

# Inhalt

Vorwort .....	21
Einleitung .....	23

## **1 Einführung** 31

---

<b>1.1 Nachhaltigkeit: Entwicklung und Wartung mit einer Testinfrastruktur ...</b>	32
<b>1.2 Agilität: Praktiken des Agile Software Engineerings .....</b>	37
<b>1.3 Effizienz: Einhaltung von Designprinzipien .....</b>	38
<b>1.4 Kommunikation: Diagramme als Gesprächsgrundlage .....</b>	42

## **TEIL I Aufbau einer Testinfrastruktur**

## **2 Beispielanwendung für diesen Buchteil** 47

---

<b>2.1 Stammdatenmanagement .....</b>	48
2.1.1 Änderungsantrag .....	48
2.1.2 Definition eines Stammdatenmodells .....	49
<b>2.2 Benutzeroberfläche .....</b>	50
<b>2.3 Backend .....</b>	51

## **3 Codebasierte Verbesserung eines Tests** 55

---

<b>3.1 Einführung in den Testcode .....</b>	55
3.1.1 Erste Definition einer Testklasse .....	56
3.1.2 Erste Implementierung der Testklasse .....	57
3.1.3 ABAP-Unit-Protokoll .....	64
3.1.4 Diagramme .....	65
<b>3.2 Allgemeine Clean-Code-Prinzipien .....</b>	67
3.2.1 Verbesserungsprozess .....	67
3.2.2 Verbesserung der Definition der Testklasse .....	69

3.2.3	Verbesserung der Implementierung der Testklasse .....	70
3.2.4	Diagramme .....	74
<b>3.3</b>	<b>Testorientierte Clean-Code-Prinzipien .....</b>	<b>76</b>
3.3.1	Verbesserungsprozess .....	76
3.3.2	Verbesserung der Implementierung der Testklasse .....	77
3.3.3	Diagramme .....	81

---

## **4 Designbasierte Verbesserung des Tests** 85

<b>4.1</b>	<b>Basisklasse von Testklassen .....</b>	<b>85</b>
4.1.1	Verbesserungsprozess .....	86
4.1.2	Extraktion einer Basisklasse aus einer Testklasse .....	86
4.1.3	Diagramme .....	88
<b>4.2</b>	<b>Verwendung von Hilfsklassen durch Testklassen .....</b>	<b>91</b>
4.2.1	Verbesserungsprozess .....	91
4.2.2	Entwurfsmuster für Klassen .....	91
4.2.3	Definition und Verwendung der Testdatenklasse .....	92
4.2.4	Diagramme .....	95

---

## **5 Robuster Integrationstest** 99

<b>5.1</b>	<b>Verbesserungsprozess .....</b>	<b>99</b>
<b>5.2</b>	<b>Unabhängigkeit von Testmethoden .....</b>	<b>100</b>
5.2.1	Testdatencontainer .....	101
5.2.2	Singleton-Entwurfsmuster .....	102
<b>5.3</b>	<b>Wiederholbarkeit von Testmethoden .....</b>	<b>105</b>
5.3.1	Automatischer Testaufbau .....	105
5.3.2	Hilfsmethoden der Testklasse .....	106
5.3.3	Diagramme .....	110

---

## **6 Minimierung von Abhängigkeiten** 113

<b>6.1</b>	<b>Vereinfachung einer Verwendung .....</b>	<b>114</b>
6.1.1	Verbesserungsprozess .....	115
6.1.2	Vereinfachung der produktiven API .....	116

