
Inhaltsverzeichnis

1	Einleitung	1
1.1	Kennen Sie die »Integrationshöhle«?	1
1.2	Warum Continuous Integration (CI)?	3
1.3	Warum Hudson?	4
1.4	Warum dieses Buch?	6
1.5	Wer dieses Buch lesen sollte	7
1.6	Wie man dieses Buch lesen sollte	9
1.7	Konventionen	10
1.8	Website zum Buch	10
1.9	Danksagungen	10
1.10	Zusammenfassung	11
2	CI in 20 Minuten	13
2.1	Was ist CI?	13
2.2	Was ist CI nicht?	16
2.3	Software entwickeln ohne CI	18
2.4	Software entwickeln mit CI	19
2.5	Zusammenfassung	20
3	Welche Vorteile bringt CI?	23
3.1	Reduzierte Risiken	23
3.2	Verbesserte Produktqualität	24
3.3	Allzeit auslieferbare Produkte	25
3.4	Gesteigerte Effizienz	26
3.5	Dokumentierter Build-Prozess	27
3.6	Höhere Motivation	28
3.7	Verbesserter Informationsfluss	29

3.8	Unterstützte Prozessverbesserung	29
3.9	Zusammenfassung	29
4	Die CI-Praktiken	31
4.1	Gemeinsame Codebasis	31
4.2	Automatisierter Build	32
4.3	Häufige Integration	34
4.4	Selbsttestender Build	35
4.4.1	Compiler	35
4.4.2	Unit-Tests	35
4.4.3	Komponententests	36
4.4.4	Systemtests	36
4.4.5	Inspektionen	37
4.5	Builds (und Tests) nach jeder Änderung	38
4.6	Schnelle Build-Zyklen	38
4.7	Tests in gespiegelter Produktionsumgebung	39
4.8	Einfacher Zugriff auf Build-Ergebnisse	40
4.9	Automatisierte Berichte	41
4.10	Automatisierte Ausbringung (Deployment)	42
4.11	Zusammenfassung	43
5	Hudson im Überblick	45
5.1	Die Hudson-Story	45
5.2	Architektur und Konzepte	46
5.2.1	Systemlandschaft	47
5.2.2	Datenmodell	50
5.2.3	Benutzerschnittstellen	55
5.3	Die Top-10-Highlights	60
5.3.1	Schnelle Installation	60
5.3.2	Effiziente Konfiguration	60
5.3.3	Unterstützung zahlreicher Build-Werkzeuge	61
5.3.4	Anbindung von Versionsmanagementsystemen ..	61
5.3.5	Testberichte	62
5.3.6	Benachrichtigungen	63
5.3.7	Remoting-Schnittstelle	64
5.3.8	Abhängigkeiten zwischen Jobs	64
5.3.9	Multikonfigurationsbuilds (»Matrix-Builds«) ..	65
5.3.10	Verteilte Builds	66
5.3.11	Plugins	67

5.4	Hudson im Vergleich zu Mitbewerbern	67
5.4.1	Proprietäre Eigenentwicklungen	68
5.4.2	CruiseControl	69
5.4.3	ThoughtWorks Cruise	72
5.4.4	Atlassian Bamboo	76
5.4.5	JetBrains TeamCity	81
5.5	Zusammenfassung	86
6	Installieren und Einrichten	87
6.1	Schnellstart in 60 Sekunden	87
6.2	Systemkonfiguration	89
6.3	Fortgeschrittene Installation	90
6.3.1	Systemvoraussetzungen	91
6.3.2	Ablaufumgebungen	92
6.3.3	Datei-Layout	93
6.3.4	Installationsmethoden	95
6.3.5	Starten und Stoppen	96
6.3.6	Aktualisieren	99
6.3.7	Backups	100
6.3.8	Deinstallation	103
6.3.9	Plugins	103
6.3.10	Tipps aus der Praxis	104
6.4	Zusammenfassung	106
7	Hudson im täglichen Einsatz	107
7.1	Ihr erster Hudson-Job im Schnelldurchlauf	108
7.1.1	Den ersten Job einrichten	108
7.1.2	Statusanzeigen auf der Übersichtsseite	112
7.2	Jobtypen	114
7.2.1	Gemeinsamkeiten	114
7.2.2	Free-Style-Projekt	116
7.2.3	Maven-2-Projekt	117
7.2.4	Multikonfigurationsprojekt (»Matrix-Builds«) ..	118
7.2.5	Überwachung eines externen Jobs	118
7.3	Versionsmanagement integrieren	120
7.3.1	Konfiguration eines Projekts	120
7.3.2	Repository-Browser anbinden	124
7.4	Auslösen von Builds	126
7.4.1	Manuell (build trigger)	127
7.4.2	Zeitgesteuert	127
7.4.3	Bei Änderungen im Versionsmanagementsystem	128
7.4.4	Abhängigkeiten zu anderen Projekten	129
7.4.5	Auslösung durch Plugins	131

