

Auf einen Blick

Über die Autorin	9
Vorwort	11
Einleitung	25
Teil I: DevOps entmystifizieren	27
Kapitel 1: Einführung in DevOps	29
Kapitel 2: Gestalten Sie Ihre Organisation	37
Kapitel 3: Überflüssiges erkennen	51
Kapitel 4: Die Kollegen überzeugen, es mit DevOps zu probieren	61
Kapitel 5: Ihr Unternehmen beurteilen	75
Teil II: Eine Pipeline einrichten	85
Kapitel 6: Den neuen Entwicklungslebenszyklus übernehmen	87
Kapitel 7: Vorausplanen	95
Kapitel 8: Aus der DevOps-Perspektive designen	109
Kapitel 9: Code entwickeln	125
Kapitel 10: Tests vor der Veröffentlichung	141
Kapitel 11: Ein Produkt deployen	151
Teil III: Den Kreis schließen	169
Kapitel 12: Rapid Iteration implementieren	171
Kapitel 13: Feedback-Schleifen rund um den Kunden einrichten	185
Kapitel 14: DevOps-Teams zusammenstellen	197
Kapitel 15: Eigenverantwortung für die Entwickler	213
Teil IV: Kaizen: die Kunst der kontinuierlichen Verbesserung	225
Kapitel 16: Erfolgreich mit Fehlschlägen umgehen	227
Kapitel 17: Auf Zwischenfälle vorbereitet sein	235
Kapitel 18: Vorfälle nachträglich untersuchen	255
Teil V: Werkzeuge für Ihre DevOps-Praxis	263
Kapitel 19: Neue Tools	265
Kapitel 20: Verteilte Systeme	277
Kapitel 21: Migration in die Cloud	295
Teil VI: Der Top-Ten-Teil	307
Kapitel 22: (Mehr als) 10 wichtige Gründe für DevOps	309
Kapitel 23: Die zehn größten DevOps-Fallstricke	315
Stichwortverzeichnis	323

Inhaltsverzeichnis

Über die Autorin	9
Vorwort	11
Einleitung	25
Über dieses Buch	25
Törichte Annahmen über den Leser	25
Symbole in diesem Buch	26
Wie geht es weiter?	26
TEIL I	
DEVOPS ENTMYSTIFIZIEREN	27
Kapitel 1	
Einführung in DevOps	29
Was ist DevOps?	29
DevOps hat sich aus Agile entwickelt	30
DevOps stellt Menschen in den Mittelpunkt	30
Unternehmenskultur ist die Grundlage von DevOps	30
Sie lernen, indem Sie den Prozess überwachen und Daten sammeln	31
Überzeugungskraft ist der Schlüssel zur Umsetzung von DevOps	31
Kleine, inkrementelle Änderungen sind unbezahlbar	32
Von DevOps profitieren	32
Das CALMS-Modell	33
Das Problem der Interessenskonflikte lösen	35
Kapitel 2	
Gestalten Sie Ihre Organisation	37
Die Gesundheit Ihrer Unternehmenskultur bewerten	38
DevOps integrieren	39
Die DevOps-Werte im Einzelnen	40
Teamwork fördern	40
Silos reduzieren	41
Denken Sie systemorientiert	41
Fehlschläge akzeptieren	41
Kommunizieren, kommunizieren, kommunizieren	42
Rückmeldungen entgegennehmen	42
Abläufe automatisieren (falls sinnvoll)	43

Die Unternehmenskultur formen	43
Die schlimmsten Fehler der Technologiekultur vermeiden	45
Eine Vision entwerfen	46
Auf ein gemeinsames Ziel hinarbeiten	47
Beurteilungen	48
Prämien	49

Kapitel 3
Überflüssiges erkennen 51

Die sieben Arten von Verschwendung	52
Unnötige Abläufe	52
Wartezeiten	52
Bewegung	53
Kosten für Fehler	53
Überproduktion	53
Transport	53
Lagerbestand	54
Verschwendung in DevOps verstehen	54
Verschwendung vermeiden	56
Flaschenhalse erkennen	56
Auf die Auswirkungen konzentrieren	59

