



Preface

SCIENCE IS FOUNDED on the principle of creating experiments that give the same results each time they are performed. Unfortunately, a software development project isn't like a scientific experiment because the outcome is always different. Even teams that use the same tools and process will still produce different solutions to the same task, each unique in terms of code set, bugs, performance, and so forth. This variability arises because the results of software development depend upon individuals and their interactions as much as the process and tools they employ.

The idea that the outcome of a software project is largely dependent upon people and the way they work together caused Kent Beck to observe the habits of successful teams and then put them into a framework of values and practices, which he called Extreme Programming (XP). This provided an alternative to the decades-old notion that the only way to impose order upon software development was to apply expensive tools and a well-prescribed process. XP joined a number of similar lightweight approaches to software development, collectively known as Agile, which shared the common aim of satisfying customers through the early and continuous delivery of valuable software. Over the past five years, this Agile movement has grown to become a significant driver of change in our industry.

Agile seems to have successfully captured the middle ground of software development methodologies. Teams with too little process to guide them have found that embracing Agile allows them to make significant improvements in the outcome of their projects, without creating the sort of

bloated bureaucracy they fear. Teams with too much process have found that adopting an Agile approach has made them much more productive and responsive, but without their projects descending into the sort of chaotic hacking they fear. Thousands of projects have been run along Agile lines. They haven't all succeeded, but this is to be expected because any worthwhile software project involves a degree of risk. However, plenty of these projects have produced spectacular results, and once people have tried Agile, they seldom want to return to their old ways of doing things. We suspect this is simply because most people find it, as we do, to be a more pleasurable and rewarding way to develop software.

■ NOTE

This book is primarily based on the values and practices of Extreme Programming as described in Kent Beck's book, *Extreme Programming Explained*.¹ We apply them in the context of a five-developer team using Microsoft's Visual Studio 2005 Team System.

Who Should Read This Book?

This is a book for people on real teams who are transitioning to Microsoft's Visual Studio Team System (VSTS), but who might not yet be ready to fully embrace a process such as MSF for Agile Software Development. It is written for people who want an easy way to gain value from the tools and at the same time lay the foundations for future process improvement. We envision our readers to include the following:

- **People new to software development**—Teaches you how to use VSTS and gives you the core skills you need in order to work effectively on an Agile team. There are few assumptions about your

1. [XPE2] Beck, Kent, with Cynthia Andres. *Extreme Programming Explained, Second Edition* (Addison-Wesley, 2005).

technical background, but some knowledge of using Visual Studio will help when completing the exercises.

- **Experienced developers**—Puts what you already know into the context of an Agile project and explains how to make good use of the new tools provided by VSTS. People who are encountering Microsoft technology for the first time should find the exercises and glossary particularly useful.
- **Architects**—Explains the new VSTS tools for software architects, but its real value lies in helping you to adapt your skills so that you can add value to an Agile team.
- **Testers**—Helps you understand the expanded role of testers on an Agile team and explains how to use the basic VSTS tools needed to test software in this new Software Project Environment.
- **Business analysts and customers**—Explains how an Agile approach can give your business a better return on investment. You'll also learn how an Agile team works to make sure you get the software you want, when you need it.
- **Project managers**—Describes how to transition your people onto a small Agile team so that they can deliver better-quality software, in less time and for less cost. In addition, you'll discover how VSTS gathers information about a project into one place to make the running of the project more transparent.
- **Software entrepreneurs**—Provides you with a road map for setting up a small, top-performing software team. It reveals the key technical and people issues you need to address through a series of anecdotes and comments gleaned from the decades we've spent working in the industry.

This book is not about process improvement applied from the top of an organization downward, it's about empowering teams to change things for themselves from the bottom up.