6.1.3	Zugriff auf den Testdatencontainer .....	117
6.1.4	Delegierende Testklasse .....	117
6.1.5	Diagramme .....	119
<b>6.2</b>	<b>Abgrenzung einer Verwendung</b> .....	<b>121</b>
6.2.1	Verbesserungsprozess .....	122
6.2.2	Geschachtelte Interfaces .....	122
6.2.3	Diagramme .....	123
<b>6.3</b>	<b>Unabhängigkeit von einer Erzeugung</b> .....	<b>125</b>
6.3.1	Verbesserungsprozess .....	125
6.3.2	Definition einer Factory .....	126
6.3.3	Implementierung einer Factory .....	127
6.3.4	Verwendung einer Factory .....	128
6.3.5	Diagramme .....	128
<b>6.4</b>	<b>Unabhängigkeit von einer Erweiterung</b> .....	<b>131</b>
6.4.1	Verbesserungsprozess .....	131
6.4.2	Abstrakte Oberklasse aller Entitäten .....	131
6.4.3	Konkrete Unterklasse für Flugverbindungen .....	133
<b>7</b>	<b>Isolierter Komponententest</b> .....	<b>137</b>
<hr/>		
<b>7.1</b>	<b>Transformation des Integrationstests</b> .....	<b>138</b>
7.1.1	Verbesserungsprozess .....	139
7.1.2	Isolierung mit einem Governance-API-Double .....	139
7.1.3	Konfiguration des Governance-API-Doubles .....	144
7.1.4	Diagramme .....	147
<b>7.2</b>	<b>Skalierung mit einer Testsprache</b> .....	<b>150</b>
7.2.1	Verbesserungsprozess .....	150
7.2.2	Zustand einer Entität .....	151
7.2.3	Testmethode: Ungesicherte Änderungen .....	153
7.2.4	Testmethode: Änderungen zu einem neuen Entitätstyp .....	155
7.2.5	Testmethode: Enthaltene Änderungen .....	156
<b>7.3</b>	<b>Testgetriebene Entwicklung</b> .....	<b>157</b>
7.3.1	Testklasse für das Hervorheben von Löschungen .....	158
7.3.2	Diagramme .....	159
<b>7.4</b>	<b>Liskov-Substitution-Prinzip</b> .....	<b>161</b>
7.4.1	Beispielhafte Verletzung .....	161
7.4.2	Beispielhafte Erfüllung .....	163

## 8 Redesign mit Unit-Tests 165

---

<b>8.1</b>	<b>Objektorientierte API für Entitäten</b> .....	167
8.1.1	Entitätstypklasse .....	167
8.1.2	Selektionsklasse .....	167
8.1.3	Entitätsmengenklasse .....	169
8.1.4	Entity-API-Klasse .....	169
<b>8.2</b>	<b>Hervorheben von Änderungen als eigenständige Einheit</b> .....	170
8.2.1	Highlight-Changes-Interface .....	170
8.2.2	Highlight-Changes-Klasse .....	171
8.2.3	Berechnungsmethode für Entitätsknoten .....	172
8.2.4	Berechnungsmethode für Entitätsbäume .....	174
8.2.5	Verwendung der Highlight-Changes-Klasse .....	175
8.2.6	Objektorientiertes Programmiermodell .....	176
<b>8.3</b>	<b>Tests für die Highlight-Changes-Klasse</b> .....	177
8.3.1	Unit-Tests .....	178
8.3.2	Komponententests .....	180
<b>8.4</b>	<b>Zusammenfassung des ersten Teils</b> .....	182

## TEIL II Testorientiertes ABAP-Design

## 9 Design von Methoden 187

---

<b>9.1</b>	<b>Regeln zur Implementierung einer Methode</b> .....	188
9.1.1	Kommentierung .....	188
9.1.2	Formatierung .....	190
<b>9.2</b>	<b>Regeln für die Signatur einer Methode</b> .....	196
9.2.1	Einfaches Beispiel für die Verbesserung einer Methodensignatur .....	196
9.2.2	Tipps und Tricks für eine kompakte Methodensignatur .....	197
9.2.3	Anspruchsvolles Beispiel für die Verbesserung einer Methodensignatur .....	199

## 10 Design von Klassen

209

<b>10.1 Erzeugung eines Objekts durch seine Klasse</b> .....	210
10.1.1 Erzeugungsmethoden .....	210
10.1.2 Testbarkeit des Konstruktors .....	211
<b>10.2 Erzeugung eines Objekts durch eine Factory-Klasse</b> .....	213
10.2.1 Erzeugung ohne Factory .....	213
10.2.2 Erzeugung mit konkreter Factory .....	214
10.2.3 Erzeugung mit abstrakter Factory .....	215
<b>10.3 Abhängigkeitsarten zwischen Klassen</b> .....	217
10.3.1 Innere Abhängigkeit .....	218
10.3.2 Äußere Abhängigkeit .....	219
<b>10.4 Interfaces einer Klasse</b> .....	220
10.4.1 Berechtigungsspezifische Interfaces .....	220
10.4.2 Rollenspezifische Interfaces .....	221
10.4.3 Interface-Prinzipien .....	222
<b>10.5 Abstraktionsebenen innerhalb einer Klasse</b> .....	222
10.5.1 Entitätsmengenklasse mit Tabellen .....	224
10.5.2 Entitätsmengenklasse mit Objekten .....	226
10.5.3 Effizienz .....	227
<b>10.6 Katalog-Entwurfsmuster</b> .....	229
<b>10.7 Innerer Zusammenhalt</b> .....	231
10.7.1 Komponentengraph .....	232
10.7.2 Horizontale Zerlegung einer Klasse .....	233
10.7.3 Vertikale Zerlegung einer Klasse .....	234
10.7.4 Beispiel für die Zerlegung einer Klasse .....	235

