



EXPLORATORY SOFTWARE TESTING

TIPS, TRICKS, TOURS, AND TECHNIQUES
TO GUIDE TEST DESIGN



JAMES A. WHITTAKER

“Great book! Whittaker delivers ideas that are innovative, smart, and memorable. He really knows how to inspire engineers to think differently about testing.”

—Patrick Copeland, Director of Test Engineering, Google

“James has perfected a fantastic manual testing methodology. The touring concept not only works, but it works so well that we’ve started sharing the tour concepts in the internal testing courses taught to all of our testers. If you want to bring your manual testing processes into the 21st century then read this book.”

—Alan Page, Director, Test Excellence, Microsoft

“I began working with James at IBM in 1990. Even then, he was inspiring testers and developers to think outside the box. With this book he’s taken his passion for software quality to a whole new level. Read it and watch yourself become a better tester. James is the real deal and this book should be read by every tester and software developer on the planet who cares about software quality or just wants to have more fun doing what they do.”

—Kaushal K. Agrawal, Sr. Director of Engineering, Cisco Systems

“James Whitaker is a true visionary in the world of testing. uTest and our global community of QA professionals regularly look to James for inspiration, interpretation of trends, and overarching testing wisdom. Now he’s finally written it down for everyone else and our industry will be smarter because of it.”

—Doron Reuveni, CEO and Co-Founder, uTest

“Only someone like James Whittaker would think of combining the idea of tourism with software testing in such a novel way—and only James could pull it off. The tours approach provides a memorable and extremely effective mental model that combines right degree of structure and organization with plenty of room for exploration and creativity. Bugs beware!”

—Alberto Savoia, Google

“James is one of the best speakers around on software testing and reading his book is much like hearing him speak. If you want to increase your knowledge of testing and make yourself a better tester, this is the book for you.”

—Stewart Noakes, Chairman and Co-Founder, TCL Group Ltd.

Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design

Table of Contents

Table of Contents

Foreword

Preface

Chapter 1 The Case for Software Quality

The Magic of Software

The Failure of Software

Conclusion

Exercises

Chapter 2 The Case for Manual Testing

The Origin of Software Bugs

Preventing and Detecting Bugs

Preventing Bugs

Detecting Bugs

Manual Testing

Scripted Manual Testing

Exploratory Testing

Conclusion

Exercises

Chapter 3 Exploratory Testing in the Small

So You Want to Test Software?

Table of Contents

Testing Is About Varying Things

User Input

- What You Need to Know About User Input

- How to Test User Input

State

- What You Need to Know About Software State

- How to Test Software State

Code Paths

User Data

Environment

Conclusion

Exercises

Chapter 4 Exploratory Testing in the Large

Exploring Software

The Tourist Metaphor

Touring Tests

- Tours of the Business District

- Tours Through the Historical District

- Tours Through the Entertainment District

- Tours Through the Tourist District

- Tours Through the Hotel District

- Tours Through the Seedy District

Putting the Tours to Use

Conclusion

Exercises

Chapter 5 Hybrid Exploratory Testing Techniques

Scenarios and Exploration

Table of Contents

Applying Scenario-Based Exploratory Testing

Introducing Variation Through Scenario Operators

- Inserting Steps

- Removing Steps

- Replacing Steps

- Repeating Steps

- Data Substitution

- Environment Substitution

Introducing Variation Through Tours

- The Money Tour

- The Landmark Tour

- The Intellectual Tour

- The Back Alley Tour

- The Obsessive-Compulsive Tour

- The All-Nighter Tour

- The Saboteur

- The Collectors Tour

- The Supermodel Tour

- The Supporting Actor Tour

- The Rained-Out Tour

- The Tour-Crasher Tour

Conclusion

Exercises

Chapter 6 Exploratory Testing in Practice

The Touring Test

Touring the Dynamics AX Client

- Useful Tours for Exploration

- The Collectors Tour and Bugs as Souvenirs

- Tour Tips

Table of Contents

Using Tours to Find Bugs

- Testing a Test Case Management Solution

- The Rained-Out Tour

- The Saboteur

- The FedEx Tour

- The TOGOF Tour

The Practice of Tours in Windows Mobile Devices

- My Approach/Philosophy to Testing

- Interesting Bugs Found Using Tours

- Example of the Saboteur

- Example of the Supermodel Tour

The Practice of Tours in Windows Media Player

- Windows Media Player

- The Garbage Collectors Tour

- The Supermodel Tour

- The Intellectual Tour

- The Intellectual Tour: Boundary Subtour

The Parking Lot Tour and the Practice of Tours in Visual Studio Team System Test Edition

