

HANSER

**Pragmatisch  
Programmieren  
Projekt-Automatisierung**

von Michael Clark

ISBN 3-446-40008-7

Leseprobe Kapitel 4.7

Weitere Informationen oder Bestellung  
unter <http://www.hanser.de/3-446-40008-8>  
sowie im Buchhandel

## 4.7 Automatisieren der Release-Prozedur

„Moment mal“, höre ich die Leute sagen, „wo ist bei all dem die Automatisierung? Unsere Fingerspitzen sind schon ganz wund von dem vielen Tippen auf der Kommandozeile. Dabei haben wir doch gelernt, dass Wiederholbarkeit nicht durch Eintippen einer Reihe von Befehlen auf der Kommandozeile erreicht wird. Warum machen wir dann alles manuell und automatisieren es nicht von Anfang an?“

Bevor Sie einen Ablauf automatisieren können, müssen Sie ihn auch manuell ausführen können. Sie können dem Computer nicht etwas beibringen, wenn Sie vorher nicht selbst wissen, wie es geht. Aber hier kommt die gute Nachricht für Sie: Der Computer wird das *nächste* Release für Sie erstellen. Denn alles, was Sie auf der Kommandozeile ausführen können, kann der Computer besser, wenn Sie ihm sagen wie.

### Rekapitulieren unseres Vorgehens

Lassen Sie uns die Schritte zum Erstellen unseres ersten Releases auf der Kommandozeile noch einmal zusammenfassen:

1. Den Inhalt des Hauptstranges testen.
2. Einen Release-Zweig erstellen.
3. Den Release-Zweig auschecken.
4. Den Release-Zweig bauen und testen.
5. Eine Distributionsdatei für das Release erstellen.
6. Den Inhalt der Distributionsdatei testen.
7. Ein Release-Tag erstellen.
8. Übergabe der Distributionsdatei an das Qualitätsmanagementteam.

Diese Abfolge von Schritten ist unsere Release-Prozedur. Es ist eine Menge Arbeit. Und wenn wir all diese Dinge jedes Mal beim Erstellen eines Release tun müssen, werden wir ein Release nicht so oft anlegen wollen. Wenn wir es doch tun, werden wir wahrscheinlich etwas durcheinander bringen, da das Ausführen detaillierter Anweisungen zu Langeweile und Fehlern neigt.

### Ausführen der Release-Prozedur per Skript

Aber noch ist nicht alles verloren. Werfen Sie einen kurzen Blick auf den zurückliegenden Ablauf. Die einzige Variable in der gesamten Gleichung ist die Versionsnummer. Diese Woche ist es Version 1.0 und nächste Woche sind wir vielleicht mit Version 1.1 fertig. Wir können also die gesamte Release-Prozedur mit zwei Skripten automatisieren.

Sie führen jedes Mal die Schritte eins bis vier aus, wenn Sie einen neuen Release-Zweig erstellen wollen. Sobald Sie den Release-Zweig haben, erstellen die Schritte fünf bis acht ein Release auf diesem Zweig. Sie könnten z.B. mehrere Releases vom selben Release-Zweig 1.0 erstellen, wobei jedes Release eine Fehlerkorrektur enthält. Das bedeutet aber, dass Sie diese zwei Prozeduren in unterschiedlichen Abständen ausführen.

Abbildung 4.8 auf Seite 102 zeigt das Skript `release_branch.sh`, das auf Knopfdruck einen Release-Zweig (Schritte 1–4) erstellt. Zum Erstellen eines Release-Zweiges für das Release 1.1 geben Sie bitte Folgendes ein:

```
$ sh release_branch.sh 1_1
```

Abbildung 4.9 auf Seite 103 zeigt das Skript `release_generate.sh`, mit dem das Release vom Release-Zweig (Schritte 5–8) erstellt wird. Zum Erzeugen einer Distributionsdatei für das Release 1.1 geben Sie bitte dies ein:

```
$ sh release_generate.sh 1_1
```

Beide Skripte setzen voraus, dass die Umgebungsvariable `CVSROOT` auf Ihr CVS-Repository zeigt. In Kombination mit einer speziellen Versionsnummer führen sie zusammen alle acht Schritte der Release-Prozedur genauso aus, wie wir es auf der Kommandozeile getan haben. Wenn der jeweilige Build-Schritt zum Testen in einem der beiden Skripte fehlschlägt, wird das Release nicht gebaut.

Natürlich sind Unix-Shell-Skripte nicht der einzige Weg, die Release-Prozedur zu automatisieren. Sie könnten auch eine Windows-Batch-Datei, eine Ant-Build-Datei oder sogar eine mächtige Skriptsprache nehmen. Ebenso wird Ihre Release-Prozedur anders aussehen, wenn Sie nicht CVS benutzen. Wie Sie zu einer automatisierten Release-Prozedur kommen, spielt keine Rolle. Wichtig ist nur, dass sie zu einer kommen.