7.5	Testwerkzeuge integrieren	131
7.5.1	JUnit, TestNG	132
7.5.2	Weitere Test-Frameworks der xUnit-Familie . . .	135
7.6	Benachrichtigungen verschicken	136
7.6.1	E-Mail	136
7.6.2	Email-ext (Plugin)	138
7.6.3	RSS-Feeds	140
7.6.4	Hudson Build Monitor (Firefox Add-on)	141
7.6.5	Twitter (Plugin)	141
7.6.6	hudson-eclipse (Eclipse-Plugin)	142
7.6.7	Informationsradiatoren	143
7.6.8	eXtreme-Feedback-Geräte anbinden	144
7.7	Dokumentationswerkzeuge integrieren	146
7.7.1	Javadoc	146
7.7.2	Maven Site	148
7.7.3	Einbindung beliebiger HTML-Berichte	148
7.8	Analysewerkzeuge integrieren	151
7.8.1	Checkstyle, PMD, FindBugs	151
7.8.2	Warnings (Plugin)	156
7.8.3	Task Scanner (Plugin)	156
7.8.4	Cobertura (Plugin)	158
7.8.5	Sonar (Plugin)	161
7.9	Issue-Tracker integrieren	163
7.9.1	Atlassian JIRA	164
7.9.2	Mantis	168
7.10	Zusammenfassung	169
8	Hudson für Fortgeschrittene	171
8.1	Parametrisierte Builds	171
8.1.1	Definition von Parametern	172
8.1.2	Typische Anwendungsfälle	173
8.2	Ansichten (views) und Dashboards	175
8.2.1	Benutzerdefinierte Listenansichten	175
8.2.2	Projekte filtern und gruppieren	176
8.2.3	Verschachtelte Ansichten	178
8.2.4	Hudson als Ihr Build-Portal	178
8.3	Parallelisierung der Build-Aktivitäten	179
8.3.1	Abhängigkeiten zwischen Projekten	181
8.3.2	Nachgelagerte Builds zusammenführen	185
8.3.3	Exklusiver Zugriff auf Ressourcen	186
8.3.4	Parameterübergabe an nachgelagerte Builds	187
8.3.5	Abbruch hängender Builds	188

8.4	Multikonfigurationsprojekte	189
8.4.1	Anlegen eines Multikonfigurationsprojekts	190
8.4.2	Verteilte Matrix-Builds	192
8.4.3	Zusammenfassung der Build-Ergebnisse	193
8.5	Verteilte Builds	194
8.5.1	Warum verteilen?	194
8.5.2	Warum nicht verteilen?	194
8.5.3	Slave-Knoten einrichten	195
8.5.4	Verteilen der Build-Umgebung	198
8.5.5	Überwachen des CI-Clusters	200
8.5.6	Virtualisierung	201
8.5.7	Bauen in der Wolke	206
8.6	Hudson absichern	212
8.6.1	Authentifizierung	213
8.6.2	Autorisierung	217
8.6.3	Auditierung	218
8.7	Weitere nützliche Plugins	220
8.7.1	Claim	220
8.7.2	Configuration Slicing	221
8.7.3	Build-Promotion	223
8.7.4	Description Setter	225
8.7.5	Green Balls	226
8.7.6	Locale	227
8.7.7	Continuous Integration Game	227
8.8	Zusammenfassung	228
9	Hudson erweitern	229
9.1	Erste Schritte	230
9.1.1	Schnellstart: Ihr erstes Plugin in 60 Sekunden ..	230
9.1.2	Anatomie eines Plugins	234
9.1.3	Einrichten einer Entwicklungsumgebung	240
9.1.4	Zusammenfassung	245
9.2	Hudsons Plugin-Konzept	245
9.2.1	Hudsons Technologie-Stapel	245
9.2.2	Was muss ein Plugin-Entwickler wissen?	248
9.2.3	Hudsons Erweiterungspunkte (Extension Points)	250

9.3	Beispiel: Das Plugin »Artifact Size«	263
9.3.1	Die Aufgabe	263
9.3.2	Erweiterungspunkt »ListViewColumn«	264
9.3.3	Erweiterungspunkte »Recorder« und »Action«	269
9.3.4	Erweiterungspunkt »CLICommand«	272
9.3.5	Bereitstellen der Artefaktgröße per XML-API	275
9.3.6	Online-Hilfetexte	276
9.3.7	Lokalisierung und Internationalisierung	277
9.4	Zusammenfassung	282
10	Aufwand einer CI-Einführung	283
10.1	Erstaufwand für Vollautomatisierung des Builds	283
10.2	Erstaufwand für Erstellung von fehlenden Tests	284
10.3	Optimieren des Build-Prozesses für CI	285
10.4	Hardwareressourcen für ein CI-System	285
10.5	Installation des CI-Systems	286
10.6	Administration des CI-Systems	286
10.7	Schulung	287
10.8	Zusammenfassung	288
11	Tipps zur Einführung von CI	289
11.1	Mit einem Pilotprojekt starten	289
11.2	Build-Zeiten kurz halten	290
11.3	Codeanalyse: Sanft starten, konstant steigern	290
11.4	Nur messen, was auch beachtet wird	291
11.5	Erscheinen neuer Plugins verfolgen	291
11.6	Den Spaß nicht vergessen	292
12	Fazit und Ausblick	293
12.1	Continuous Integration – wie geht's weiter?	293
12.2	Hudson – wie geht's weiter?	294
12.3	Zusammenfassung	296
	Literatur und Quellen	297
	Index	299