Kapitel 4
Die Kollegen überzeugen, es mit DevOps zu probieren 61

Angst vor Veränderungen	61
Die Leute um Sie herum vom Wechsel zu DevOps überzeugen	63
Unterstützung von Führungskräften erhalten	65
Eine Dünung im Entwicklungsteam aufbauen	66
Die mittleren Manager managen	67
Die Sturköpfe überzeugen	68
Die Adaptionkurve verstehen	69
Den Wandel vorantreiben	71
Auf Gegenwind reagieren	72
Den Abgrund überqueren	72
Fragen Sie »warum«.	73

Kapitel 5
Ihr Unternehmen beurteilen 75

DevOps quantifizieren	77
Menschen	77
Abläufe	78
Technologie	79
Die Daten erheben	80
Interne Fallstudien entwickeln	80
Eine qualitative Fallstudie: Konzentrieren Sie sich auf Ihre Mitarbeiter	81
Eine quantitative Fallstudie: Konzentrieren Sie sich auf Deployments	83

TEIL II EINE PIPELINE EINRICHTEN 85

Kapitel 6 Den neuen Entwicklungslebenszyklus übernehmen 87

Alle an einen Tisch bitten	87
Prozesse umwandeln: Von der Linie zum Kreis	88
Administrative Aufgaben »nach links« schieben: über Infrastruktur nachdenken	92
Auch Deployments nach links verschieben	93
Simulation der Produktion durch Staging	93

Kapitel 7 Vorausplanen 95

Über das Agile-Modell hinausgehen	95
Herausforderungen vorhersehen	97
Herausforderungen und Einschränkungen bei Projekten identifizieren	98
Zeitplan	98
Budget	99
Anforderungen bestimmen	99
Ein MVP entwickeln	100
Herausfinden, welches Problem Ihr MVP lösen muss	101
Herausfinden, wer Ihre Kunden sind	102
Die Konkurrenz unter die Lupe nehmen	102
Funktionen priorisieren	103
Die Benutzererfahrung gestalten	104
Ihre Hypothese überprüfen	105
Beta-Release, ja oder nein?	106
Personas entwerfen, um Ihre Kunden besser kennenzulernen	106
Was ist eine Persona?	107
Eine Persona ausarbeiten	107

Kapitel 8 Aus der DevOps-Perspektive designen 109

Ihr Design konstruieren	110
Für DevOps gestalten	112
Softwareentwicklung für den Wandel	112
Software kontinuierlich verbessern	113
Ihre Software dokumentieren	114
Codearchitektur für die sechs Leistungsmerkmale von DevOps	115
Wartungsfreundlichkeit	116
Skalierbarkeit	116
Sicherheit	118
Benutzerfreundlichkeit	119
Zuverlässigkeit	120
Flexibilität	120
Designentscheidungen dokumentieren	121
Fallstricke bei der Architektur vermeiden	122

Kapitel 9

Code entwickeln	125
Kommunikation rund um den Code	125
Für den Fehlerfall entwickeln	128
Wartungsfreundlichen Code schreiben	128
Code testen	129
Code debuggen	129
Code protokollieren	130
Unveränderbaren Code schreiben	130
Lesbaren Code erstellen	131
Programmiermuster	131
Objektorientierte Programmierung	131
Funktionale Programmierung	132
Eine Programmiersprache wählen	132
Anti-Patterns vermeiden	133
Nach DevOps-Prinzipien entwickeln	134
Sauberen Code schreiben	135
Das Geschäft verstehen	135
Anderen zuhören	135
Die richtigen Schwerpunkte setzen	136
Die Komfortzone verlassen	136
Gute Vorgehensweisen etablieren	137
Ordnung im Quellcode halten	137
Tests schreiben	137
Features dokumentieren	138
Legen Sie den Kollegen Ihren Code zur Kontrolle vor	139

Kapitel 10

Tests vor der Veröffentlichung	141
Warum Tests?	141
Automatisierte Tests sind nicht optional	142
Testen in verschiedenen Umgebungen	143
Lokale Umgebung	144
Entwicklungsumgebung	144
Testumgebung	145
Staging-Umgebung	146
Produktionsumgebung	146
Über den Komponententest hinaus	147
Komponententests: Es lebt!	147
Integrationstests: Spielen alle Teile zusammen?	148
Regressionstests: Verhält sich der Code nach Änderungen noch genauso?	148
Visuelle Tests: Sieht alles noch genauso aus?	148
Performance-Tests	149
Kontinuierliches Testen	149

