predictive Analytics — 185
    - 15.7.2 Maschinelles Lernen — 186

**16 Patterns — 188**

- 16.1 Design Patterns — 188
- 16.2 Architekturpatterns — 189
  - 16.2.1 Aufteilung in Module — 189
  - 16.2.2 Laufzeit — 190
  - 16.2.3 Kommunikation — 190
- 16.3 Algorithmen und Datenstrukturen — 192
  - 16.3.1 Datenstrukturen — 192
  - 16.3.2 Algorithmen — 193
- 16.4 Webapplikationen — 194
  - 16.4.1 Server — 194
  - 16.4.2 Client — 195
  - 16.4.3 Push — 196
- 16.5 Games — 197
- 16.6 Programmierung allgemein — 198
  - 16.6.1 Konzeptionelle Patterns — 198
  - 16.6.2 Regular Expressions — 199
  - 16.6.3 Sonstige Patterns — 200
- 16.7 Anti-Patterns — 201
  - 16.7.1 Die Idee — 201
  - 16.7.2 Anti-Patterns beim Programmieren — 201
  - 16.7.3 Anti-Patterns beim objektorientierten Design — 202
  - 16.7.4 Architektonische Anti-Patterns — 202
  - 16.7.5 Anti-Patterns im Management — 203

**17 Sicherheit — 205**

- 17.1 Grundlagen — 205
- 17.2 Symmetrische Kryptosysteme — 205
  - 17.2.1 Symmetrische Verschlüsselung — 205
  - 17.2.2 Signaturen mit symmetrischen Schlüsseln — 206
  - 17.2.3 Schlüsselaustausch — 206
- 17.3 Public-Key-Kryptosysteme — 207
  - 17.3.1 Asymmetrische Verschlüsselung — 207
  - 17.3.2 Signaturen mit Private-Key — 207
  - 17.3.3 Zertifikate und PKI — 207
  - 17.3.4 KSI — 210
- 17.4 Hacking — 210
  - 17.4.1 Allgemeine Methoden — 210
  - 17.4.2 Malware — 211
  - 17.4.3 Informationen über das Opfer einholen — 212
  - 17.4.4 Kategorien von Attacken — 213
  - 17.4.5 Angriffe über das Web — 214

- 17.4.6 **Wireless Hacking — 216**
- 17.5 **Maßnahmen für die Sicherheit — 217**
- 17.5.1 **IT-Infrastruktur allgemein — 217**
- 17.5.2 **Schutz des Servers — 218**
- 17.5.3 **Schutz des Clients — 218**
- 17.5.4 **Anonymisierung — 219**
- 17.6 **Authentifizierung und Autorisierung — 220**
- 17.6.1 **Authentifizierung des Clients — 220**
- 17.6.2 **Single Sign-On — 222**
- 17.6.3 **Zusammenspiel WAF und IAM — 223**
- 17.6.4 **Passwörter — 225**
- 17.7 **Applikatorische Sicherheit — 225**
- 17.7.1 **Validierungen — 225**
- 17.7.2 **Transaktionssignierung — 226**
- 17.8 **Technologien und Methoden — 227**
- 17.8.1 **Technologien und Standards — 227**
- 17.8.2 **SSL/TLS — 228**
- 17.8.3 **Penetration-Tests — 228**
- 17.9 **Cracking Software — 229**
- 17.9.1 **Reverse Engineering — 229**
- 17.9.2 **Technischer Schutz — 230**
- 17.10 **Sicherheit bei der Entwicklung — 230**
  
- 18 Konfigurations- und Releasemanagement — 232**
- 18.1 **Releases — 232**
- 18.1.1 **Motivation — 232**
- 18.1.2 **Versionierung — 232**
- 18.1.3 **Erstellung eines Releases — 233**
- 18.1.4 **Schnittstellenversionierung — 234**
- 18.2 **Versionsverwaltung (Repository) — 234**
- 18.2.1 **Sourcecode — 234**
- 18.2.2 **Binäre Artefakte — 236**
- 18.2.3 **Produkte — 236**
- 18.2.4 **Git — 237**
- 18.3 **Umgebungen — 238**
- 18.3.1 **Von der Entwicklung bis zur Produktion — 238**
- 18.3.2 **Fallbeispiel Konfigurations- und Releasemanagement — 238**
- 18.3.3 **DevOps — 240**
- 18.4 **Deployment — 240**
- 18.4.1 **Rollout — 240**
- 18.4.2 **Installation auf Unix, Linux, macOS und Windows — 240**

- 18.4.3 Installation von Java-Applikationen — **241**
- 18.4.4 Installation auf eingebetteten Systemen — **241**

**19 Qualitätssicherung — 242**

- 19.1 Testarten — **242**
- 19.2 Testen — **243**
  - 19.2.1 Testdrehbuch — **243**
  - 19.2.2 Unit-Tests — **243**
  - 19.2.3 Automatisierte Systemtests — **244**
  - 19.2.4 Testwerkzeuge — **244**
- 19.3 Issue-Tracking — **245**
  - 19.3.1 Umgang mit Issues — **245**
  - 19.3.2 Change Control Board (CCB) — **247**
  - 19.3.3 Produkte — **247**
- 19.4 Continuous Integration — **247**
- 19.5 Qualitätsbeurteilung — **248**
  - 19.5.1 Softwaremetrik — **248**
  - 19.5.2 Code-Reviews — **248**

**20 Dokumentation — 250**

- 20.1 Vorgehensweise — **250**
- 20.2 Tools — **251**
- 20.3 Schulung — **251**

**21 Aspekte im Betrieb — 252**

- 21.1 Monitoring und Reporting — **252**
- 21.2 Organisation des Supports — **252**
- 21.3 Update im laufenden Betrieb — **253**

**22 Der Beruf des Softwareentwicklers — 254**

- 22.1 Der Arbeitsplatz — **254**
- 22.2 Ethik — **254**
- 22.3 Ausbildung und Weiterbildung — **255**
- 22.4 Networking und Bewerbungen — **256**
- 22.5 Zehn zeitlose Leitlinien — **256**

**Literatur und Links — 259**

**Register — 263**