HANSER

Ship it!

von Jared R. Richardson
und William A. Gwaltney

ISBN 3-446-40425-2

Leseprobe Kapitel 5.16

16 Hilfe! Ich habe Quelltext einer Alt-Anwendung geerbt

Sie haben eine Alt-Anwendung geerbt, die Sie pflegen und erweitern sollen. Wie bekommen Sie das am schnellsten in den Griff? Lernen Sie, die Anwendung zu bauen, zu automatisieren und schließlich zu testen.

1. *Bauen*: Finden Sie als erstes heraus, wie die Anwendung gebaut wird, und schreiben Sie dann ein Skript für den Build-Prozess. Diese Aufgabe ist nicht immer leicht, besonders wenn der Quelltext bisher nur einen Eigentümer hatte. Der Build-Prozess wird oft nur auf einem Rechner lauffähig sein, da er auf eine spezifische Umgebung angepasst ist. Wenn Sie diese Aufgabe geschafft haben, kann jeder das Produkt auf jedem Rechner bauen. Danach sollte es nicht mehr schwierig sein, den Build-Prozess zu automatisieren.
2. *Automatisieren*: Ihr Ziel ist es, die gesamte Anwendung automatisch zu bauen und zu testen, auf einem sauberen Rechner und mit einem Minimum an manuellen Eingriffen. Wir haben bewusst nicht gefordert, auf manuelle Eingriffe *völlig* zu verzichten; hier gilt es ein Gleichgewicht zu finden. Manchmal ist es einfacher, eine Komponente der Umgebung manuell zu installieren und zu konfigurieren, als ein Automatisierungsskript zu schreiben. Anwendungen, die Sie nur ein Mal installieren (z.B. Compiler), sind erstklassige Kandidaten dafür. Dokumentieren Sie alle Build-Schritte und machen Sie diese Dokumentation allgemein zugänglich.
3. *Testen*: Sobald der Build-Prozess ordentlich durchläuft, wollen Sie auch bestätigen, dass Ihre Anwendung richtig funktioniert. Finden Sie heraus, was der Quelltext tut, und beginnen Sie dann mit dem Testen, indem Sie Mock-Client-Tests dafür schreiben (siehe Box auf Seite 53). Mock-Client-Tests sind ein guter Ausgangspunkt: Sie testen die allgemeine Funktionalität einer Anwendung, indem sie genau das tun, was auch ein Nutzer tun würde.

4. *Weiter testen*: Gewinnen Sie ein Verständnis für das Innenleben der Anwendung (wie z.B. Struktur, Kontrollfluss, Performance und Skalierbarkeit) und schreiben Sie weitere Tests dafür. Sofern die Anwendung nicht völlig ungenutzt ist, *wird* es darin Fehler geben, die Sie beheben (oder zumindest dokumentieren), oder Erweiterungen, die Sie vornehmen müssen. Normalerweise sind solche Änderungen an Alt-Anwendungen ziemlich Furcht einflößend, da Sie nie wissen, welche Nebenwirkungen Ihre Änderungen haben können. Dank der Mock-Client-Tests, die Sie als Sicherheitsnetz geschrieben haben, können Sie dies aber nun furchtlos tun; sie verhindern, dass Sie bei Ihren Änderungen allzu großen Schaden anrichten. (Sie *haben* diese Tests geschrieben, oder?)
   Schreiben Sie einen Test für jeden Fehler, den Sie beheben, und für jede Erweiterung, die Sie vornehmen. Die Art des Tests wird davon abhängen, was Sie geändert haben (z.B. ein Unit-Test für die Änderung einer systemnahen internen Funktion oder ein Mock-Client-Test für ein neues Feature). An dieser Stelle behandeln Sie dann eine Alt-Anwendung genauso wie jedes andere Produkt, das Sie betreuen.

Nachdem Sie mit all dem fertig sind, wird *jeder* mit diesem Quelltext arbeiten können! Jeder wird in der Lage sein, ihn überall automatisch zu bauen, und dann mit Hilfe der automatischen Tests festzustellen, dass alles richtig funktioniert. Und das automatische Build-System wird in einer sauberen Umgebung den Build-Lauf und die Tests erneut durchführen und damit sicherstellen, dass alles *immer noch* funktioniert.

Tipp 25: *Ändern Sie keine Alt-Anwendung, bevor Sie sie nicht testen können*