TESTING IT: AN OFF-THE-SHELF SOFTWARE TESTING PROCESS, 2ND EDITION

Testing IT provides a complete, off-the-shelf software testing process framework for any testing practitioner who is looking to research, implement, roll out, adopt, and maintain a software testing process. It covers all aspects of testing for software developed or modified in-house, modified or extended legacy systems, and software developed by a third party. Software professionals can customize the framework to match the testing requirements of any organization, and six real-world testing case studies are provided to show how other organizations have done this. Packed with a series of real-world case studies, the book also provides a comprehensive set of downloadable testing document templates, proformas, and checklists to support the process of customizing. This new edition demonstrates the role and use of agile testing best practices and includes a specific agile case study.

John Watkins has more than thirty years of experience in the field of software development, with some twenty-five years in the field of software testing. During his career, John has been involved at all levels and phases of testing and has provided high-level test process consultancy, training, and mentoring to numerous blue chip companies.

He is both a Chartered IT Professional and a Fellow of the British Computer Society, where he is an active member of the Specialist Group in Software Testing (SIGiST), previously serving on committees of the Intellect Testing Group (representing the U.K. technology industry) and the SmallTalk User Group.

He is the author of *Agile Testing: How to Succeed in an Extreme Testing Environment* and currently works for IBM's software group.

Simon Mills has more than thirty years of experience in the field of software quality, having transferred into the world of system testing from a business role. Simon has been involved in testing software in both business and technical or scientific environments from major investment and insurance systems to laser control, cryogenic control, and superconducting applications. He is the founder of Ingenuity System Testing Services, the preeminent testing authority in the field of electronically traded insurance in the United Kingdom.

Simon is widely published internationally in conference proceedings, papers, and contributions to books and has presented as an invited speaker in the United States, at EuroStar, and at the World Congress for Software Quality.

TESTING IT

An Off-the-Shelf Software Testing Process, 2nd Edition

JOHN WATKINS

IBM Software Group, U.K.

SIMON MILLS

Ingenuity System Testing Services Ltd., U.K.

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

Shaftesbury Road, Cambridge CB2 8EA, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9780521148016

First published 2001
Second edition 2011

A catalogue record for this publication is available from the British Library

Library of Congress Cataloging-in-Publication data
Watkins, John (John Edward)
Testing IT : an off-the-shelf software testing process / John Watkins, Simon Mills. – 2nd ed.
p. cm.
Includes index.
ISBN 978-0-521-14801-6 (pbk.)
1. Computer software – Testing. I. Mills, Simon. II. Title.
QA76.76.T48W38 2010
005.1´4–dc22 2010041956

ISBN 978-0-521-14801-6 Paperback

Additional resources for this publication at www.cambridge.org/9780521148016

To Francesca, Julie, and Valerie

Contents

Foreword to the Second Edition

Geoff Thompson

So what is this testing thing then?

A question that I and many other aspiring software testing professionals get asked frequently. We try to explain, and watch the questioner's eyes glaze over as he or she furtively starts looking around to find a reason to step away or change the subject.

Isn't it strange that this happens? Is it that the explanation is simply incomprehensible by any sane person, or is it that software testers themselves need to understand in simple terms what it is they do?

I can add my own personal perspective on this issue; I help organize the British Computer Society Specialist Group in Software Testing (BCS SIGiST), and I am regularly surprised by two things: first, 70% of the attendees have never attended any form of networking meeting before, and second (and perhaps more important), none of the attendees read any of the many software testing books that exist.

Having had the opportunity to discuss this with many of the attendees, it is clear that the reason for this is the content of the typical testing book – it's just too complex for them to get their teeth into and understand. What is needed is a straightforward, simple-to-read, and simple-to-use testing book.

When the first edition of *Testing IT* was released in 2001, this changed. For once there was a book that provided a simple overview of what testing was, with straightforward guidance for test practitioners, plus a selection of easy-to-use testing templates. Since then I have recommended *Testing IT* to numerous testers that I have met, as well as providing copies to the testers on projects I have managed, and have received positive feedback from them regarding the practical benefits that it provides.

