

Inhaltsverzeichnis

Einführung	1
Wer sollte dieses Buch lesen	2
Aufbau dieses Buchs	3
Konventionen in diesem Buch	6
Systemvoraussetzungen	7
Downloads: Codebeispiele	8
Danksagungen	9
Errata und Support	9
Kostenlose E-Books von Microsoft Press	10

TEIL I **Eine agile Grundlage**

Kapitel 1	Einführung in Scrum	13
	Scrum und Wasserfall	15
	Rollen und Verantwortungsbereiche	17
	Product Owner	17
	Scrum Master	19
	Entwicklungsteam	19
	Schweine und Hühner	20
	Artefakte	20
	Das Scrum-Board	21
	Diagramme und Kennzahlen	35
	Backlogs	40
	Der Sprint	42
	Releaseplanung	43
	Sprintplanung	43
	Daily Scrum	45
	Sprintvorführung	47
	Sprint-Retrospektive	48
	Scrum-Kalender	50
	Probleme beim Einsatz von Scrum und Agile	51
	Nicht adaptiver Code	52
	Zusammenfassung	56

Kapitel 2	Abhängigkeiten und Schichten	57
	Die Definition von Abhängigkeit	58
	Ein simples Beispiel	59
	Modellieren von Abhängigkeiten in einem gerichteten Graphen	66
	Verwalten von Abhängigkeiten	70
	Implementierungen und Schnittstellen	71
	Der verdächtige Geruch von <i>new</i>	71
	Alternativen zur Objekterstellung	75
	Das Anti-Pattern Entourage	78
	Das Stairway-Pattern	80
	Auflösen von Abhängigkeiten	82
	Abhängigkeitsverwaltung mit NuGet	93
	Schichten	98
	Wichtige Patterns	99
	Grenzüberschreitende Angelegenheiten	106
	Asymmetrische Schichteinteilung	107
	Zusammenfassung	110
Kapitel 3	Schnittstellen und Entwurfs-Patterns	111
	Was ist eine Schnittstelle?	112
	Syntax	112
	Explizite Implementierung	115
	Polymorphie	120
	Adaptive Entwurfs-Patterns	121
	Das Null-Object-Pattern	122
	Das Adapter-Pattern	128
	Das Strategy-Pattern	131
	Mehr Vielseitigkeit	132
	Duck-Typing	133
	Mixins	137
	Flüssige Schnittstellen	143
	Zusammenfassung	144
Kapitel 4	Unit-Tests und Refaktorisierung	147
	Unit-Tests	148
	Arrange, Act, Assert	148
	Testorientierte Entwicklung	153
	Komplexere Tests	158

Refaktorisieren	175
Vorhandenen Code ändern	175
Ein neuer Kontotyp	185
Zusammenfassung	190

TEIL II Schreiben von SOLID-Code

Kapitel 5	Das Single-Responsibility-Prinzip	195
	Das Problem	196
	Refaktorisieren, um die Übersichtlichkeit zu verbessern	199
	Refaktorisieren, um Abstraktion zu erreichen	204
	SRP und das Decorator-Pattern	212
	Das Composite-Pattern	213
	Predicate-Decorator	218
	Branching-Decorator	221
	Lazy-Decorator	222
	Logging-Decorator	223
	Profiling-Decorator	225
	Asynchronous-Decorator	229
	Decorator für Eigenschaften und Ereignisse	231
	Verwenden des Strategy-Patterns statt <i>switch</i>	233
	Zusammenfassung	235
Kapitel 6	Das Open/Closed-Prinzip	237
	Einführung in das Open/Closed-Prinzip	237
	Die Meyer-Definition	237
	Die Martin-Definition	238
	Fehlerkorrekturen	238
	Lose Kopplung	239
	Erweiterungspunkte	239
	Code ohne Erweiterungspunkte	240
	Virtuelle Methoden	240
	Abstrakte Methoden	241
	Schnittstellenvererbung	242
	Vererbung in den Entwurf einarbeiten	243
	Geschützte Variation	244
	Prognostizierte Variation	244
	Eine stabile Schnittstelle	245
	Gerade genug Anpassungsfähigkeit	245
	Zusammenfassung	246

