

Vorwort

Willkommen in der Welt der Entwickler-fokussierten Unit-Tests. Wir hoffen, dieses Buch wird Ihnen und Ihrem Team eine wichtige Wissensquelle sein. Wenn Sie uns mitteilen möchten, wie Ihnen das Buch geholfen hat oder was wir besser machen können, besuchen Sie bitte die Webseite zum Buch¹ und klicken auf „Feedback“.

Derartige Rückmeldungen machen Bücher erst wirklich gut. Das gilt auch für Menschen und Projekte. Pragmatisch Programmieren bedeutet, das Herangehen ständig anhand realer Rückmeldungen neu abzustimmen und anzupassen.

Und schon sind wir bei den Unit-Tests. Sie werden erleben, dass Unit-Tests für Sie als Programmierer wichtig sind, weil sie die Rückmeldungen liefern, die Sie brauchen. Ohne Unit-Tests könnten Sie ein Programm auch auf Papier schreiben und sich Glück wünschen, wenn es einmal ausgeführt wird.

Das ist nicht sehr pragmatisch.

Dieses Buch kann helfen. Es richtet sich hauptsächlich an jene Java-Programmierer, die schon ein paar Erfahrungen mit Entwurf und Implementierung sammeln konnten, aber noch keine Erfahrungen mit Unit-Tests haben.

Obwohl die Beispiele in diesem Buch in Java sind und dabei das JUnit-Framework verwenden, sind die Konzepte sprachunabhängig. Sie gelten ebenso für C++, Fortran, Ruby, Smalltalk und VisualBasic. Test-Frameworks vergleichbar mit JUnit gibt es für mehr als 60 Sprachen und sie können kostenlos heruntergeladen werden.²

¹ <http://www.pragmaticprogrammer.com/sk/ut/> bzw.
<http://www.pragmatischprogrammieren.de/>

² <http://www.xprogramming.com/software.htm>

Wir hoffen auch für die erfahreneren Programmierer, die schon mit Unit-Tests gearbeitet haben, ein paar angenehme Überraschungen parat zu haben. Überfliegen Sie die Grundlagen zu JUnit und konzentrieren Sie sich darauf, wie man über Tests nachdenkt, wie Testen das Design beeinflusst und wie man Herausforderungen angeht, die beim Testen in Teams entstehen.

Aber dieses Buch ist nur der Anfang. Es ist vielleicht Ihr erstes Buch zu Unit-Tests, aber hoffentlich nicht Ihr letztes.

Quelltexte

Im ganzen Buch finden Sie Beispiele in Java. Einige sind vollständige Programme, andere sind Programmfragmente. Wenn Sie eines der Beispiele ausführen wollen oder auch nur den vollständigen Quelltext sehen möchten (statt des abgedruckten Fragmentes), finden Sie am Seitenrand neben dem Fragment die entsprechenden Dateinamen.

Einige Quelltext-Teile werden im Laufe des Buches weiterentwickelt, Sie werden im Hauptverzeichnis und in Unterverzeichnissen mit den späteren Versionen (`rev1`, `rev2`, usw.) mehrere Dateien gleichen Namens finden.

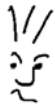
Alle Quelltexte in diesem Buch stehen auf der Seite zu *Pragmatic Unit Testing* auf unserer Webseite zum Herunterladen bereit.

Typographische Konventionen

<i>kursiv</i>	Hebt Begriffe hervor, die definiert oder aus einer anderen Sprache übernommen werden.
<code>computer</code>	Diese Schrift wird für Methodennamen, Datei- und Klassennamen und eine Menge anderer String-Literale verwendet.
<code>xx xx xx;</code>	Dieser Text steht für unwichtige Teile des Quelltextes, die bewusst weggelassen wurden.



Dieses Verkehrszeichen weist auf besonders anspruchsvolles Material hin, das man beim ersten Lesen getrost überspringen kann.



„Der Entwickler Joe“, unser gezeichneter Freund, stellt weitergehende Fragen und hilft Ihnen vielleicht so weiter.



Eine Pause im Text. Unterbrechen Sie das Lesen und durchdenken Sie die aufgeworfenen Fragen. Führen Sie ein kleines Experiment am Computer durch, bevor Sie weiterlesen.

Danksagung

Wir möchten uns ganz besonders bei den folgenden Praktikern für ihre wertvollen Zuarbeiten, Empfehlungen und Erfahrungsberichte bedanken: Mitch Amiano, Nascif Abousalh-Neto, Andrew C. Oliver, Jared Richardson und Bobby Woolf.

Wir danken auch unseren Korrekturlesern, die Zeit und Energie investiert haben, um unsere Fehler und Versäumnisse zu finden und um den manchmal zu komplizierten Schreibstil zu entwirren: Will Gwaltney, Sharee L. Johnson, Eric Kalendra, Al Koscielny, James J. O'Connor III, Mike Stok, Drew Thompson und Eric Vought.

Danke für Eure Unterstützung und die viele Arbeit, die Ihr Euch gemacht habt.

Andy Hunt und Dave Thomas

September, 2003

`pragprog@pragmaticprogrammer.com`