Preface

There were two motivations for writing this book. First, I am continually trying to organize the flood of information that comes at me from all directions to make sense of it all. In the vain hope that there is always a better way to organize ideas, I find myself grouping ideas together and continually looking for patterns in the way these groups relate to each other. It's a bit like indexing, and it helps me to remember stuff. When I encounter a new problem and a solution is required, I can retrieve the relevant information quickly, even if I learned it years ago and never used it in the same way before. In deciding to write this book, I wanted to provide an organized collection of methods, techniques, and tools for testing e-business systems. The organization of these methods should make it easy for someone embarking on a new project to make some sensible decisions about what to do. I wanted the book to work for high-adrenaline, high-pressure, fast-paced projects, but also for more comfortable, well-resourced, highly organized projects (although all projects involve adrenaline, pressure, and a fast pace at the end).

The second motivation for writing this book was my dissatisfaction with the use of risk as a driver for testing. For as long as I can remember, testing experts and practitioners have talked about risk, risk-based testing, and how testing addresses risk. I tried to understand risk-based testing better, but when I looked for deeper insight into risk and testing, there seemed to be little hard information to go on. The usual mantra trotted out states that risks are bad; testing addresses risk, so we should test to find as many bugs as we can to mitigate that risk. When I speak to testers and ask what a risk is, I have usually gotten a description of various types of bugs. Now, I'm not saying there's anything wrong with the notion that risks can be types of potential bugs and that testers should focus on these in their testing. The main goal of a tester is to expose faults in software, and I've been preaching that to testing classes for 10 years now. This simple idea is fine when it comes to designing some test cases, but I don't think it is a useful strategy for planning the test effort in a large project or in a project with extreme time pressures or without requirements or, well, any project I can think of.

The discipline commonly called *risk management* defines risk subtly and less tangibly. I've read a few books on risk now, and the relationship between risk management and testing is a little tenuous too. Most risk-management methods treat risks at too high a level to be useful for testers. This is no surprise. How many types of bugs could you think of? Thousands before lunchtime, probably! There is a gap between what management thinks a risk is and what testers do. There is a second problem, however; testers can't use the risk-management discipline as their managers do. Risk management doesn't scale up to larger volumes. What does risk-based testing mean if there is such a disconnect between management and test practitioners? Risk-based testing sounds good, but what does it mean to a manager who is doing risk-based project management?

The risk-based test methodology set out in the initial chapters of this book attempts to use early risk analysis to connect the concerns and objectives of senior management with the test process and activities of the project's testers. The consequences of so doing are obvious; most of them are beneficial. If the test process is developed specifically to address the risks of concern to stakeholders and management, we get the following results:

- Management is better able to view the testing and so has more control over it.
- The budget for testing is likely to be more realistic.
- The testers have a clearer understanding of what they should be doing.
- The budget allocated to testing is determined by consensus, so everyone buys into the quantity of testing planned.
- The information-provision role of testers is promoted, but fault detection is as important as ever.
- The information provided by testers increases their influence on management decision making.
- Release decision making is better informed.

The risk-based methodology integrates nicely with an inventory of product risks, grouped by the specific test techniques that address them. The product risks detailed in Chapters 9 through 15 can be taken together and used as a candidate risk register to get a project risk analysis off the ground. By combining a methodology for creating a test process with a library of risks, each having associated test techniques, test managers have a ready-made handbook for building a master test plan that is comprehensive without being too prescriptive.

The risk-based test method is universal. Risks pervade all software projects, and risk taking is inevitable. This book attempts to help identify a project's relevant risks and focus the attention of testers onto the risks of most concern. Testing aims to mitigate risk by finding faults, but it also aims to reduce uncertainty about risk by providing better risk information to management, enabling management to better steer their projects away from hazards and make better decisions.

Audience

This book is aimed at test managers and people who test. It should be useful to developers and testers working on large, well-organized projects, as well as those working on projects so small (or chaotic) that no specific testing or test management role has been defined. The test techniques chapters should be informative to people who are new to e-business and the Internet and those who have specialized in one testing area. Project managers interested in implementing a risk-based approach to testing should find the early chapters on risk and the later chapters on making it happen of use. The book should also be of interest to students wishing to understand how test planning, design, and execution activities can be integrated with project risk management.

Structure

Part I Chapters 1 through 4 provide an introduction to risk-based testing and the relationship between risk management, risks, and the testing process. Testers are encouraged to use the language of risk to articulate their missions and negotiate test budgets with management and stakeholders. A methodology for creating a test process based on a consensus view of product risk is presented.

Part II Chapters 5 through 8 set out the types of failure that e-business systems are prone to and the threat those failures present to these systems. Of course, a test strategy needs to take into account many factors other than risks, and these other factors are described. The test types and techniques are organized into a test process framework, together with the other considerations relevant to formulating a test process.

Part III After a short introduction, Chapters 9 through 15 present 24 techniques for testing e-business systems. Each chapter summarizes the risks covered by the techniques, then presents the techniques in some detail, and ends with a list of tools and references for further reading. Chapter 16 covers postdeployment monitoring, and Chapter 17 provides a summary of the tool types that are useful to the e-business tester, together with some guidelines for their selection and use.

Chapters 9 through 15 describe the e-business test techniques in more detail. The test techniques are grouped under the test type headings identified in Table 7.1, the Test Process Framework. There are two further chapters added: Chapter 16 covers postdeployment monitoring and Chapter 17 provides a summary of the tool types that are useful to the e-business tester with some guidelines for the use of proprietary tools. Appendix B provides some ideas for simple home-brew tools that might help in your functional testing.

