

# Preface

---

## *Why We Wrote This Book*

We were early adopters of Extreme Programming, testing on XP teams that weren't at all sure where testers and testing fit in. At the time, there wasn't much in the agile (which wasn't called agile yet) literature about acceptance testing, or how professional testers might contribute to teams. We learned from our own experiences, as well as from others in the agile community. As others helped us, we started helping other teams and testers to be successful at agile testing. In 2002, Lisa co-wrote [Testing Extreme Programming](#) with Tip House, with lots of help from Janet. In the intervening years, agile development has evolved, and the agile testing community has flourished. With so many people contributing ideas, we've learned a whole lot more about agile testing.

Individually and together, we've helped teams transition to agile, helped testers learn how to contribute on agile teams, and worked with others in the agile community to explore ways agile teams can be more successful at testing. Our experiences differ, as Lisa as spent most of her time as an agile tester with web apps in the retail, telephony, and finance industries for years at a time with stable teams. Janet has worked with software organizations developing enterprise systems in a variety of industries. These agile projects have included developing a message handing system, an environmental tracking system, a remote data management system (included an embedded application, with a communication network as well as the application), an oil and gas production accounting application, and applications in the airline transportation industry. She has played different roles, sometimes tester, sometimes coach, but always working to better integrate the test team with the rest of the team. She has been with teams from as short as six months to as long as 1.5 years.

With these different points of view, we have learned to work together and complement each other over the years giving presentations and tutorials together.

Several excellent books oriented towards agile development on testing and test patterns have been published (see our bibliography). These books deal mainly with tests that are focused on tests that help the developer. We decided to write a book aimed at helping agile teams be more successful at delivering business value using tests that the business can understand. We want to help testers and quality assurance professionals who have worked in more traditional development methodologies make the transition to agile development. We've worked with teams of all sizes. We've taken ideas from other agile practitioners and figured out how to apply them on a practical, day-to-day level. We've put our experiences into this book to help testers, quality assurance managers, developers, development managers, product owners, and everyone else with a stake in effective testing on agile projects. We've focused on the role of the tester on an agile team, a role which may be adopted by a variety of professionals.

Agile testing practices aren't limited to members of agile teams. We've used many of these practices to improve testing on projects using traditional development methodologies. This book is also intended to help testers on any type of project to get the software their customers need.

Agile development isn't the only way to successfully deliver software. However, the successful teams we've been on, agile or waterfall, have several critical commonalities. The programmers write and automate unit and integration tests that provide good code coverage. They are disciplined in the use of source code control and code integration. Skilled testers are involved from the start of the development cycle, and are given time and resources to do an

adequate job of all necessary forms of testing. An automated regression suite that covers the system functionality at a higher level is run and checked regularly. The development team understands the customers' jobs and their needs, and works closely together with the business experts.

People, not methodologies or tools, make projects successful. We enjoy agile development because its values, principles and core practices allow people to do their best work, and testing and quality are central to agile development. In this book, we hope to explain how to apply agile values and principles to your unique testing situation and enable your teams to succeed. More about that in Chapter 1 "What is this Agile Testing, Anyway?" and in Chapter 2, "The Ten Principles of an Agile Tester".

---

## *Our Audience*

If you've ever asked any of the following excellent questions which we've heard many times, this book will help you.

- [lb] If developers are writing tests, what do the testers do?
- [lb] I'm a QA manager, and our company is implementing agile development (Scrum, XP, DSDM, name your flavor). What's my role now?
- [lb] I've worked as a tester on a traditional waterfall team, and I'm really excited by what I've read about agile. What do I need to know to work on an agile team?
- [lb] What's an "agile tester"?
- [lb] I'm a developer on an agile team. We're writing code test-first, but our customers still aren't happy with what we deliver. What are we missing?
- [lb] I'm a developer on an agile team. We're writing our code test-first. We make sure we have tests for all our code. Why do we need testers?
- [lb] I coach an agile development team. Our QA team can't keep up with us, testing always lags behind. Should we just plan to test an iteration behind development?
- [lb] I'm a software development manager. We recently transitioned to agile, but all our testers quit. Why?
- [lb] I'm a tester on a team that's going agile. I don't have any programming or automation skills. Is there any place for me on an agile team?
- [lb] How can testing possibly keep up with two-week iterations?
- [lb] What about load testing, performance testing, usability testing, all the other 'ilities'? Where do these fit in?
- [lb] We have audit requirements. How does agile development and testing address these?

If you have similar questions, and you're looking for practical advice about how testers contribute to agile teams and how agile teams can do an effective job of testing, you've picked up the right book.

There are many "flavors" of agile development, but they share much in common. This book follows the Agile Manifesto, as we explain in Chapter 1. Whether you're practicing Scrum, Extreme Programming, Crystal, DSDM or your own variation of agile development, you'll find information here to help with your testing efforts.

When Robin Dymond, a managing consultant and trainer we know who has helped many teams adopt lean and agile, heard we were writing this book, he sent us the user story he'd like

to have fulfilled. It encapsulates many of the requirements we hoped to deliver.

---

## ***A User Story for an Agile Testing Book***

My main desire, as expressed to me by two QA people to me today, is to have the following user story fulfilled, hopefully from your book.