Kapitel 11	
Ein Produkt deployen	151
Code freigeben.	151
Kontinuierliche Integration und Auslieferung.	152
Von CI/CD profitieren	152
CI/CD implementieren.	153
Kontinuierliche Integration.	153
Kontinuierliche Bereitstellung	154
Kontinuierliches Deployment.	154
Deployments managen.	155
Richtig automatisieren	155
Versionierung	156
Fehler abmildern	158
Rollbacks	158
Flucht nach vorne.	159
Deployments demokratisieren	159
Einen Deployment-Stil wählen.	160
Blue-Green-Deployment.	160
Schrödingers Kanarienvogel: Der Deploy ist tot (oder doch nicht?)	162
Rolling Release	163
Feature-Flags.	165
Ihre Systeme überwachen	165
Telemetrie verstehen.	166
Verhalten aufzeichnen.	166
SLAs, SLIs und SLOs	167
TEIL III	
DEN KREIS SCHLIEßEN	169
Kapitel 12	
Rapid Iteration implementieren	171
Wichtige Aufgaben zuerst.	172
Wichtig und dringend	173
Wichtig, nicht dringend	173
Dringend, nicht wichtig	175
Weder wichtig noch dringend	176
Schneller werden.	177
Die Performance verbessern	180
Unvollkommenheit akzeptieren.	181
Kleine Teams gestalten	181
Ihre Arbeit nachverfolgen	182
Reibung verringern	183
Warnmeldungen menschlicher gestalten.	183

Kapitel 13**Feedback-Schleifen rund um den Kunden einrichten 185**

Einen Kundenrückmeldungsprozess erstellen	186
Eine Feedback-Schleife erstellen	187
Empfangen	187
Analysieren	188
Kommunizieren	188
Verändern	189
Feedback sammeln	190
Umfragen zur Zufriedenheit.	190
Fallstudien.	191
Dogfooding: Selbstanwendung	191
Um kontinuierliche Rückmeldung bitten	193
Promotorenüberhang: Net Promoter Score (NPS)	194
Einen Rhythmus finden	194

Kapitel 14**DevOps-Teams zusammenstellen 197**

DevOps-Teams formen	197
So funktionieren funktionale Teams	198
Ein spezielles DevOps-Team bereitstellen	199
Funktionsübergreifende Produktteams bilden	200
Schnell zum Vorstellungsgespräch (aber nicht zu schnell)	202
Eine Stellenbezeichnung wählen	203
Die Personalbeschaffung endet nie	205
Die richtigen Leute finden.	206
Hervorragende Kandidaten weiterreichen.	206
Technische Fähigkeiten bewerten.	207
Überarbeitetes Whiteboarding	207
Hausaufgaben	208
Code-Reviews	209
Schnell feuern	209
Das Ekel	210
Der Märtyrer	211
Der Underperformer	211

Kapitel 15**Eigenverantwortung für die Entwickler 213**

Entwicklungsteams mit DevOps skalieren	213
Drei Phasen eines Unternehmens	214
Start-up	215
Etabliertes Start-up oder mittelständisches Unternehmen.	215
Großunternehmen.	216

Die Mühen der Ebene	218
Die Motivation ergründen	219
Motivation für Entwickler	220
Abhängigkeit von extrinsischen Belohnungen vermeiden	220
Autonomie	221
Meisterschaft	221
Sinnhaftigkeit	222
Arbeit zum Vergnügen machen	222
Den Leuten die Möglichkeit geben, ihre Teams auszuwählen	223
Motivation messen	223

**TEIL IV
KAIZEN: DIE KUNST DER KONTINUIERLICHEN
VERBESSERUNG 225**

**Kapitel 16
Erfolgreich mit Fehlschlägen umgehen 227**

Schnelles Scheitern im Tech-Bereich	227
Sicheres Scheitern	228
Fehlerausbreitung einschränken	228
Menschliches Versagen akzeptieren – und keine Schuldzuweisungen!	229
Gut scheitern	230
Wachstumsmentalität	230
Die Freiheit zum Scheitern schaffen	231

