

# Foreword

If you go to [junit.org](http://junit.org), you'll see a quote from me: "Never in the field of software development have so many owed so much to so few lines of code." JUnit has been criticized as a minor thing, something any reasonable programmer could produce in a weekend. This is true, but utterly misses the point. The reason JUnit is important, and deserves the Churchillian knock-off, is that the presence of this tiny tool has been essential to a fundamental shift for many programmers: Testing has moved to a front and central part of programming. People have advocated it before, but JUnit made it happen more than anything else.

It's more than just JUnit, of course. Ports of JUnit have been written for lots of programming languages. This loose family of tools, often referred to as xUnit tools, has spread far beyond its java roots. (And of course the roots weren't really in Java—Kent Beck wrote this code for Smalltalk years before.)

xUnit tools, and more importantly their philosophy, offer up huge opportunities to programming teams—the opportunity to write powerful regression test suites that enable teams to make drastic changes to a code-base with far less risk; the opportunity to re-think the design process with Test Driven Development.

But with these opportunities come new problems and new techniques. Like any tool, the xUnit family can be used well or badly. Thoughtful people have figured out various ways to use xUnit, to organize the tests and data effectively. Like the early days of objects, much of the knowledge to really use the tools is hidden in the heads of its skilled users. Without this hidden knowledge you really can't reap the full benefits.

It was nearly twenty years ago when people in the object-oriented community realized this problem for objects and began to formulate an answer. The answer was to describe their hidden knowledge in the form of patterns. Gerard Meszaros was one of the pioneers in doing this. When I first started exploring patterns, Gerard was one of the leaders that I learned from. Like many in the patterns world, Gerard also was an early adopter of eXtreme Programming, and thus worked with xUnit tools from the earliest days. So it's entirely logical that he should have taken on the task of capturing that expert knowledge in the form of patterns.

I've been excited by this project since I first heard about it. (I had to launch a commando raid to steal this book from Bob Martin because I wanted it to

## FOREWORD

grace my series instead.) Like any good patterns book it provides knowledge to new people in the field, and just as important, provides the vocabulary and foundations for experienced practitioners to pass their knowledge on to their colleagues. For many people, the famous Gang of Four book *Design Patterns* unlocked the hidden gems of object-oriented design. This book does the same for xUnit.

Martin Fowler  
Series Editor  
Chief Scientist, ThoughtWorks