



Roy
Osherove

The Art of
Unit Testing
Deutsche Ausgabe



Stichwortverzeichnis

A

AAA siehe Arrange-Act-Assert
Abbau 53
Abbaumethode 227
Abhängigkeit 68
Abhängigkeitsgrad 252
Abstract Factory Design Pattern, siehe Muster
Abstrakte Testinfrastruktur-Klassen, siehe Muster
Abstrakte Testtreiber-Klassen, siehe Muster
Abstrakte-Testinfrastruktur-Klassenmuster 165
Accidental Bugging siehe Versehentliches Bugging
Adapt Parameter Pattern, siehe Muster
Agent des Wandels 231
Akzeptanztests 293
API 70
Arbeiten mit Legacy Code 112, 196
Arrange-Act-Assert 143
 Modell 118
 Syntax 141
Assert
 mehrfaches 211
 Nachrichten 226
Assert-Hilfsklasse 181
Assert-Klasse 43
Astronaut 67
Attribut 43
Aufbau 53
Aufbaumethode 227
Aufnahmephase 121, 124
Ausgangszustand 53
Ausnahmen
 erwartete 56
Autofac 80, 286
AutomatedQA 207

B

Basisklasse 43, 74
Beck, Kent 23
Bedingte Kompilierung 86, 97
Beizer, Boris 239
Benachrichtigen 225
Blockierer 232, 233
Bottom-up siehe Guerilla-Implementierung
Bug 39
 Abhängigkeits-Bug 53
Build
 automatisches 155, 237
 automatisierter Prozess 156
 Break 156
 Skript 156

C

C++ 45, 116
Callback 135
Castle Windsor 80, 285
Champion 232
 extern 236
CI siehe Kontinuierliche Integration
Code
 Design 39
 Integration 156
 Konsistenz 237
Code Churn 239
Code Churn Perspective 240
Code Complete 240
Code-Abdeckung 43, 239
Code-Integrität 237
Code-Qualität 23, 39, 243
Collaborator 68
Common Service Locator 286
Communications of the ACM 247
Complete Guide to Software Testing 27
Conditional 96

Constraints siehe Parameter Constraints
Constructor Injection 76, 79, 81
Container 80
Continuous Integration siehe Kontinuierliche Integration
CppUnit 45
Crash Test Dummy 100
CRUD 172
CruiseControl.NET 157
CUT siehe System Under Test

D

Datenbank 287
Delegate 140
 benutzerdefinierter 269
Dependency Injection 284
Depender 259
Design 36, 267
 Interface-basiertes 270
 testhemmendes 69
 zum Zwecke der Testbarkeit 274
Design Pattern 73, 86
Dr. Boris Beizer on software testing
 an interview, part 1 239
DRY 165, 197
Duplizitäten 197
Dynamisches Fake-Objekt
 Definition 118

E

EasyMock 116
Eclipse 42
Effektives Arbeiten mit Legacy Code 30, 73, 91, 264
Eine Testklasse pro Feature, siehe Muster
Eine Testklasse pro Klasse, siehe Muster

