



Application Engineering

Grundlagen für die objektorientierte Softwareentwicklung
mit zahlreichen Beispielen, Aufgaben und Lösungen

Hansruedi Tremp und Markus Ruggiero

Inhaltsverzeichnis

	Vorwort zur 1. Auflage	7
Teil A	Grundlagen	9
1	Software-Engineering-Prozess gestalten	10
1.1	Welche Bedeutung haben Vorgehensmodelle?	10
1.2	Welche Dimensionen definiert ein Vorgehensmodell?	11
1.3	Wie sind phasenorientierte Vorgehensmodelle beschaffen?	12
1.4	Was zeichnet agile Vorgehensmodelle aus?	16
1.5	Wie können Prozesse bewertet und verbessert werden?	19
2	Model Driven Architecture (MDA) verstehen	23
2.1	Was beinhaltet die Modellierung?	23
2.2	Welches sind die wichtigen MDA-Normen?	24
2.3	Welche Gesichtspunkte spielen eine Rolle?	26
2.4	Wie kann ein Modell umgesetzt werden?	28
3	Überblick über die Unified Modeling Language (UML)	32
3.1	Wie ist UML entstanden?	32
3.2	Welche wichtigen Diagrammtypen enthält UML?	34
3.3	Welches sind die Basisnotationen von UML?	35
3.4	Welche Beziehungen kennt UML?	37
3.5	Wie kann ein Modell präzisiert werden?	39
3.6	Wie kann UML erweitert werden?	40
3.7	Welche Kategorien von Modellierungswerkzeugen für UML gibt es?	42
4	Softwarequalität planen	45
4.1	Wie wird die Qualitätssicherung im Projekt umgesetzt?	45
4.2	Welche Merkmale weist eine gute Software auf?	47
4.3	Wie läuft der QS-Prozess im Projekt ab?	50
4.4	Welches sind wesentliche Inhalte eines Testkonzepts?	51
Teil B	Analyse und Planung	55
5	Anforderungen erheben und verwalten	56
5.1	Welchen Einfluss hat das Requirements Engineering?	56
5.2	Welche Bedeutung spielt der Systemkontext?	57
5.3	Welche Arten von Anforderungen werden unterschieden?	58
5.4	Welche Prozessschritte beinhaltet das Requirements Engineering?	60
5.5	Welche Eigenschaften sollte der Requirements Engineer haben?	68
6	Anwendungsfallmodell erstellen	70
6.1	Wozu dient ein Anwendungsfallmodell?	70
6.2	Welche Schritte sind zur Erstellung des Anwendungsfallmodells notwendig?	72
6.3	Wie werden Anwendungsfalldiagramme erstellt?	75
6.4	Was enthält die Anwendungsfallbeschreibung?	79
6.5	Vom Anwendungsfall zum Testfall?	80
7	Statisches Analysemodell erstellen	83
7.1	Wie lassen sich Klassen identifizieren?	83
7.2	Wie werden Klassen in UML beschrieben?	84
7.3	Wie werden Beziehungen im Klassendiagramm modelliert?	87
7.4	Wie helfen Analysemuster, Lösungen zu finden?	91
7.5	Anwendung am Beispiel Hotelreservation	93
8	Dynamisches Analysemodell erstellen	97
8.1	Funktion und Darstellungsformen	97
8.2	Aktivitätsdiagramm	97
8.3	Zustandsdiagramm	102
8.4	Sequenzdiagramm	103

Teil C	Architektur und Entwurf	109
9	Softwarearchitektur wählen	110
9.1	Was beeinflusst die IT-Architektur?	110
9.2	Wie können Mehrschichtenarchitekturen gestaltet sein?	111
9.3	Wie wird die Integration gestaltet?	117
10	Graphical User Interface (GUI) entwerfen	124
10.1	Wie ist der GUI-Entwurf zu gestalten?	124
10.2	Welche Vorgaben sind zu berücksichtigen?	126
10.3	Wie soll eine Software audiovisuell gestaltet sein?	128
10.4	Wie soll die Interaktion gestaltet werden?	134
10.5	Wie ist das Prototyping zu gestalten?	136
11	Komponentenentwurf festlegen	139
11.1	Was ist eine Komponente?	139
11.2	Wie werden Subsysteme gebildet?	140
11.3	Welches sind wichtige Entwurfsprinzipien?	140
11.4	Was leisten Architektur- bzw. Entwurfsmuster?	141
11.5	Schichtenmuster	142
11.6	Model View Control (MVC)	142
11.7	Wrapper-Muster	143
11.8	Singleton	143
11.9	Adapter	144
11.10	Proxy	144
11.11	Facade	145
11.12	Kontrollmuster	146
11.13	Wie sind die Komponenten festzulegen?	146
12	Persistenz gestalten	150
12.1	Weshalb ist Persistenz notwendig?	150
12.2	Welche Vorteile bringt der Einsatz eines DBMS?	150
12.3	Wie kann der Zugriff auf den Persistenz-Layer gestaltet werden?	151
12.4	Welche Ebenen beinhaltet die ANSI-SPARC-Architektur?	152
12.5	Welches sind die Merkmale des relationalen Datenmodells?	153
12.6	Welche Schritte enthält das ORM?	154
12.7	Wie wird ein relationales Datenmodell mit UML beschrieben?	157
Teil D	Entwicklung	161
13	Entwicklungsframeworks	162
13.1	Von der Quellcode-Bibliothek zum Framework	162
13.2	Was bedeutet der Einsatz eines Frameworks?	169
13.3	Welche Frameworks gibt es?	170
13.4	Was ist ein CASE-Tool?	172
13.5	Welche Frameworks und Werkzeuge werden eingesetzt?	172
13.6	Welche Architektur wird realisiert?	176
14	Datenschicht implementieren	179
14.1	Die relationale Welt	179
14.2	Die Objektwelt	180
14.3	Wie kommen die Objekte zu ihren Daten?	180
14.4	Konsequenzen für die Fallstudie	181
14.5	Projekt für die Datenhaltungs- und Geschäftslogik	182
14.6	EO Model erstellen	182
14.7	Datenbank generieren	189
14.8	Java-Code generieren	191
14.9	Notwendige Anpassung für die Vererbung	193
14.10	Ein Blick hinter die Kulissen des EOF	193