Tools Needed

In order to follow the exercises in this book, you will need access to an existing installation of VSTS or have the ability to install Visual Studio Team Suite in one of the following environments:

- Desktop PC able to host the Microsoft Virtual PC
- Single-server PC running Windows Server 2003 (SP2 or R2)
- Network comprising a server and several desktop PCs

You will be glad to hear that Visual Studio Team Suite is freely available from Microsoft's technical Web site² as a trial edition (full functionality, but expires after 180 days) as well as for purchase from your usual Microsoft reseller. In addition, MSDN subscribers can obtain Team System VPC, which is a "ready to run" virtual machine image of VSTS for use with the freely available Microsoft Virtual PC. Appendix A covers how to set up VSTS in all of these environments.

NOTE

Framework for Integrated Test (FIT) is required for Section 7, but it is freely available from the C2 Web site.³ InstallShield and Installation Collaboration are needed for the exercises in Chapter 29, but free evaluation editions are available on the Macrovision Web site.⁴

Structure of the Book

The book's Introduction contains a story about a fictional software team called OSPACS that has a broken process; the team always delivers late, has high staff turnover, and is surprised to discover that its software is full of

2. Microsoft's Web site for Visual Studio Team System (<http://msdn.microsoft.com/teamsystem>).

3. Ward Cunningham's C2 Web site for FIT (<http://fit.c2.com>).

4. Macrovision's Web site for InstallShield (www.macrovision.com/downloads).

bugs and has gone three times over budget. The rest of the book is about how the team fixed these problems, but along the way we aim to give you insight into the use of VSTS and the meaning of better software development for a small Agile team.

The main body of the book is divided into ten sections, each concerned with a specific aspect of software development as practiced by Agile teams. These sections are ordered into a sequence that helps build up a team's proficiency in a step-by-step manner. For example, we don't present information about project planning until we've covered material such as testing and Team Build because clearly your team's plans will not be very reliable until you can dependably deliver quality software. However, with that said, each section is largely self-contained, so you can read them in any order that makes sense to you. Indeed, we expect this to happen as each reader will have different priorities for things they want to learn about.

Each section starts with a short story and ends with a review describing how the OSPACS team put the ideas into practice, the team's impressions about the material, and its relationship to a set of Agile values. In this way, we provide you with some light relief from the technical stuff while presenting another perspective on the subject matter that might help you apply it on your own team. Within each section, the chapters usually start by explaining some basic concepts and then put them in a practical context by giving you a series of exercises to follow using the tools provided by VSTS. You will also find sidebars in various chapters that summarize particular XP practices relevant to what is being discussed. In this way, theory and practice are put together into something that is hopefully reasonably entertaining and interesting to read.

 NOTE

At the back of the book is information about relevant resources, a glossary, a bibliography, and a number of appendixes, as well as a list of the XP practices and a complete list of all the exercises.

Conventions

The XP practices listed on the inside cover are described in appropriate places throughout the book as sidebars which are given a different font and layout to distinguish them from the main body of the text. In addition to normal printing conventions, the following special conventions are also used in the book:

TIP

Best practice or suggestion.

WARNING

Issue that requires particular care or consideration.

NOTE

Item of particular interest.

[XPE2]	Reference to an item in the bibliography.
File New Team Project	Shorthand for “select the menu item New from the File menu and then select its Team Project submenu item.”
Right-click Delete	Shorthand for “choose Delete from the selected item’s context menu.”
... the Agile team	Words that are capitalized are used in a specific sense, so here we mean a team that shares the values of the <i>Agile</i> software movement.

 NOTE

James Newkirk and Will Stott collaborated in the production of this book, but as Will did most of the writing, it's his voice you hear when you read something such as "I did so and so" or "We did this and that."

About the Book's Web Site

We have created a Web site for this book that contains most of the code created for the exercises, information about any errors in the text found after publication, and other supplementary material which we feel might be useful to readers:

www.BetterSoftwareDevelopment.org

We strongly encourage people to visit this site for the latest information about both VSTS and Agile software development. We would be delighted to receive feedback from readers and will try to respond to you as promptly as our other work commitments allow.