



## PRACTICAL SOFTWARE TESTING

Welcome to the world of software testing. Since most testers have had little or no training, this class is specifically designed just for you. Those of you that are testers will find that the material will fill in the gaps between your experience and QA/test theory. Finally, you'll learn the reasons why the testing process works the way it does.

### Who should attend?

- People new to testing. You've tried *Try Testing* and liked it; now get the skills you need to begin your testing career
- Junior and mid-level testers – elevate your QA I.Q. and add to your tester's toolbox

**Prerequisite:** *Tester Assessment Test*. Take this short, online test and receive immediate feedback on which courses are right for you.

**Course Duration:** 3 days

**What you get:** A buggy application to practice on, tip sheets, likely interview questions, free two week follow-up support from the instructor, all class materials and new knowledge you can apply the minute you get back to work!

## Course Outline

### Testing Basics

This class is more than how to test; you'll learn how to *think* like a tester! We start at the beginning and define what a test, test data and expected results are. You'll learn how to identify bugs and how to choose test data. We'll also talk about what the difference between quality assurance and testing is. You'll get to try and practice everything we talk about on a buggy application. As you learn bug hunting skills you'll be given an easy memory aid to use to help you get good coverage as you test. You'll even get to run several different styles of test cases from ad-hoc to very detailed.

### Bug Writing Basics

You've found a bug, so now how do you write it down with all the information that an engineer needs to fix it? Not to stress you or anything, but a bug report is often the sole output of a tester and you'll be judged by what you find and how you write it. Breathe easier because here you'll learn an easy format to remember everything the engineers need and to make your bug reports look professional too. You'll get to see some pretty nasty, poorly written bugs and try to re-write them more accurately and professionally. You'll now enter into the bug database all those bugs you found in Testing Basics.

### Bug Life

After you write the bug report, then what happens? Go deep inside the life of a bug (report) and see how it travels through the company and winds up back in your hands for verification. What are all those extra fields like Priority and Severity for? You'll find out, along with how to decide what priority, severity or status to enter. Learn how to verify a bug fix and what to do in tricky situations with partially fixed bugs or deferred bugs and how to "scrub" bugs reported by non-testers. You'll practice scrubbing and verifying bugs on a "fixed" version of the buggy application.



## Test Reporting

"How's the software looking?" Everyone from the CEO to the technical support analysts wants to know. "What have you tested?" "When will you be done testing?" Learn what test results and test analysis are, how to write them and how to use them. In this class you'll learn how to communicate the answers to another tester, an engineer, the QA/testing manager and the CEO. You'll also discover what each of those people does with the information you provide.

## Software Development and Test Lifecycles

How does a product move from an idea into the customer's hands? Where does testing fit in and what do you, as a tester, do all along the way? You'll learn all that and we'll discuss the roles of the CEO, marketing, engineering, testing, support, documentation, the documents produced and how they are used. You'll learn why we don't want to test just any old thing, how to conduct Acceptance and Regression Tests, and how to use quality criteria to accept or reject a build of software. You'll actually receive a bug fixed version/build of the buggy application. You'll conduct an acceptance test, notify the team of the outcome and then do a regression test.

## Tester's Toolbox Basics

When you start your job as a tester, your manager will automatically assume that you know how to use some tools. We'll briefly introduce you to these tools and give you a chance to try out the ones that are new to you. Some of the tools you'll meet are: anti-virus, disk doctor, email, meeting schedulers, browsers, WinZip, backup, zip drives, disk scanning and defragmenting tools.

## QA Theory – The story of how it all began and why we're where we are today

When did QA start? How (and why) did the field evolve to where we are today? In this class, we'll tell the story of how it all began. We'll also discuss some common QA conundrums such as "When do you stop testing?", "Have you tested thoroughly?", "How can you prevent defects?" and the always popular "Can we ship yet?"

## On the Job

Now you know how to test and communicate bugs and results. You know where you fit in, what everyone else is doing and how it all began. But there are a few more things about being a tester to learn before you venture forth. In this class you'll learn how to handle yourself in a meeting and work with software people (and difficult people), how to talk to engineers and upper management and negotiate commitments and priorities, and (finally) you'll learn a few survival tips from those that have tested before you.

## Getting the Job

Now you're ready to be a tester. So, before you leave class we'll give you some tips about what to put in a resume, how to work with technical recruiters and what interview questions to expect.