**Kapitel 17
Auf Zwischenfälle vorbereitet sein 235**

Mit Automatisierung gegen »menschliches Versagen« ankämpfen	236
Fokussierung auf Systeme: realistische Automatisierung	237
Mit Automatisierungstools Probleme bei der Codeintegration vermeiden	238
Deployments und Infrastruktur managen	240
Overengineering eingrenzen	240
Bereitschaftsdienste menschlicher gestalten	242
Wenn Bereitschaftsdienste unmenschlich werden	242
Humane Erwartungen an den Bereitschaftsdienst	243
Notfallmanagement	245
Beständigkeit zum Ziel machen	246
Standardverfahren einführen	247
Ein realistisches Budget ansetzen	248
Reaktion auf Vorfälle vereinfachen	248
Auf eine ungeplante Unterbrechung reagieren	249
Fortschritt empirisch messen	253
MTTR: Mean Time to Repair	253
MTBF: Mean Time between Failures	254
CPI: Cost per Incident	254

Kapitel 18
Vorfälle nachträglich untersuchen **255**

- Über die Analyse der Grundursache hinaus 255
- Die einzelnen Phasen eines Vorfalls durchgehen 257
- Vorfälle erfolgreich nachbereiten 258
 - Das Treffen sofort anberaumen 258
 - Alle miteinbeziehen 258
 - Schuldzuweisungen vermeiden 258
 - Den zeitlichen Ablauf betrachten 259
 - Schwierige Fragen stellen 260
 - Im Nachhinein sind Sie immer schlauer 261
 - Gesprächsprotokolle anfertigen 262
 - Einen Plan erstellen 262

TEIL V
WERKZEUGE FÜR IHRE DEVOPS-PRAXIS **263**

Kapitel 19
Neue Tools **265**

- Integration von Open-Source-Software 265
 - Open Computing als Innovationstreiber 266
 - Open-Source-Lizenzierung 267
 - Entscheidung für Open Source 268
- Auf neue Sprachen umstellen 270
 - Compiler- und Interpretersprachen 270
 - Parallelisierung und Multithreading 271
 - Funktionale Programmierung 272
 - Speicherverwaltung 273
 - Sprachen sinnvoll auswählen 273

Kapitel 20
Verteilte Systeme **277**

- Monolithen und Microservices 278
 - Zuerst eine monolithische Architektur wählen 279
 - Umstieg auf Microservices 280
- Großartige APIs entwickeln 281
 - Was ist eine API? 282
 - Auf einheitliches Design achten 282
- Container: Viel mehr als virtuelle Maschinen 285
 - Container und Images verstehen 286
 - Microservices in Containern deployen 286
 - Orchestrierer vergleichen: Die Harmonisierung des Schwarms 288
 - Container konfigurieren 290
 - Container überwachen: Halten Sie sie am Leben, bis Sie sie töten 291
 - Container absichern: Diese Kisten brauchen ein Schloss 292

Kapitel 21	
Migration in die Cloud	295
DevOps in der Cloud	295
Ihre DevOps-Kultur in die Cloud bringen	296
Lernen durch Übernahme	296
Von Cloud-Diensten profitieren	297
Arten von Clouds	298
Public Cloud	298
Private Cloud	299
Hybrid Cloud	299
Cloud as a Service	299
Infrastructure as a Service	300
Platform as a Service	300
Software as a Service	301
Den besten Cloud-Anbieter wählen	301
Amazon Web Services (AWS)	302
Microsoft Azure	302
Google Cloud Platform (GCP)	303
Tools und Services in der Cloud finden	303
TEIL VI	
DER TOP-TEN-TEIL	307
Kapitel 22	
(Mehr als) 10 wichtige Gründe für DevOps	309
Beständigen Wandel akzeptieren	309
Die Cloud nutzen	310
Die Besten einstellen	310
Wettbewerbsfähig bleiben	311
Menschliche Probleme lösen	311
Mitarbeiter fordern	312
Brücken schlagen	312
Gut scheitern	312
Kontinuierliche Verbesserung	313
Mühsame Arbeiten automatisieren	314
Auslieferung beschleunigen	314
Kapitel 23	
Die zehn größten DevOps-Fallstricke	315
Kultur vernachlässigen	315
Nicht alle mitnehmen	316
Anreize schlecht aufeinander abstimmen	316
Stillschweigen	317
Vergessen zu messen	318

24 Inhaltsverzeichnis

Micromanaging	318
Zu schnell zu viel verändern	319
Schlechte Werkzeugauswahl	319
Angst vor Misserfolgen	320
Zu hart sein	320
Stichwortverzeichnis	323