FUNDAMENTALS OF SOFTWARE TESTING

Course Code: 2515

Learn the fundamental techniques and approaches to software testing and better understand what to test, how to test it, and in what contexts certain practices make sense.

Testing is a critical role in software development that requires special skills and knowledge not commonly taught to software developers, business analysts, and project managers. This often results in insufficient time and resources being allocated for this important function, and quality suffers—as do the users of the software. Equally important is the need to measure quality quickly and efficiently, because limitations in resources and schedules are realities that aren’t going away. We need to do the best we can with what we have and still deliver high-quality, proven software.

This course provides an eye-opening view into this challenging task. You will get a complete picture of the testing process, how it fits into the development life cycle, how to properly scope and prioritize testing activities, and which techniques to use for optimal results. This software-testing training begins with a deep-dive into the Universal Testing Method and follows with a close look at testing phases, testing approaches, non-functional testing, and testing for different platforms. As time permits, you will cover bonus material that includes an introduction to automation testing and behavior-driven development.

Cases studies, examples, in-class exercises and reviews are used to reinforce the concepts and practices. You will gain experience in modeling the test space, establishing scope and application coverage, identifying “testing oracles”, writing test procedures, and more. You will leave the course with a wealth of new knowledge you can begin to apply immediately.

Go Beyond Basic Functional Testing to Improve the Overall Experience for Users

In addition to hands-on experience, you will gain insight into a wide variety of testing aspects that go beyond the usual requirements-based functional testing. You will learn about positive and negative testing concepts and effective methods for exploratory testing. You will discover testing aspects that are often overlooked and best practices for addressing them, and you will be given excellent references to guide you further in best practices and selecting effective tools.
You will come away from this software testing training course with many ideas that you can apply in your own projects to improve the effectiveness and efficiency of testing efforts. You will learn the best ways to report on the testing activity, with practical, fast and effective testing as the focus of this course.

Case Studies, Exercises, and In-Class Reviews

As the Universal Testing Method (UTM) unfolds during class, real-life case studies and examples will help drive home the pertinent concepts. Each one is discussed as it pertains to a step in the UTM and encourages you to think "outside the box." You will also work in groups through key steps in the UTM using a real application accessible via the Internet. The same application is used for each exercise, with each step building on the previous one, to provide hands-on experience and new skills you can begin working with on the job right away. For example, you will:

- Develop a model of the application
- Use your model to determine test coverage
- Identify test oracles for the application
- Create test cases based on the oracles
- Run your tests against the live application

Each chapter is followed by a series of review questions that you will discuss in class. The review questions draw on key points from each section and help you retain the concepts. Also included is an "introspective review" that helps you comprehend how the material applies in your own circumstances.

Who Needs to Attend

- Testers of all types and levels
- Professionals from other disciplines who perform their own testing or are involved in testing
- Quality assurance (QA) professionals, including QA managers and directors
- Software engineers
- Business analysts
- Project managers
- IT specialists (security, capacity, management, networking, etc.)
- Business stakeholders
- Outsourcer staff (buyers and suppliers)
- Application development managers
FUNDAMENTALS OF SOFTWARE TESTING
Course Code: 2515

CLASSROOM LIVE $1,495 USD 2 days

Classroom Live Outline
1. Introduction and Overview
2. What to Test and How to Test It-The Universal Testing Method
   • Modeling the testing space
   • Determining test coverage
   • Determining test oracles
   • Determining test procedures
   • Configuring the test system
   • Operating the test system
   • Observing the test system
   • Evaluating testing results
   • Reporting test results
3. Common Phases of Testing
   • Characteristics of the six common phases of software testing
     ‧ Who typically performs the tests
     ‧ The scope of the tests
     ‧ Common tools used to perform the tests
     ‧ Useful techniques and practices
     ‧ What the phase addresses well
     ‧ What the phase often misses
   • Test phases and contexts
     ‧ Unit testing
     ‧ Integration testing
     ‧ System testing
     ‧ Regression testing
     ‧ Acceptance testing
     ‧ Alpha/Beta testing
The V-Model for software testing
Testing phases in an agile project