Book Story 1
As a QA professional, I can understand the main difference between
Traditional QA and and agile team member with QA background in one
Hour, so that I can begin internalizing my new responsibilities and deliver
Value to the customer sooner and with less difficulty.

Acceptance conditions:

- [lb] My concerns and fears about losing control of testing are addressed
- [lb] My concerns and fears about having to write code (never done it) are addressed
- [lb] The Agile coach is satisfied that the team-based perspective is clear
- [lb] As a tester I understand my new value to the team
- [lb] As a new to Agile tester I can easily read about things that are most important to my new role
- [lb] As a new to Agile tester I can easily ignore things that are less important to my new role
- [lb] As a new to Agile tester I can easily get further detail about agile testing that is important to MY context

Were I to suggest a solution to this problem, I think of Scrum vs XP. With Scrum you get a simple view that allows people to quickly adopt Agile. However, Scrum is the tip of the iceberg for successful Agile teams. For testers who are new, I would love to see Agile testing ideas expressed in layers of detail. What do I need to know today, what should I know tomorrow, and what context sensitive things should I consider for continuous improvement?

-- Robin Dymond

---

---

## ***How to Use This Book***

If you aren't sure where to start in this book, or you just want a quick overview, we suggest you read the last chapter, Chapter 25: Key Success Factors, and follow wherever it leads you.

## **Part 1: Introduction**

If you want quick answers to questions such as “Is agile testing different than testing on waterfall projects?” and “What’s the difference between a tester on a traditional team and an agile tester?”, start with Part 1. This is the “tip of the iceberg” that Robin requested. It includes an overview of how agile differs from a traditional phased approach and explores the “whole team” approach to quality and testing. We define the “agile testing mindset”. We explain how testers apply agile values and principles to contribute their particular expertise.

## **Part 2: Organizational Challenges**

If you’re a tester or manager on a traditional QA team, or you’re coaching a team that’s moving to agile, Part 2 will help you with the organizational challenges faced by teams in transition. The “whole team” attitude represents a lot of cultural changes, but it helps overcome the fear testers have when they wonder how much control they’ll have or whether they’ll be expected to write code.

When is a story “done”? How can we engage the QA team? What about management’s expectations? How should we structure our agile team, where do the testers fit? What do we look for when hiring an agile tester? How do we cope with a team distributed across the globe? These are some of the questions addressed in Part 2.

Part 2 also looks at some topics we don’t always enjoy talking about. Metrics and how they’re applied can be a controversial issue, but there are positive ways to use them to benefit the team. Defect tracking easily becomes a point of contention for teams with questions such as “Do we use a defect tracking system?” or “When do we log bugs?”

Two common questions about agile testing from people with traditional test team experience are: “What about test plans?”, and “Is it true there’s no documentation on agile projects?” Part 2 clears up these mysteries.

## **Part 3: Using the Agile Testing Quadrants**

Want more details on what types of testing are done on agile projects? Are you wondering who does what testing? How do you know if you’ve done all the testing that’s needed? How do you decide what practices, techniques and tools fit your particular situation? If these are your concerns, check out Part 3.

We use Brian Marick’s Agile Testing Quadrants to explain the purpose of testing. The quadrants help you define all the different areas your testing should address, from unit level to reliability and other “ilities” and everything in between. This is where we get down into the nitty-gritty of how to deliver a high quality product. We explain techniques to help you communicate well with your customers, and understand their requirements. This section shows how tests drive development at multiple levels. Part 3 also provides toolkits to help you effectively define, design and use business-facing tests to support the team and critique the product.

## **Part 4: Test Automation**

Test automation is a central focus of successful agile teams, and it’s a scary topic for lots of people (we know, because it’s had us running scared before!) How do you squeeze test automation into short iterations and still get all the stories completed?

Part 4 gets into the details of when and why to automate, how to overcome barriers to test automation, how to overcome fears about automation, and how to develop and implement a test automation strategy that works for your team. Since test automation tools change and evolve so

rapidly, our aim is not to explain how to use specific tools, but to help you select and use the right tools for your situation. Our agile test automation tips will help you with difficult challenges such as testing legacy code.

### **Part 5: An Iteration in the Life**

If you just want to get a feel for what testers do throughout the agile development cycle, or you need help putting together all the information in this book, go to Part 5. Here we chronicle an iteration and more, in the life of an agile tester. Testers contribute enormous value throughout the agile software development cycles. Part 5 demonstrates in detail what testers do on a daily basis, starting with planning releases and iterations, getting each iteration off to a good start, collaborating with the customer and development teams to test and write code, delivering each release, and finding ways for the team to improve its process.

### **Part 6: Summary**

We chose seven key factors agile teams can use for successful testing. If you're having trouble deciding where to start with agile testing, or how to work on improving what you're doing now, these success factors will give you some direction.

Agile development is all about doing your best work. Every team has unique challenges. We've tried to present all the information that we think may help agile testers, their teams, their managers and their customers. Choose and apply the techniques that you think are appropriate for your situation. Experiment constantly, evaluate the results, and come back to this book to see what might help you improve. Our goal is to help testers and agile teams enjoy delivering the best and most valuable product they can.