Kapitel 4

Wie macht man ...

Obwohl sich Versionsverwaltung in der Theorie sehr gut anhört, benutzen viele Teams keine. Manchmal, weil sie vermuten, dass sich die Theorie nicht sehr gut in die Praxis umsetzen lässt. Ein Dokument, in dem etwas über das „Erzeugen eines Release-Zweiges“ erzählt wird, ist schnell gelesen. Doch was bedeutet das, wenn man wirklich CVS-Befehle eingeben muss?

Ein weiteres Problem ist, dass Teams manchmal Versionsverwaltung zu weit treiben und dabei sehr komplexe Strukturen zur Aufbewahrung ihres Quelltextes erzeugen. Dazu gibt es oft eine beängstigende Liste von Instruktionen selbst für die einfachsten Aufgaben. Das Ergebnis? Letztendlich (und, wie die Erfahrung zeigt, sehr schnell) gibt das Team auf, da die Versionsverwaltung als zu aufwändig betrachtet wird.

Die verbleibenden Kapitel in diesem Buch werden beide Probleme behandeln. Sie präsentieren einen einfachen Weg zur Organisation Ihrer Versionsverwaltung und eine Menge von grundlegenden Techniken zur Erledigung alltäglicher Aufgaben, mit denen ein Team konfrontiert wird. Wir schlagen vor, dass Sie diese grundlegenden Techniken als eine Sammlung von Rezepten ansehen, denen Sie folgen, wenn Sie ein bestimmtes Ergebnis erreichen wollen. Versuchen Sie, nicht zu stark davon abzuweichen. Wenn Sie feststellen, dass Sie ein Szenario erzeugen wollen, das wir nicht behandelt haben, denken Sie gründlich darüber nach, bevor Sie fortfahren. Wahrscheinlich brauchen Sie es nicht wirklich.

Wie mit jeder Sammlung von Rezepten werden Sie sich bald mehr und mehr mit ihnen vertraut fühlen. Das ist der Zeitpunkt, an dem Sie anfangen können, ein wenig zu experimentieren. Wir schlagen trotzdem vor, neue Dinge nicht im Repository eines echten Projektes auszuprobieren. Stellen Sie stattdessen das Szenario in einem Test-Repository nach (wie das aus dem vorhergehenden Kapitel) und probieren Sie diese Dinge dort aus.

4.1 Unsere grundlegende Philosophie

Wir denken, dass Versionsverwaltung eine der drei unentbehrlichen technischen Fertigkeiten ist. Jedes Team muss in allen dreien geübt sein (die anderen beiden sind Unit-Tests mit JUnit [PUT03], [PUT04] und Projektautomatisierung [PPA04], [PPA05]). Jedes Team sollte eine Versionsverwaltung benutzen, immer und für alles, was es erzeugt. Also müssen wir es einfach gestalten, durchschaubar und leichtgewichtig. Denn wenn wir das nicht tun, werden die Leute es irgend wann nicht mehr benutzen.

Einfachheit bedeutet, dass etwas, das einfach sein *soll*, auch tatsächlich einfach ist. Alle meine Änderungen einzuchecken, ist ein einfacher (und alltäglicher) Ablauf, also sollte dieser einfache Ablauf aus ein oder zwei Aktionen bestehen. Ein neues Release für den Kunden zu erstellen, ist eine etwas kompliziertere Angelegenheit. Also ist es akzeptabel, dafür ein paar Schritte mehr durchzuführen. Trotzdem sollte es so einfach wie möglich sein.

Versionsverwaltung muss *durchschaubar* sein: Wir müssen die Dinge so einrichten, dass deutlich wird, woran und an welcher Version wir arbeiten. Es sollte nicht zum Rätselraten kommen, wenn es um den Quelltext geht.

Letztendlich beschreiben wir einen leichtgewichtigen Prozess: Wir wollen nicht, dass die Versionsverwaltung zwischen uns und der zu erledigenden Arbeit steht.

4.2 Eine Versionsverwaltung organisieren

Nun zu unseren grundlegenden Regeln für die Organisation Ihres Quelltextes in einem CVS-Repository:

- Bevor Sie loslegen, müssen Sie eine effektive und sichere Methode zum Zugriff auf das Repository einrichten. *siehe Seite 59*
- Sobald Sie Zugriff darauf haben, gibt es einen einfachen Satz von CVS-Befehlen, den Sie täglich benutzen werden. *siehe Seite 67*
- Jedes Projekt, das Ihre Firma entwickelt, muss als eigenständiges CVS-Modul vorhanden sein. Es sollte möglich sein, den gesamten Quelltext eines Projektes von einer einzigen Stelle aus zu checken. *siehe Seite 117*
- Wenn ein Projekt Komponenten enthält, an denen isoliert gearbeitet werden kann, oder wenn Sie die Absicht haben, Komponenten gemeinsam mit anderen Projekten zu benutzen, sollten diese Komponenten in benannten Modulen abgelegt werden. *siehe Seite 127*
- Wenn in Ihrem Projekt Quelltext von Fremdanbietern verwendet wird (gekauft oder vielleicht von Open-Source-Projekten), müssen Sie diesen als Ressource verwalten. *siehe Seite 141*
- Entwickler verwenden Tags, um signifikante Punkte zu identifizieren. Dazu zählen Releases, Fehlerkorrekturen und der Beginn größerer Experimente am Quelltext. *siehe Seite 103*