- Eine-Verantwortlichkeit-Prinzip siehe Single Responsibility Principle
 Einfaches zuerst 254
 Eingeschränkte Testreihenfolge 205
 Encapsulation siehe Kapselung
 Entfernen doppelter Tests 191
 Entwicklungsprozess 36
 Erwartung 120, 149
 an Mocks 121
 an Stubs 121
 Even Simpler Database Unit Testing with TransactionScope 288
 Event 136
 abonnieren 137
 triggern 138
 EventsVerifier 140
 Extended Reflection siehe Profiler API
 External-Shared-State Corruption 205, 211
 Externe Abhängigkeit
 Definition 68
 Externe Ressource 69
 Extract and Override 88, 91, 93, 94, 269, 271
 Extreme Programming 237
- F**
- Fabrikklasse 76, 84, 86, 88
 Fabrikmethode 76, 86, 89, 178, 179
 lokale 88
 Fabrikmuster, siehe Muster
 Factory Class siehe Fabrikklasse
 Factory Pattern siehe Fabrikmuster
 Fake 68, 87, 118, 258
 Definition 107
 dynamischer siehe Dynamisches Fake-Objekt
 Objekt 111
 Rückgabe von Werten 124
 Feathers, Michael 30, 73, 91, 264
 Fehlermeldung 34
 Fehlersuche 39
 FICC 267
 FinalBuilder 157
 FIT 207
 FitNesse 207, 263, 294
- Funktionalitätstest 27
- G**
- Gallio 283
 Gamma, Erich 86
 Generics 179
 Guerilla-Implementierung 235
- H**
- Hanselman, Scott 80, 272
 Haskell 45
 Helm, Richard 86
 Hetzel, Bill 27
 Hilfsfunktion 36
 Hilfsklasse 181
 Hilfsmethode 34, 181
 HUnit 45
 Hunt, Andy 165
- I**
- IDE 42
 IDE Refactoring Tools 179
 Indirektionsschicht 70, 71, 87, 88, 195
 Insider 232
 Integration Test 23, 27, 67, 257
 Definition 27
 Merkmale 31
 Nachteile 29
 versteckter 160
 Integration Zone 161
 Integrierte Entwicklungsumgebung siehe IDE
 Intellectual Property siehe IP
 Interaction Testing 99, 100
 Definition 99
 Interagieren 99
 Interface 43, 70, 73, 74, 76, 92
 internal 96
 InternalsVisibleTo 96
 Inversion der Kontrolle siehe IoC
 Inversion of Control siehe IoC
 IoC 80, 81
 Container 80, 81, 284
 IP 275
 Isolation Framework 144, 149, 279
 Definition 116
 Ivonna 289
- J**
- Java 42, 45, 116, 262
- jMock 116
 JMockit 260
 Johnson, Ralph 86
 Jones, Capers 243
 JUnit 45, 205
- K**
- Kapselung 95
 Kapselungsproblem 95
 Klassenbibliothek 43
 Komplexität 275
 Konfigurationsklasse 181
 Konfigurationsproblem 160
 Konkrete Klassen 271
 Konstruktor 76, 80, 81, 88
 Kontinuierliche Integration 158, 237
- L**
- Legacy Code 29
 Definition 30
 Legacy-System 69
 Lesbarkeit 24
 List of .NET Dependency Injection Containers 272
 LogAn-Projekt 45
 Logik 42, 44
 Abdeckung 31
 Logik in Tests 191
 Logische Komplexität 252
 Logischer Code
 Definition 32
 Logisches Verhalten 31
- M**
- Magische Werte 151
 Martin, Bob 96
 Martin, Robert C. 268
 MbUnit 205, 213, 283
 McConnell, Steve 240
 MEF siehe Microsoft Managed Extensibility Framework
 Meszaros, Gerard 68, 104
 Methode
 Logik 47
 Methoden-Strings 147
 Microsoft CHES 293
 Microsoft Managed Extensibility Framework 287
 Microsoft Unit Testing Framework 282
 Microsoft Unity 80, 285
 Mock 68, 101, 106, 107, 111, 116, 219, 258

- dynamischer 118
- handgeschrieben 112
- nicht-strikter 122, 151
- ordered 125
- strikt 122
- Mock Framework 92, 101
- Mock-Objekt 99, 101
 - Definition 101
- mockpp 116
- MockRepository 119
- Moq 116, 148, 280
- MSBuild 157
- MSTest siehe Microsoft Unit Testing Framework
- Muster
 - Abstract Factory Design Pattern 86
 - Abstrakte Testinfrastruktur-Klasse 165
 - Abstrakte Testtreiber-Klasse 165, 172
 - Adapt Parameter Pattern 112
 - Eine Testklasse pro Feature 164
 - Eine Testklasse pro Klasse 163
 - Fabrikmuster 84
 - Template Testklasse 165, 171
 - Template-Testklassenmuster 165, 171
- Myre, Glenford 247
- N**
 - NAnt 157
 - NASA 67
 - NCover 245
 - NDepend 264
 - .NET
 - Profiler-API 146
 - .NET Generics siehe Generics
 - Newkirk, James 249
 - Ninject 80, 287
 - NMock 116, 146, 281
 - NMock2 146
 - NUnit 41, 45, 46, 205, 213, 282
 - Assert 50
 - Attribut 49, 53
 - Runner 46
 - SetUp 53
 - TearDown 53
 - Test 49
 - TestFixture 49
 - NUnit.Mocks 145, 281
 - NUnitAsp 290
 - NUnitForms 291
- O**
 - Objektorientierte Prinzipien 95
 - Objektorientierung 95
 - Open Closed Principle 268
 - Osherove.ThreadTester 293
- P**
 - Parameter
 - String Constraints 131
 - Parameter Constraints 130
 - Parameter Injection 76
 - Parameter Object Refactoring 80
 - Peer Reviews in Software
 - A Practical Guide 239
 - Pex 284
 - Pilotprojekt 235
 - Pragmatic Programmer, The 165
 - Principles of OOD 268
 - Priorität 252
 - Produktions-Bug 187
 - Produktionscode 36, 38
 - Produktionsklasse 33
 - Produktivität 243
 - Profiler API 259
 - Programming Productivity 243
 - Project White 291
 - Property 32, 60, 76, 81, 82, 88, 110
 - Injection 83
- Q**
 - QS 245, 246
 - QS-Ingenieur 232, 245
 - Quellcodekontrolle 161
- R**
 - Record-and-Replay 143
 - Modell 118
 - Syntax 120
 - Recording Phase siehe Aufnahme-phase
 - Refactoring 37, 38, 73, 257, 271
 - Definition 38, 73
 - Variationen 92
 - Refactoring von Tests 190
 - Refactoring-Technik 165
 - Regression
 - Bug 42
 - Definition 29
 - Reihenfolge siehe Mock
 - Release-Modus 86, 95
 - ReSharper 148, 265, 275
 - Rhino Mocks 116, 118, 147, 280
 - Richtigkeit 24
 - Rollback-Attribut 288
 - Rot-Grün-Konzept 53
- S**
 - Safe Green Test Area siehe Sichere grüne Zone
 - Schichttiefe 87
 - Schwieriges zuerst 255
 - Scrum 237
 - Seams 73, 86, 92, 271
 - Definition 73
 - Interface-basierte 76
 - Selenium 290
 - Setup-Methode 201, 204, 227
 - Shared-State Corruption 209
 - Sichere grüne Zone 161
 - Sicheres grünes Testgebiet siehe Sichere grüne Zone
 - Simian 265
 - Simplify Data Layer Unit Testing using Enterprise Services 288
 - Simulator 67
 - Single Responsibility Principle 95, 272
 - Singleton 272
 - Smalltalk 23
 - Software Assessments, Benchmarks, and Best Practices 243
 - Software QA Quarterly, The 239
 - Softwareentwicklung 24, 43
 - SOLID 96, 249
 - Artikelserie 96
 - Source Control Repository 156
 - Space Shuttle 67
 - Spring.NET 80, 286
 - SRP siehe Single Responsibility Principle
 - State-Based Testing
 - Definition 59
 - State-Verification siehe State-Based Testing
 - Statische Methoden 271
 - StoryTeller 294
 - StructureMap 80, 285
 - Stub 67, 68, 84, 87, 93, 101, 106, 107, 116, 126, 219, 258