It should be noted for the majority of techniques described here, the description of the technique could not be comprehensive. Whole books have been written about security audit, performance evaluation, and object-oriented testing and usability, for example. This book aims to provide an overview of all of the broad categories of risk with enough detail to give test managers and testers sufficient information (including references to other books) to decide on the scope of testing and what pitfalls lay ahead. These chapters are intended to give you an insight into the essential characteristics of the test techniques, why you might select them for inclusion in your test strategy and how to structure them into an effective test process. Each chapter provides references for further reading on the techniques or tools that support them.

On the subject of tools, most chapters provide examples of the use of tools or remote testing services. We are not promoting specific tools, but are using the tools that we have personal experience of to illustrate the typical way that such tools are used. Chapter 17 provides an overview of the types of tools and sources of tool information. One further point to note is that many tools have features that support multiple test types (as we have defined them). We suggest you mix and match tools to suit your specific concerns, work practices, development technologies, and budgets. We haven't yet found a perfect combination of tools and we doubt whether one exists, as all projects have their different needs. Where a tool takes a novel or innovative approach to a particular aspect of the testing task in hand, we may highlight it if it offers a particular benefit to testers.

The chapters may refer occasionally to a specific technology such as Active Server Pages, Java, or VBScript for example. There are many development products on the market and in the time taken to publish this book, no doubt there will be many more. We have not attempted to keep pace with such developments. Some technologies such as XML for example, open up new possibilities for business-to-business systems and their users. Wireless Application Protocol (WAP) phones, PDAs, and other mobile devices will soon make the Web all-pervasive. The risk implications of these technologies have not yet been fully established. The testing of these technologies requires more experience to be gained. Perhaps a revision to this book will include them someday. In the meantime, testers will have to innovate to address these emerging risks. It is hoped that the risk-based approach, the test process framework, and the outline test techniques described in this book will provide enough support and guidance for you to create your own test techniques for specific types of risks.

Each chapter begins with a summary of the risks that the chapter's test techniques address. For each test technique described in the chapter, the potential causes of the problems are discussed and the method described. If there are alternative methods or there are tools that have different approaches, these are described and illustrated where appropriate.

One of the challenges of e-Business testing is: where does subsystem (developer) testing stop and system level testing begin? Even though you might be a system or acceptance tester and not particularly interested in what developers do to test, we encourage you to look into the "more technical" techniques further. Allocation of responsibility for more and less technical techniques is a major consideration for your test strategy. You can only make these decisions by knowing what kinds of testing can be done and who is best placed to perform it. If you do not liaise with earlier testers or developers, you may find that some of your tests may duplicate tests performed by others. Even worse, you may find that if you assume others do some types of tests, they may not be done at all.

Each chapter ends with a listing of tools and references for further reading.

The risk listings that appear at the start of each of Chapters 9 through 15 are not ordered or prioritized, but are grouped by the test techniques that we proposed are most appropriate to addressing them. Some of the risks listed would have dramatically different consequences but we have not prejudged any of the probability or consequence scores that you might assign in your own risk workshops. The groups of risks are in the same order as the test techniques are described in the chapters. Some risks could be addressed by more than one technique. For example, in Chapter 9, confusing text on Web pages could be checked using spell and grammar checking tools or as part of a usability assessment (Chapter 13). You must decide which technique(s) you use to address these specific tests. The risk and test objective tables are a guideline.

When the W-Model is described in Chapter 4, we suggest that any and all deliverables from the development activities could be tested. Requirements, specifications, designs, and code itself can all be subjected to some form of review or inspection process.

We haven't included reviews as Web testing techniques because these techniques are not specific to Web-based systems or affected by the technology used to build them. Many of the risks associated with requirements, designs, and specifications can be addressed at least in part by review activities. When considering the behavior-related failures that could occur, consider whether these failures could be traced back to failings in earlier documentation and whether a review could expose shortcomings in these documents. Of course, you will probably conduct unit, integration, and system tests on the built software but early reviews are effective at finding the most potentially expensive faults much earlier and cheaply than later dynamic testing. References [1, 2] provide a good introduction to inspections and review techniques.

In Chapters 9 through 15, we describe test techniques that are either Web specific, or are influenced by the fact that Web technology is being used to build the software under test. In effect, all of the dynamic test techniques (and the static analysis of Web pages) are affected to greater or lesser degrees and because of this, they were described in the techniques chapters.

Part IV Chapters 18 through 20 address some of the practicalities of "making it happen." Chapter 18 discusses the influence of different development methodologies and technologies on the test process. Chapter 19 presents insights into the test-planning and organizational challenges presented by e-business projects. Chapter 20 covers the particular difficulties of test execution and making the release decision.

Appendixes Appendix A outlines the essential technical knowledge that e-business testers should have. Appendix B sets out some of the opportunities for building your own test tools and presents some examples of how they might be used.

Finally, a list of acronyms and abbreviations is included, followed by a comprehensive glossary of terms.

The Web Site

A Web site has been created at http://www.riskbasedtesting.com to accompany this book. The site not only promotes the book (you can order copies on-line, of course), but also contains downloadable material extracted from the book, including document templates and a candidate risk inventory. Please take the time to visit.

Needless to say, any errors in this book are my responsibility. If you find any, please e-mail me at paulg@evolutif.co.uk, and I will post errors and their corrections on the Web site.

References

[1] Freedman, D. P., and G. M. Weinberg, *Handbook of Walkthroughs, Inspections and Technical Reviews*, NY: Dorset House Publishing, 1990.

[2] Gilb, T., and D. Graham, *Software Inspection*, Wokingham, U.K.: Addison Wesley, 1993.