15	Business Layer implementieren	198
15.1	Sind Datenbankklassen auch Geschäftsklassen?	198
15.2	Geschäftsklassen für Fallstudie implementieren	200
15.3	Reservationsquellen implementieren	200
15.4	Reservationsobjekte implementieren	202
15.5	Zimmer reservieren	202
15.6	Freie Zimmer suchen	203
15.7	Verschiedene Preise implementieren	205
15.8	Wie kommen Kunden zu ihrer Kundennummer?	206
16	Präsentationsschicht	208
16.1	Reservationssystem für Internetbenutzer implementieren	208
16.2	WebObjects und Model View Controller	209
16.3	Seitengerüst realisieren	210
16.4	Sitzungsverwaltung in einer WebObjects-Anwendung	210
16.5	HTML-Teil der Reservationsseite realisieren	211
16.6	Controller-Teil programmieren	212
16.7	View und Controller verbinden	214
16.8	Reservation validieren und speichern	216
16.9	Wie werden die Fehlermeldungen angezeigt?	219
16.10	Wie kommt die Reservation in die Datenbank?	221
17	Schnittstellen zu Umsystemen	225
17.1	Reservationskomponente einrichten	225
17.2	Code für Web-Service schreiben	227
17.3	Die WSDL-Datei	230
Teil E	Test und Transition	235
18	Artefakte statisch prüfen	236
18.1	Welchen Nutzen bringen statische Prüfverfahren?	236
18.2	Wie sieht ein Prüfplan aus?	237
18.3	Welche Arten von Prüfverfahren gibt es?	238
18.4	Wie sehen die Rollenverteilung und der Ablauf eines technischen Reviews aus?	239
18.5	Wie läuft ein Audit ab?	241
18.6	Was kann eine Werkzeugunterstützung bieten?	242
18.7	Welches sind die Erfolgsfaktoren für statische Prüfungen?	242
19	Lauffähige Komponenten testen	245
19.1	Wie sieht die Teststufenplanung im Detail aus?	245
19.2	Wie werden Testfälle ermittelt und beschrieben?	251
19.3	Wie werden Tests durchgeführt?	254
19.4	Wie sieht das Fehlermanagement aus?	255
20	Dokumentieren und Schulen	259
20.1	Warum und wie soll dokumentiert werden?	259
20.2	Welche Werkzeuge kommen zum Einsatz?	260
20.3	Was enthält die Projektdokumentation?	261
20.4	Was ist bei der Entwicklerdokumentation zu beachten?	262
20.5	Wie kann die Vermarktung unterstützt werden?	262
20.6	Worauf ist bei der Installationsanleitung zu achten?	263
20.7	Was enthält die Betriebsdokumentation?	263
20.8	Wie ist die Benutzerdokumentation aufgebaut?	264
20.9	Wie ist der Schulungsprozess zu organisieren?	265
21	System einführen	269
21.1	Wie und mit wem ist die Systemeinführung zu planen?	269
21.2	Wie sieht der Einsatz des Verteilungsdiagramms aus?	270
21.3	Wie läuft die Datenmigration ab?	272
21.4	Wie erfolgt der Rollout?	274
21.5	Welche Aspekte ergeben sich in der Wartungsphase?	274

Teil F	Anhang	277
	Fallstudie «Seminarhotel»	278
	Lösungen zu den Aufgaben	283
	Weiterführende Literatur zum Thema	300
	Nützliche Links zum Thema	301
	Stichwortverzeichnis	302