4. Approaches to Testing
   • The testing approach continuum
   • Scripted testing
   • Exploratory testing
   • Scenarios, checklists, and charters
   • Structuring tester interaction

5. Non-Functional Testing
   • Usability
   • Accessibility
   • Performance, scalability, and capacity
   • Security testing
   • Internationalization and localization
   • Compatibility and portability
   • Maintainability and supportability
   • Testability
   • Disaster recovery / business continuity

6. Platform Specialization
   • Mobile
   • Web
   • Cloud
   • Package implementations (COTS)
   • SOA
   • Data warehouse and business intelligence
   • Telephony and voice

7. Test Automation (Bonus Section)
   • Automation types
   • Web automation demonstration
   • Challenges of automation
   • Optimizing automation efforts
   • Tool selection process

8. Behavior Driven Development (BDD) and Test Driven Development (TDD) (Bonus Section)
   • BDD and TDD defined
   • Feature files and the Gherkin language
   • Gherkin tools for different languages
FUNDAMENTALS OF SOFTWARE TESTING

Course Code: 2515

VIRTUAL CLASSROOM LIVE  $1,495 USD  3 days

Virtual Classroom Live Outline

1. Introduction and Overview

2. What to Test and How to Test It-The Universal Testing Method
   • Modeling the testing space
   • Determining test coverage
   • Determining test oracles
   • Determining test procedures
   • Configuring the test system
   • Operating the test system
   • Observing the test system
   • Evaluating testing results
   • Reporting test results

3. Common Phases of Testing
   • Characteristics of the six common phases of software testing
     - Who typically performs the tests
     - The scope of the tests
     - Common tools used to perform the tests
     - Useful techniques and practices
     - What the phase addresses well
     - What the phase often misses
   • Test phases and contexts
     - Unit testing
     - Integration testing
     - System testing
     - Regression testing
     - Acceptance testing
     - Alpha/Beta testing
The V-Model for software testing
Testing phases in an agile project

4. Approaches to Testing
   • The testing approach continuum
   • Scripted testing
   • Exploratory testing
   • Scenarios, checklists, and charters
   • Structuring tester interaction

5. Non-Functional Testing
   • Usability
   • Accessibility
   • Performance, scalability, and capacity
   • Security testing
   • Internationalization and localization
   • Compatibility and portability
   • Maintainability and supportability
   • Testability
   • Disaster recovery / business continuity

6. Platform Specialization
   • Mobile
   • Web
   • Cloud
   • Package implementations (COTS)
   • SOA
   • Data warehouse and business intelligence
   • Telephony and voice

7. Test Automation (Bonus Section)
   • Automation types
   • Web automation demonstration
   • Challenges of automation
   • Optimizing automation efforts
   • Tool selection process

8. Behavior Driven Development (BDD) and Test Driven Development (TDD) (Bonus Section)
   • BDD and TDD defined
   • Feature files and the Gherkin language
   • Gherkin tools for different languages

Jul 29 - 31, 2019 | 12:00 - 4:30 PM EST
Aug 12 - 14, 2019 | 12:00 - 4:30 PM EST
Sep 16 - 18, 2019 | 12:00 - 4:30 PM EST
Oct 1 - 3, 2019 | 12:00 - 4:30 PM EST
Nov 6 - 8, 2019 | 12:00 - 4:30 PM EST
Dec 11 - 13, 2019 | 12:00 - 4:30 PM EST
GLOBAL KNOWLEDGE

FUNDAMENTALS OF SOFTWARE TESTING
Course Code: 2515

PRIVATE GROUP TRAINING  2 days
FUNDAMENTALS OF SOFTWARE TESTING
Course Code: 2515

PRIVATE GROUP TRAINING
2 days

Visit us at www.globalknowledge.com or call us at 1-866-716-6688.

Date created: 7/14/2019 11:45:37 AM
Copyright © 2019 Global Knowledge Training LLC. All Rights Reserved.