With the publication of the second edition of *Testing IT*, John has built on the success of the first edition, revising and bringing it up to date to ensure it continues to be relevant for the next ten years and beyond. Having been a champion of the first edition, I was very pleased to have been invited to play a part in this process, having used my involvement in the Information Systems Examination Board (ISEB), International Software Testing Qualification Board (ISQTB), and the Test Maturity

Model Integrated (TMMi) initiatives to ensure the second edition continues to be relevant going forward.

So, back to my original question – what is this testing thing then? Well, although there is no single simple answer to this question, in my humble opinion, John's book goes a long way to helping practitioners involved in test process, test management, and testing make a really good stab at answering the question for themselves.

Foreword to the First Edition

Maurice Rosenburgh

Why is astronomy considered a science while astrology is considered only a pseudo-science? In other words, how can we prove that a theory faithfully describes reality, and that this theory can then be used to predict unknown facts? Karl Popper, the well-known philosopher, studied these problems and summarized his conclusions in one phrase: "The criterion of the scientific status of a theory is its falsability, or refutability, or testability."* For Popper, "confirming evidence should not count except when it is the result of a genuine test of the theory."

The testing process of a scientific theory is quite similar to the process of providing confirmation either to risky predictions or to attempts to falsify that theory. Testing is a complex activity. It has to simultaneously bear in mind the theory and the external reality; it has to provide objective answers to complex questions related to our own perceptions of a rational reality.

When developing software, we follow the same thought process, since one builds an abstract model between the external world and the user. In our software, we define strict processes that will guide our actions, and we build the data we want to manipulate in complex databases and templates.

Can we test our software with Popper's principles in mind? The answer is definitively yes, because software testing should not only be a confirmation that the application is working correctly but also that it will react correctly when unexpected conditions occur. This constant and complex relationship between the software one tests and external reality should guide testers in their daily work.

Although testing is usually perceived as a necessity in software development, it is rarely applied as a rigorous activity. Within many projects, testing is simply omitted; in others, it is executed with the distinct intent to prove that the application performs correctly under test conditions.

After reading John Watkins's *Testing IT* you will be convinced that testing is not as complex as it seems and that it can be managed like any other development activity.

* Karl Popper, *Conjectures and Refutations*. London: Routledge and Kegan Paul, 1963.

The first thing you notice in reading *Testing IT* is that John puts the accent on testing processes and real-world case studies, which are, in my opinion, the most important aspects of software testing, implicitly applying Popper's conclusions.

Testing IT is divided into three logical, distinct parts: Part 1 focuses on traditional testing processes. Although technology is evolving at lightening speed, processes remain. They become even more important because they are at the heart of any activity. You will find this part very useful since it presents the testing phases starting from unit testing up to regression testing in the order found on all projects. Each phase is presented in the same coherent plan, facilitating access to the information.

Part 2 gives practical case studies. Five projects are reported, each enhancing a different reality; we have again the confirmation that success is always related to a correct adaptation of processes to reality.

Part 3 presents ready-to-use templates and reference documents that you can adapt to your needs and that you will find very useful in your daily testing work.

Testing IT is not just another book on testing. It is, in addition, a guide for all testers who want to understand what testing is really about, and it proves once more that applying easy-to-understand processes is the key to success. In one word: indispensable.

I'm certain you will enjoy reading this book, as I did, and that you will keep it on your desk to profit in your daily work from John's rich experience.

Maurice Rozenberg, Paris
Author of *Test Logiciel* (1998, Eyrolles)

Acknowledgments

Second Edition Acknowledgments

I would like to express my gratitude to Dave Evans, Isabel Evans, Pete Kingston, and Jon Tilt for allowing me to cite their experiences in agile testing, and for their valuable insights into improving agile communications, revealing how to succeed with agile off-shore projects, and describing innovative solutions for visualizing project progress and defect detection (next time I see him, I must ask Dave Evans where he buys his lava lamps!). Thanks are also due to 2nd edition reviewers Anne Mills, Dave Burgin, and Nathalie Allen for their keen eyes and valuable observations and comments.