Kapitel 7	Das Liskovsche Substitutionsprinzip	247
	Einführung in das Liskovsche Substitutionsprinzip	247
	Formale Definition	248
	LSP-Regeln	248
	Verträge	249
	Vorbedingungen	251
	Nachbedingungen	252
	Dateninvarianten	253
	Liskovsche Vertragsregeln	255
	Codeverträge	263
	Kovarianz und Kontravarianz	271
	Definitionen	271
	Liskovsche Typsystemregeln	278
	Zusammenfassung	281
Kapitel 8	Interface-Segregation	283
	Ein Beispiel für Interface-Segregation	284
	Eine einfache CRUD-Schnittstelle	284
	Caching	290
	Dekorieren mehrerer Schnittstellen	294
	Verwenden von Schnittstellen in Clients	297
	Mehrere Implementierungen, mehrere Instanzen	297
	Einzelne Implementierung, einzelne Instanz	300
	Das Interface-Soup-Anti-Pattern	301
	Zerlegen von Schnittstellen	302
	Anforderungen des Clients	302
	Anforderungen der Architektur	309
	Schnittstellen mit einer einzigen Methode	313
	Zusammenfassung	315
Kapitel 9	Dependency-Injection	317
	Bescheidene Anfänge	318
	Die Aufgabenlistenanwendung	321
	Erstellen des Objektgraphen	324
	Inversion of Control	328

Jenseits der simplen Injection	344
Das Service-Locator-Anti-Pattern	344
Das Illegitimate-Injection-Anti-Pattern	348
Die Composition-Root	350
Konvention vor Konfiguration	356
Zusammenfassung	361

TEIL III **Entwickeln von adaptivem Code in der Praxis**

Kapitel 10	Beispiel für die Entwicklung von adaptivem Code: Einführung	365
	Trey Research	365
	Das Team	366
	Das Produkt	369
	Anfängliches Backlog	369
	User Stories in Prosabeschreibungen finden	370
	Abschätzen der Story-Punkte	372
	Zusammenfassung	377
Kapitel 11	Beispiel für die Entwicklung von adaptivem Code: Sprint 1	379
	Planung	379
	»Ich will Räume anlegen, um Unterhaltungen thematisch einzugrenzen.« ..	382
	Der Controller	382
	Das Raum-Repository	387
	»Ich will eine Liste der Räume ansehen, die Unterhaltungen repräsentieren.«	392
	»Ich will die Nachrichten ansehen, die an einen Raum geschickt wurden.« ..	397
	»Ich will reine Textnachrichten an andere Raummitglieder schicken.«	400
	Sprintvorführung	401
	Erste Vorführung von Proseware	402
	Sprint-Retrospektive	403
	Was lief gut?	403
	Was lief schlecht?	404
	Was sollte sich ändern?	405
	Was sollte beibehalten werden?	406
	Überraschungen?	407
	Zusammenfassung	408

Kapitel 12	Beispiel für die Entwicklung von adaptivem Code: Sprint 2	409
	Planung	409
	»Ich will Markdown senden, der richtig formatiert ist.«	411
	»Ich will Nachrichteninhalte so filtern, dass sie nicht anstößig sind.«	415
	»Ich will Hunderte von Benutzern gleichzeitig bedienen.«	419
	Sprintvorführung	421
	Sprint-Retrospektive	422
	Was lief gut?	422
	Was lief schlecht?	423
	Was sollte sich ändern?	423
	Was sollte beibehalten werden?	424
	Überraschungen?	424
	Zusammenfassung	424

TEIL IV Anhang

Anhang A	Adaptive Tools	427
	Quellcodeverwaltung mit Git	427
	Kontinuierliche Integration	432
Anhang B	GitHub-Codebeispiele	online
	Stichwortverzeichnis	435