Software Testing Bootcamp

- Create a Master Test Plan using the Entry, Task, Verification, and Exit (ETVX) technique
- Identify key strategy issues that influence all test planning
- Document the heart of a QA Test Plan with the Test/Traceability Matrix
- Communicate risks using a risk planning template
- Practice six different black-box (functional) test design techniques to find the most bugs with the least effort (no redundancy)
- Explore white-box (structural) methods to add more completeness to your test cases
- Examine the exploratory testing approach to replace ad hoc testing
- Choose which test design techniques are most useful for given projects

Applicable regardless of your methodology: Agile, Waterfall, DevOps or any journey along the way. Testing is a critical element for successful software development and delivery. This course provides a framework for all activities for successful testing. All techniques are practiced with hands-on exercises. Students can choose to use course-provided examples or an actual example from their current projects. It can be tailored for minimal documentation for Agile projects or full documentation for traditional projects – all options are included.

Software Testing Bootcamp starts with brainstorming about the current challenges being faced by the students. In real-time, the instructor will decide which materials to include based on these challenges (there are optional materials for white box testing and testing User Scenarios from Use Cases). The course concludes with the instructor mapping the identified challenges to the course techniques that can help with the individual challenges.

Who Should Attend

This course is appropriate for both novice and experienced software testing roles. Developers in an agile environment who are expected to write test cases will find it extremely useful. Test and development leadership role can also benefit. A background of basic development processes and test levels is helpful but not required.

Course Outline

Testing Fundamentals
Brainstorming class challenges with testing
Testing within development lifecycles
Testing as a part of quality assurance
Importance of risk-based testing
Documenting risk (exercise)

Test Planning
Identifying testing strategies (exercise)
Creating a testing Master Plan using the ETVX diagram (exercise)
Starting a Test Plan with a Test Matrix (exercise)

Introduction to Test Design
What are test cases?
The impossibility of testing everything

Black Box Art
Hunches and guessing
Exploratory testing (exercise)
Creating creative invalids
Using a defect taxonomy
Choosing the appropriate technique

Wrap-up
Review
Challenges revisited

OPTIONAL
White Box Science
What is white-box testing?
Coverage measures
Control flow concepts

Black Box Science

What is black-box testing?
Black-box testing at different testing levels
Equivalence class partitioning (exercise)
Boundary value testing (exercise)
Decision table testing (exercise)
State-transition diagram testing (exercise)
Chow’s N-Switch technique (exercise)
All pairs test methods (exercise)

Applying control flow to code (exercise)
McCabe’s Design Predicate technique for Integration Testing (exercise)

Creating Test Cases from Use Case Scenarios

High level test cases
Detailed test cases
Data challenges