## 11 Design von Paketen

237

<b>11.1 Paketkonzept</b> .....	237
11.1.1 Sichtbarkeit .....	239
11.1.2 Paketschnittstellen .....	240
11.1.3 Verwendungserklärungen .....	242
<b>11.2 Produktpakete</b> .....	244
<b>11.3 Testpakete</b> .....	246

## 12 Testfälle

249

<b>12.1 Testdesign</b> .....	250
12.1.1 Parameterorientierte Testverfahren .....	250
12.1.2 Zustandsorientiertes Testverfahren .....	253
<b>12.2 Testpyramide</b> .....	253
12.2.1 Weiterentwicklung von Bestandscode .....	253
12.2.2 Neuentwicklung .....	254
<b>12.3 Testabdeckung</b> .....	256
12.3.1 Zuständigkeiten der Testtypen .....	257
12.3.2 Schrittweise Vervollständigung der Testabdeckung .....	257

## 13 Testdoubles

259

<b>13.1 Vorteile von Testdoubles</b> .....	259
13.1.1 Abhängigkeiten eines Tests .....	260
13.1.2 Überprüfung des Verhaltens einer Methode .....	261
<b>13.2 Spezifikation von Testdoubles</b> .....	263
13.2.1 Testdoubletypen .....	264
13.2.2 Testdoubles verwenden .....	268
<b>13.3 Design von Testdoubles</b> .....	269
13.3.1 Interface- und Unterklassendoubles .....	270
13.3.2 Proxydoubles .....	271
13.3.3 Änderungen des Produktdesigns zugunsten der Testbarkeit .....	273
<b>13.4 Injektion von Testdoubles</b> .....	275
13.4.1 Von der CUT-Produktklasse angebotene Injektionsmechanismen ....	275
13.4.2 Von der CUT-Klasse ermöglichte Injektionsmechanismen .....	277
13.4.3 Von einem DOC-Singleton ermöglichte Injektionsmechanismen ....	280
13.4.4 Von einer statischen Factory ermöglichte Injektionsmechanismen .....	283
13.4.5 Von einer Singleton-Factory ermöglichte Injektionsmechanismen .....	284
13.4.6 Von einer abstrakten Factory ermöglichte Injektionsmechanismen .....	285

## 14 Globale Testdoubles 289

---

<b>14.1 Testdouble für eine DOM-Methode</b> .....	290
14.1.1 Doubleklasse mit vielen Erzeugungsmethoden .....	292
14.1.2 Doubleklasse mit einer Erzeugungsmethode .....	294
<b>14.2 Testdouble für zwei DOM-Methoden</b> .....	295
14.2.1 Doubleklasse für mehrere DOM-Methoden .....	295
14.2.2 Definition und Implementierung der Testklasse .....	298
<b>14.3 Kombination zweier Methodendoubles</b> .....	301
14.3.1 Starker Zusammenhalt einer Doubleklasse .....	301
14.3.2 Decorator-Doubleklasse .....	303
<b>14.4 Globalisierung von Testdoubles</b> .....	307
<b>14.5 Design globaler Testdoubles</b> .....	310
<b>14.6 Anpassung globaler Testdoubles</b> .....	313
14.6.1 Job-Interface .....	313
14.6.2 Globale Jobdoubleklasse .....	314
14.6.3 Testklasse für eine Jobverwenderklasse .....	316

## 15 Testklassen 319

---

<b>15.1 Lokale Testklassen</b> .....	321
<b>15.2 Entwurfsmuster</b> .....	324
15.2.1 Gegeben-Wenn-Dann-Entwurfsmuster .....	324
15.2.2 Testklassen-Entwurfsmuster .....	325
<b>15.3 Testklassenhierarchien</b> .....	327
<b>15.4 Globale Testklassen</b> .....	331
<b>15.5 Namenskonventionen für Testcode</b> .....	333

## 16 Testdaten 335

---

<b>16.1 Testdatencontainer</b> .....	335
16.1.1 Verwendung von Testdatencontainern in ABAP Unit .....	335
16.1.2 Vor- und Nachteile von Testdatencontainern .....	336
<b>16.2 Testdatenobjekte</b> .....	343

