## Contents

About the Starter Kit .............................................. x

Preface ....................................................................... xii

1 Introduction .......................................................... 1
   1.1 Coding with Confidence .................................. 2
   1.2 What Is Unit Testing? .................................... 3
   1.3 Why Should We Bother with Unit Testing? .......... 4
   1.4 What Do We Want to Accomplish? .................... 5
   1.5 How Do We Do Unit Testing? ......................... 7
   1.6 Excuses for Not Testing ................................. 9
   1.7 Road Map .................................................. 14

2 Your First Unit Tests ............................................... 15
   2.1 Planning Tests ............................................ 16
   2.2 Testing a Simple Method .............................. 18
   2.3 Running Tests with NUnit ............................ 19
   2.4 Running the Example ................................. 26
   2.5 More Tests ............................................. 30

3 Writing Tests in NUnit .......................................... 31
   3.1 Structuring Unit Tests ................................. 31
   3.2 Classic Asserts .......................................... 33
   3.3 Constraint-Based Asserts ............................. 36
   3.4 NUnit Framework ....................................... 41
   3.5 NUnit Test Selection .................................. 43
   3.6 More NUnit Asserts ................................... 52
   3.7 NUnit Custom Asserts ................................. 54
   3.8 NUnit and Exceptions ................................. 56
   3.9 Temporarily Ignoring Tests ......................... 58
4  **What to Test: The Right BICEP**  61
   4.1  Are the Results Right?  62
   4.2  Boundary Conditions  65
   4.3  Check Inverse Relationships  67
   4.4  Cross-Check Using Other Means  68
   4.5  Force Error Conditions  69
   4.6  Performance Characteristics  70

5  **CORRECT Boundary Conditions**  72
   5.1  Conformance  73
   5.2  Ordering  75
   5.3  Range  76
   5.4  Reference  81
   5.5  Existence  82
   5.6  Cardinality  83
   5.7  Time  85
   5.8  Try It Yourself  87

6  **Using Mock Objects**  92
   6.1  Stubs  94
   6.2  Fakes  96
   6.3  Mock Objects  103
   6.4  When Not to Mock  116

7  **Properties of Good Tests**  121
   7.1  Automatic  122
   7.2  Thorough  124
   7.3  Repeatable  125
   7.4  Independent  127
   7.5  Professional  128
   7.6  Testing the Tests  130

8  **Testing on a Project**  134
   8.1  Where to Put Test Code  134
   8.2  Where to Put NUnit  137
   8.3  Test Courtesy  139
   8.4  Test Frequency  140
   8.5  Tests and Legacy Code  142
   8.6  Tests and Code Reviews  145
## 9 Design Issues

9.1 Designing for Testability .......................... 149
9.2 Refactoring for Testing ......................... 152
9.3 Testing the Class Invariant ..................... 165
9.4 Test-Driven Design ................................. 168
9.5 Testing Invalid Parameters ....................... 170

## 10 UI Testing

10.1 Unit Testing WinForms ......................... 172
10.2 Unit Testing Beyond Windows Forms .......... 176
10.3 Web UIs ........................................ 179
10.4 Programmer UIs .................................. 184
10.5 Command-Line UIs ............................... 185
10.6 GUI Testing Gotchas .............................. 186

## A Gotchas

A.1 As Long As the Code Works .................... 189
A.2 Smoke Tests ..................................... 189
A.3 “Works on My Machine” .......................... 190
A.4 Floating-Point Problems ....................... 190
A.5 Tests Take Too Long .............................. 191
A.6 Tests Keep Breaking ............................. 192
A.7 Tests Fail on Some Machines ................... 192
A.8 Tests Pass in One Test Runner, Not the Other 193
A.9 Thread State Issues ............................... 194
A.10 C# 2.0–Specific Issues ......................... 194

## B Resources

B.1 On the Web ..................................... 196
B.2 Bibliography ..................................... 199

## C Summary: Pragmatic Unit Testing


## D Answers to Exercises


About the Starter Kit

Our first book, *The Pragmatic Programmer: From Journeyman to Master*, is a widely acclaimed overview of practical topics in modern software development. Since it was first published in 1999, many people have asked us about follow-on books, or sequels. Toward that end, we started our own publishing company, the Pragmatic Bookshelf. By now we have dozens of titles in print and in development and have received major awards and five-star reviews.