I am very grateful to Peter Quentin for his extensive knowledge of the ISEB and ISQTB syllabuses, and his advice and guidance on ensuring the content of the book was in line with current thinking.

Many thanks to independent testing guru Tom Gilb and Trond Johansen of Confirmit for allowing me to document their fascinating real-world agile software development and testing project in the form of a case study. Also, thanks to Gary Schwartz for his reviewing efforts on the Confirmit case study.

I would very much like to thank Geoff Thompson for his insightful, well-observed, and very flattering foreword to the 2nd edition (which is likely to cost me a fortune in beer), and for being such a fantastic ambassador and champion of the first edition of *Testing IT* (which I guess will more than cover the cost of the beers).

Special thanks to Simon Mills for kindly agreeing to join the *Testing IT* project, and for adding to and enhancing the quality of the 2nd edition by bringing his pragmatic and extensive experience and knowledge of testing and the testing process to the new edition.

Finally, many thanks to my editor Lauren Cowles and to David Jou for their support, encouragement, and encyclopedic knowledge of the publishing process.

First Edition Acknowledgments

I would very much like to thank the following people for their advice, assistance, and encouragement with the writing of the first edition of this book:

Martin Adcock, Steve Allott, Jo Armitage, Mike Ashton, James Bach, Chris Ball, Bob Bartlett, Clive Bates, Judith Bays, Lesley Beasant, Colin Bendall, Duncan Brigginshaw, Miriam Bromnick, Kevin Buchta, Dorothy Buck, Dave Burgin, Trevor J. Collins, Richard Coop, Victor Copeland, Chris Cupit, Jim Davies, Steve Dawson, Barbara Eastman, Will Eisner, Isabel Evans, Mark Fewster, Virpi Flyktman, Ian Francis, Ian George, Paul Gerrard, Peter Gillespy, Dot Graham, Sam Guckenheimer, Jon Hall, Marion Hampson, Richard Hands, Steve Harcombe, Kim Harding, David Hayman, Mark Hirst, Dave Hodgkinson, Alan and Alison Jones, John Kent, Shelagh Kidd, Sami Korppi, Shaun Legeyt, Carol Li, Neil Love, Nick Luft, Philip March, Andrew Martin, Peter McConnell, Alec McCutcheon, Aidus McVeigh, Simon Mills, Kevin Minier, Eric Nelson, Mike Newton, Tommi Oksanen, David Palfreeman, Richard Pollard, Andy Redwood, Susan and Andy Roach, Ranj Sekhon, Darran Shaw, David Shohet, Graham Shore, Michael Summerbell, Peter Terelak, Pete Thomas, Andrew Thompson, John Thompson, Richard Tinker, Julie Valentine, Paul Warne, Tony Wells, Martin Whitaker, and Helen White.

I would like to give particular thanks to Dorothy Graham for giving me the benefit of her extensive experience in the testing field; Graham Titterington for his comprehensive and informed review comments; Geoff Quentin for his testing insight, encyclopedic knowledge of testing standards, and extensive collection of testing references; and James Bach, Paul Gerrard, and Steve Hancock for their input and informed discussion regarding risk-based testing issues.

I would also like to thank the following people for their assistance in writing the case studies in Part 2 of this book, as well as for their general help and encouragement with the book and its contents: Morag Atkins, Ken Eves, Paul Goddard, Martin Kemble, David Marsh, Annette Philips, and James Toon.

I am also very grateful to Maurice Rozenberg for finding the time to both write the foreword to the book, and for giving me the benefit of his extensive testing expertise in reviewing the chapters, case studies, and appendices.

And last but certainly not least, I would like to express my appreciation for the insight and experience of my technical reviewer Kamesh Pemmaraju, and for the constant "encouragement" and guidance from my editor Lothlorien Homett.