## 17 Testinfrastruktur 355

---

<b>17.1 Anwendungsszenarien für eine globale Testinfrastruktur</b> .....	357
17.1.1 Komponententests .....	357
17.1.2 Integrationstests .....	359
<b>17.2 Entwicklungsprozesse mithilfe einer Testinfrastruktur</b> .....	362
17.2.1 Entwicklung einer API für Legacy-Komponenten .....	362
17.2.2 Teamübergreifende Testinfrastruktur für ein Produkt .....	364
<b>17.3 Zusammenfassung des zweiten Teils</b> .....	365

## TEIL III Agile Neuentwicklung

## 18 Planung und Vorarbeit 369

---

<b>18.1 Spezifikation der Beispielanwendung</b> .....	370
<b>18.2 Architektur und Design der Beispielanwendung</b> .....	372
18.2.1 Paketstruktur .....	372
18.2.2 Datenmodell .....	374
18.2.3 Objektmodell .....	375
<b>18.3 Teststrategie für die Beispielanwendung</b> .....	377
18.3.1 Akzeptanztests .....	377
18.3.2 Prozesskomponententests .....	378
18.3.3 Modellkomponententests .....	379
18.3.4 Zugriffsintegrationstests .....	379
<b>18.4 Vorbereitung der testgetriebenen Entwicklung</b> .....	380
18.4.1 Skelett aller Komponenten .....	380
18.4.2 Skelett der Modellkomponente .....	383
18.4.3 Externe Factory eines Pakets .....	384
18.4.4 Interne Factory eines Pakets .....	386
18.4.5 API eines Pakets .....	387
18.4.6 Datenklasse in einem Paket .....	388
18.4.7 Basisklasse für die Komponententests eines Pakets .....	392



## 19 Testgetriebene Entwicklung 393

---

<b>19.1 Akzeptanztestgetriebene Entwicklung</b> .....	393
19.1.1 Basisklasse für die Akzeptanztests .....	394
19.1.2 Akzeptanztest (Gegeben- und Wenn-Phase) .....	395
<b>19.2 Komponententestgetriebene Entwicklung</b> .....	396
19.2.1 Komponententestklasse für den Erzeugungsprozess .....	397
19.2.2 Globales Double für den Vertragsmanager .....	398
19.2.3 Entwicklungstaktik .....	400
<b>19.3 Unit-Test-getriebene Entwicklung</b> .....	401
<b>19.4 Objektbasierte Überprüfung</b> .....	403
19.4.1 Akzeptanztest (Dann-Phase) .....	403
19.4.2 Objektbasierter Vergleich .....	404
19.4.3 Refactoring des Akzeptanztests .....	406
<b>19.5 Testen des Änderns und Sicherens eines Vertrags</b> .....	407
19.5.1 Komponententest .....	410
19.5.2 Integrationstest .....	412
<b>19.6 Zusammenfassung des dritten Teils</b> .....	413

## TEIL IV Agile Methoden

### 20 Scrum 417

---

<b>20.1 Artefakte</b> .....	417
20.1.1 Produkt-Backlog .....	418
20.1.2 Ready-Kriterien .....	418
20.1.3 Done-Kriterien .....	419
20.1.4 Sprint-Backlog .....	421
20.1.5 Sprint-Burndown-Chart .....	422
20.1.6 Release-Burndown-Chart .....	422
<b>20.2 Rollen</b> .....	423
20.2.1 Product Owner .....	423
20.2.2 Scrum Master .....	423
20.2.3 Team .....	424

<b>20.3 Meetings</b> .....	424
20.3.1 Backlog-Grooming .....	424
20.3.2 Sprintplanung – Teil 1 .....	425
20.3.3 Sprintplanung – Teil 2 .....	425
20.3.4 Daily Scrum .....	426
20.3.5 Sprint-Review .....	426
20.3.6 Sprint-Retrospektive .....	427
20.3.7 Zyklus .....	427
<b>20.4 Eigenschaften</b> .....	429

## **21 Agile Software Engineering** 433

---

<b>21.1 Refactoring</b> .....	434
21.1.1 Durchführung .....	434
21.1.2 Vor- und Nachteile .....	434
21.1.3 Abgrenzungen .....	435
<b>21.2 Testgetriebene Entwicklung</b> .....	437
21.2.1 Durchführung .....	437
21.2.2 Vor- und Nachteile .....	439
21.2.3 Alternativen .....	440
<b>21.3 Paarweise Programmierung</b> .....	441
21.3.1 Durchführung .....	442
21.3.2 Vor- und Nachteile .....	444
21.3.3 Alternativen .....	446
<b>21.4 Walking Skeleton</b> .....	447
21.4.1 Durchführung .....	447
21.4.2 Vor- und Nachteile .....	452
<b>21.5 Gemeinsame Codeverantwortung</b> .....	453
21.5.1 Durchführung .....	455
21.5.2 Vor- und Nachteile .....	456
21.5.3 Alternativen .....	457
<b>21.6 Kontinuierliche Integration</b> .....	459
21.6.1 Kontinuierliche Integration mit ABAP .....	459
21.6.2 Quasilokale Änderung .....	460