But the very first books we published are still some of the most important ones. Before embarking on any sequels to *The Pragmatic Programmer*, we thought we’d go back and offer a prequel of sorts.

Over the years, we’ve found that many of our pragmatic readers who are just starting out need a helping hand to get their development infrastructure in place so they can begin forming good habits early. Many of our more advanced pragmatic readers understand these topics thoroughly but need help convincing and educating the rest of their team or organization. We think we have got something that can help.

The Pragmatic Starter Kit is a three-volume set that covers the essential basics for modern software development. These volumes include the practices, tools, and philosophies that you need to get a team up and running and productive. Armed with this knowledge, you and your team can adopt good habits easily and enjoy the safety and comfort of a well-established “safety net” for your project.

Volume I, *Pragmatic Version Control*, describes how to use version control as the cornerstone of a project.
A project without version control is like a word processor without an Undo button: the more text you enter, the more expensive a mistake will be. Pragmatic Version Control shows you how to use version control systems effectively, with all the benefits and safety but without crippling bureaucracy or lengthy, tedious procedures.

This volume, Pragmatic Unit Testing, is the second volume in the series. Unit testing is an essential technique because it provides real-world, real-time feedback for developers as we write code. Many developers misunderstand unit testing and don’t realize that it makes our jobs as developers easier. This volume is available in two different language versions: in Java with JUnit and in C# with NUnit.

Volume III, Pragmatic Automation, covers the essential practices and technologies needed to automate your code’s build, test, and release procedures. Few projects suffer from having too much time on their hands, so Pragmatic Automation will show you how to get the computer to do more of the mundane tasks by itself, freeing you to concentrate on the more interesting—and difficult—challenges.

These books were created in the same approachable style as our first book, and they address specific needs and problems that you face in the trenches every day. But these aren’t dummy-level books that give you only part of the picture; they’ll give you enough understanding that you’ll be able to invent your own solutions to the novel problems you face that we haven’t addressed specifically.

For up-to-date information on these and other books, as well as related pragmatic resources for developers and managers, please visit us on the Web at the following address:

http://www.pragmaticprogrammer.com

Thanks, and remember to make it fun!
Pragmatic Starter Kit

**Version control. Unit Testing. Project Automation.** Three great titles, one objective. To get you up to speed with the essentials for successful project development. Keep your source under control, your bugs in check, and your process repeatable with these three concise, readable books from The Pragmatic Bookshelf.

Visit Us Online

**Unit Testing in C# Home Page**
http://pragmaticprogrammer.com/titles/utc2
Source code from this book, errata, and other resources. Come give us feedback, too!

**Register for Updates**
http://pragmaticprogrammer.com/updates
Be notified when updates and new books become available.

**Join the Community**
http://pragmaticprogrammer.com/community
Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

**New and Noteworthy**
http://pragmaticprogrammer.com/news
Check out the latest pragmatic developments in the news.

Buy the Book

If you liked this PDF, perhaps you’d like to have a paper copy of the book. It's available for purchase at our store: http://pragmaticprogrammer.com/titles/utc2.

Contact Us

Phone Orders: 1-800-699-PROG (+1 919 847 3884)
Online Orders: www.pragmaticprogrammer.com/catalog
Customer Service: orders@pragmaticprogrammer.com
Non-English Versions: translations@pragmaticprogrammer.com
Pragmatic Teaching: academic@pragmaticprogrammer.com
Author Proposals: proposals@pragmaticprogrammer.com