

Inhaltsverzeichnis

1	Einleitung	1
1.1	Wer dieses Buch lesen sollte	2
1.2	Warum Subversion, Maven und Redmine?	3
1.3	Wo ist das Ant-Kapitel?	5
1.4	Abgrenzung und Begriffserläuterungen	5
1.5	Aufbau des Buches	7
1.6	Beispielprojekt e2etrace	7
1.7	Konventionen	8
1.8	Webseite zum Buch	9
1.9	Danksagung	9
2	Einführung in das Konfigurationsmanagement	11
2.1	Was ist Konfigurationsmanagement?	12
2.1.1	Ziele des Konfigurationsmanagements	12
2.1.2	Argumente für den Einsatz im Projekt	17
2.1.3	Normen und Standards	20
2.2	Aufgaben und Verfahren des Kernprozesses	22
2.2.1	Auswahl der Konfigurationselemente	22
2.2.2	Erstellung des Konfigurationsmanagement- Handbuches	26
2.2.3	Beschreibung der Konfigurationselemente	30
2.2.4	Festlegung der Projektstruktur	32
2.2.5	Verwaltung der Konfigurationselemente	35
2.2.6	Projektautomatisierung	46
2.2.7	Änderungs- und Fehlermanagement	50

- 2.3 Erweiterter Prozess 56
 - 2.3.1 Releasemanagement 56
 - 2.3.2 Audits 60
 - 2.3.3 Metriken 61
 - 2.3.4 Berichte 68
- 3 Konfigurationsmanagement-Werkzeuge 71**
 - 3.1 Subversion 72
 - 3.1.1 Funktionsumfang 72
 - 3.1.2 Architektur 74
 - 3.1.3 Bezugsquelle und Installation 76
 - 3.1.4 Dokumentation 76
 - 3.2 Maven 77
 - 3.2.1 Funktionsumfang 78
 - 3.2.2 Architektur 80
 - 3.2.3 Bezugsquelle und Installation 82
 - 3.2.4 Dokumentation 82
 - 3.3 Redmine 82
 - 3.3.1 Funktionsumfang 83
 - 3.3.2 Architektur 83
 - 3.3.3 Bezugsquelle und Installation 84
 - 3.3.4 Dokumentation 85
 - 3.4 Nexus 85
 - 3.4.1 Funktionsumfang 85
 - 3.4.2 Architektur 86
 - 3.4.3 Bezugsquelle und Installation 86
 - 3.4.4 Dokumentation 87
 - 3.5 Hudson 87
 - 3.5.1 Funktionsumfang 88
 - 3.5.2 Bezugsquelle und Installation 88
 - 3.5.3 Dokumentation 90
 - 3.6 Weitere Werkzeuge und Frameworks 90
- 4 Verwaltung der Konfigurationselemente mit Subversion 93**
 - 4.1 Einrichten des Repositorys 93
 - 4.1.1 Arbeitsweise des Repositorys 94
 - 4.1.2 Erstellung eines Repositorys 95
 - 4.1.3 Benutzer und Zugriffsrechte festlegen 97
 - 4.1.4 Zugriff auf das Repository 100

4.2	Projektstruktur und Konfiguration festlegen	102
4.2.1	Einfluss des Releaseplans	103
4.2.2	Vorbereitung der Tags und Branches	104
4.2.3	Festlegung der detaillierten Struktur	105
4.2.4	Konfiguration des Clients	106
4.2.5	Anlegen der Struktur im Repository	110
4.2.6	Check-out des Arbeitsbereiches	111
4.3	Durchführen von Änderungen	113
4.3.1	Arbeitsbereich aktualisieren	114
4.3.2	Dateien ändern	115
4.3.3	Änderungen rückgängig machen	115
4.3.4	Elemente hinzufügen	116
4.3.5	Elemente löschen	117
4.3.6	Elemente kopieren	118
4.3.7	Elemente verschieben und umbenennen	121
4.3.8	Änderungen überprüfen	121
4.3.9	Änderungen in das Repository schreiben	125
4.4	Arbeiten mit der Versionshistorie	128
4.4.1	Abfrage der Versionshistorie	129
4.4.2	Auswählen einer Revision	131
4.4.3	Verwendung von Peg-Revisions	132
4.4.4	Unterschiede zwischen Versionen ermitteln	135
4.4.5	Unterschiede in der Projektstruktur ermitteln ...	140
4.5	Umgang mit Konflikten	141
4.5.1	Entstehung von Konflikten	142
4.5.2	Automatische Auflösung einfacher Konflikte ...	142
4.5.3	Behebung von echten Konflikten	145
4.6	Verwaltung binärer Dateien	152
4.6.1	Was sind binäre Dateien?	153
4.6.2	Konflikte in binären Dateien	155
4.6.3	Vermeidung von Konflikten	160
4.7	Festlegung von Tags	165
4.8	Arbeiten mit Branches	168
4.8.1	Einen Branch erstellen	168
4.8.2	Änderungen in einem Branch durchführen	169
4.8.3	Branches zusammenführen	170
4.8.4	Fehlerquellen beim Zusammenführen	181
4.8.5	Binäre Dateien und Branches	186
4.8.6	Einen Branch abschließen	187
4.8.7	Alternative Branch-Strategien	189

5	Projektautomatisierung mit Maven	193
5.1	Umsetzung eines einfachen Build-Prozesses	193
5.1.1	Aufbau des Projektmodells	194
5.1.2	Benutzerspezifische Einstellungen festlegen	196
5.1.3	Quellelemente ermitteln	197
5.1.4	Projektexterne Dateien einbinden	199
5.1.5	Produkt erstellen	206
5.1.6	Produkt prüfen	215
5.1.7	Produkt ausliefern	217
5.1.8	Zusammenfassung und Ausblick	223
5.2	Einführung von Build-Varianten	223
5.2.1	Prinzipielle Vorgehensweise	224
5.2.2	Einrichtung von Nexus	224
5.2.3	Verwendung von Nexus	230
5.2.4	Entwickler-Build	239
5.2.5	Integrations-Build	240
5.2.6	Release-Build	267
5.3	Qualitätssicherung durch Audits und Metriken	274
5.4	Einrichtung einer Projekt-Homepage	278
6	Änderungsmanagement mit Redmine	289
6.1	Einrichten eines Projektes	289
6.1.1	Konfiguration des SMTP-Servers	289
6.1.2	Benutzer einrichten	291
6.1.3	Rollen und Rechte definieren	295
6.1.4	Tracker, Ticketstatus und Workflows	296
6.1.5	Globale Einstellungen	302
6.1.6	Projekt anlegen	304
6.1.7	Projekt konfigurieren	306
6.2	Arbeiten mit Tickets	309
6.2.1	Spielregeln festlegen	309
6.2.2	Tickets erstellen	310
6.2.3	Tickets verifizieren	313
6.2.4	Tickets zuweisen	316
6.2.5	Tickets filtern	317
6.2.6	Tickets implementieren	319
6.2.7	Implementierung eines Tickets prüfen	324
6.2.8	Tickets schließen	327

6.3	Projektdokumentation	327
6.3.1	Projekthistorie	328
6.3.2	Releaseplanung und -dokumentation	329
6.3.3	Datei- und Dokumentenarchiv	331
6.3.4	Wiki	331
6.3.5	Foren	335
6.3.6	News	336
6.4	Fazit	336
	Literatur und Links	337
	Stichwortverzeichnis	341