## **22 Lean-Entwicklungsmodell** 463

---

<b>22.1 Grundlagen</b> .....	463
<b>22.2 Umsetzung der Lean-Prinzipien mit Agile Software Engineering</b> .....	465
22.2.1 Refactoring .....	465
22.2.2 Testgetriebene Entwicklung .....	465
22.2.3 Paarweise Programmierung .....	466
22.2.4 Walking Skeleton .....	466
22.2.5 Gemeinsame Codeverantwortung .....	467
<b>22.3 Testinfrastruktur</b> .....	467
<b>22.4 Agile Coaching</b> .....	468

## **23 Entwicklung der Teams** 469

---

<b>23.1 Nachhaltige Ausbildung</b> .....	469
<b>23.2 Lücken in der Ausbildung</b> .....	473
23.2.1 Keine Unit-Tests .....	473
23.2.2 Lange Methoden .....	474
23.2.3 Ausbildungsplan .....	474
<b>23.3 Agile Coaching</b> .....	476
23.3.1 Durchführung .....	476
23.3.2 Vor- und Nachteile .....	481
<b>23.4 Netzwerk für Agile Coaching</b> .....	483

## **24 Entwicklung des Backlogs** 485

---

<b>24.1 Design Thinking</b> .....	486
24.1.1 Verstehen .....	488
24.1.2 Beobachten .....	489
24.1.3 Sichtweise definieren .....	490
24.1.4 Ideen finden .....	491
24.1.5 Prototypen entwickeln .....	491
24.1.6 Testen .....	491
<b>24.2 User Story Mapping</b> .....	492
24.2.1 Aufbau einer User Story Map .....	492
24.2.2 Überprüfung einer User Story Map .....	495

## **25 Entwicklung des Produkts** 497

---

<b>25.1 Nachhaltige Entwicklung</b> .....	498
25.1.1 Erhaltung der Produktivität .....	498
25.1.2 Verbesserung der Produktivität .....	499
<b>25.2 Entwicklungsstrategien für Legacy Code</b> .....	501
25.2.1 Charakterisierung .....	501
25.2.2 Transformation .....	501
<b>25.3 Entwicklungsstrategien für neuen Code</b> .....	502
<b>25.4 Zusammenfassung des vierten Teils</b> .....	504

## **TEIL V Testorientierte ABAP-Werkzeuge**

### **26 ABAP Unit** 507

---

<b>26.1 Klasse CL_ABAP_UNIT_ASSERT</b> .....	507
26.1.1 Methoden .....	508
26.1.2 Parameter .....	508
26.1.3 Bedingungen .....	509
<b>26.2 Ausführung von Tests</b> .....	512
26.2.1 Statische Grundlagen der Testausführung .....	512
26.2.2 Dynamische Grundlagen der Testausführung .....	513
26.2.3 Testbeziehungen .....	516
<b>26.3 Entwicklungsobjekte</b> .....	518

### **27 ABAP Development Tools** 521

---

<b>27.1 Einführung</b> .....	521
27.1.1 SAP-Hilfe .....	522
27.1.2 Projekt anlegen .....	523
27.1.3 Produktklasse anlegen .....	523
<b>27.2 Testgetriebene Entwicklung mit den ABAP Development Tools</b> .....	525
27.2.1 Berechtigungsprüfung als Test-Stub .....	526
27.2.2 Berechtigungsprüfung als Test-Spy .....	534
27.2.3 Berechtigungsprüfung als Mock-Objekt .....	540

## **28 ABAP-Werkzeuge zur Testisolierung** 545

---

<b>28.1</b>	<b>Beispielklasse</b> .....	545
<b>28.2</b>	<b>Open SQL Test Double Framework</b> .....	548
<b>28.3</b>	<b>Test Seams</b> .....	551
<b>28.4</b>	<b>ABAP Test Double Framework</b> .....	553
<b>28.5</b>	<b>Zusammenfassung des fünften Teils und des gesamten Buches</b> .....	557

## **Anhang** 559

---

<b>A</b>	<b>Namenskonventionen</b> .....	561
<b>B</b>	<b>Literaturverzeichnis</b> .....	563
<b>C</b>	<b>Der Autor</b> .....	564
	<b>Index</b> .....	565