Definition 68
 handgeschrieben 112
 Implementierung 76
 Kette 111
 Klasse 71
 Konfigurierbarkeit 79
 SUT siehe System Under Test
 System Under Test
 Definition 24

T

TDD siehe Testgetriebene Entwicklung
 Team System Web Test 289
 TeamCity 157
 Template-Testklassenmuster, siehe Muster
 Test
 automatisierter 155
 lesbarer 222
 parametrisierter 213
 Test Review 183, 194, 204
 Test Runner 44
 Test Spy 104
 Testable object-oriented design
 siehe TOOD
 Testbares objektorientiertes Design
 siehe TOOD
 Testbarkeit 73
 Test-Bug 187
 Testcode-Abdeckung siehe
 Code-Abdeckung
 Test-Driven Development in Microsoft .NET 249
 Test-Driven Development siehe
 Testgetriebene Entwicklung
 Test-First siehe Testgetriebene
 Entwicklung
 Testgetriebene Entwicklung 23,
 36, 237, 246, 249
 Test-Isolierung 204
 Testkategorie 58
 Testklassenhierarchie 179

Testklassenvererbung 165
 Testmissbrauch 151
 The Software QA Quarterly 239
 Thomas, Dave 165
 Thread 292
 TOOD 95
 Top-down 235, 236
 ToString() 216
 Traditionelle Codierung 36
 Traditionelle Entwicklung 36
 TransactionScope 288
 Trennung der Belange 85
 Typemock Isolator 116, 142,
 146, 258, 277, 280, 289
 Typemock Racer 266, 292

U

Über-Refaktorisierung 203
 Überspezifizierung 150, 218
 UI-Form 42
 UI-Testing 291
 Umbenennen von Tests 190
 Unit 39
 Unit Test 23, 31, 68
 ausführen 43
 auswerten 43
 automatisiert 30
 Benennung 38, 222
 Definition 24, 31
 Eigenschaften 26
 guter Unit Test 23, 24, 26
 ignorieren 58
 Lebenszyklus 53
 Lesbarkeit 38
 schreiben 43
 Wartbarkeit 38
 Unit Testing 237
 Unit Testing Framework 31, 36,
 41, 43, 44
 Unity siehe Microsoft Unity
 Unix 42
 User Interface 27

V

Variable
 Benennung 223
 vererben 43
 versehentliches Bugging 29
 Versteckte Testaufrufe 205,
 207
 Vertrauen 39
 Vertrauenswürdig 185
 Virtualisieren 93
 Vise 262
 Visual Build Pro 157
 Visual Studio 42, 47, 148, 158,
 289, 292
 Visual Studio Team Foundation
 Server 157
 Vlissides, John M. 86

W

Wartbarkeit 24, 195
 WatIN 290
 Watir 290
 Webform 42
 Widersprüchliche Tests 190
 Wiederholbarkeit 42
 Wiederverwendung 95
 Wiegers, Karl 239
 WinRunner 207
 Wrapper-Klasse 112

X

xUnit Framework 45, 283
 xUnit Test Patterns 68, 104,
 163, 164, 218

Z

Zeit 243
 Ziele 239
 Zustandsbasiertes Testen siehe
 State-Based Testing