Tours in Sprints

Parking Lot Tour

Test Planning and Managing with Tours

- Defining the Landscape

Planning with Tours

Letting the Tours Run

Analysis of Tour Results

Making the Call: Milestone/Release

- In Practice

Table of Contents

Conclusion

Exercises

Chapter 7 Touring and Testings Primary Pain Points

The Five Pain Points of Software Testing

Aimlessness

Define What Needs to Be Tested

Determine When to Test

Determine How to Test

Repetitiveness

Know What Testing Has Already Occurred

Understand When to Inject Variation

Transiency

Monotony

Memorylessness

Conclusion

Exercises

Chapter 8 The Future of Software Testing

Welcome to the Future

The Heads-Up Display for Testers

Testipedia

Test Case Reuse

Test Atoms and Test Molecules

Virtualization of Test Assets

Visualization

Testing in the Future

Post-Release Testing

Conclusion

Table of Contents

Exercises

Appendix A: Building a Successful Career in Testing

How Did You Get into Testing?

Back to the Future

The Ascent

The Summit

The Descent

Appendix B: A Selection of JWs Professorial Blog

Teach Me Something

Softwares Ten Commandments

1. Thou Shalt Pummel Thine App with Multitudes of Input
2. Thou Shalt Covet Thy Neighbors Apps
3. Thou Shalt Seek Thee Out the Wise Oracle
4. Thou Shalt Not Worship Irreproducible Failures
5. Thou Shalt Honor Thy Model and Automation
6. Thou Shalt Hold Thy Developers Sins Against Them
7. Thou Shalt Revel in App Murder (Celebrate the BSOD)
8. Thou Shalt Keep Holy the Sabbath (Release)
9. Thou Shalt Covet Thy Developers Source Code

Testing Error Code

Will the Real Professional Testers Please Step Forward

The Common Denominators I Found Are (In No Particular Order)

My Advice Can Be Summarized as Follows

Strike Three, Time for a New Batter

Formal Methods

Tools

Process Improvement

The Fourth Proposal

Table of Contents

Software Testing as an Art, a Craft and a Discipline

Restoring Respect to the Software Industry

The Well-Intentioned but Off-Target Past

Moving On to Better Ideas

A Process for Analyzing Security Holes and Quality Problems

Appendix C: An Annotated Transcript of JWs Microsoft Blog

Into the Blogoshere

July 2008

Before We Begin

PEST (Pub Exploration and Software Testing)

Measuring Testers

Prevention Versus Cure (Part 1)

Users and Johns

Ode to the Manual Tester

Prevention Versus Cure (Part 2)

Hail Europe!

The Poetry of Testing

Prevention Versus Cure (Part 3)

Back to Testing

August 2008

Prevention Versus Cure (Part 4)

If Microsoft Is So Good at Testing, Why Does Your Software Still
Suck?

Prevention Versus Cure (Part 5)

Freestyle Exploratory Testing

Scenario-Based Exploratory Testing

Strategy-Based Exploratory Testing

Feedback-Based Exploratory Testing

The Future of Testing (Part 1)

Table of Contents

The Future of Testing (Part 2)

September 2008

On Certification

The Future of Testing (Part 3)

The Future of Testing (Part 4)

The Future of Testing (Part 5)

October 2008

The Future of Testing (Part 6)

The Future of Testing (Part 7)

The Future of Testing (Part 8)

Speaking of Google

Manual Versus Automated Testing Again

November 2008

Software Tester Wanted

Keeping Testers in Test

December 2008

Google Versus Microsoft and the Dev:Test Ratio Debate

January 2009

The Zune Issue

Exploratory Testing Explained

Test Case Reuse

More About Test Case Reuse

Im Back

Of Moles and Tainted Peanuts

Index