Testing Concepts

Lesson 00:

Capgemini

Course Goals and Non Goals

Course Goals

- At the end of this program, participants gain an understanding of Verification & Validation process in project
- Participants get an understanding of different testing approaches, techniques & types
- They also learn how to create effective test cases using the different testing techniques to get a good test coverage of a software application
- Participants get an understanding of Importance of monitoring progress in testing process & different project metrics

Course Non Goals

- This course does not cover automation process of testing.

Pre-requisites

None

Intended Audience

Test Engineers, Software Engineers and Senior Software Engineers

Day Wise Schedule

Day 1

- Lesson 1: Fundamentals of Testing
- Lesson 2: Testing throughout SDLC
- Lesson 3 : Static Testing
- Lesson 4 : Test Techniques

Day 2

- Lesson 4: Test Techniques
- Lesson 5: Test Management and Test Metrics
- Lesson 6: Requirement Engineering
- Lesson 7 : Use Cases Testing

Table of Contents

Lesson 1: Fundamentals of Testing

- 1.1 What is Testing?
- 1.1.1 Typical Objectives of Testing
- 1.1.2 Testing and Debugging
- 1.2 Why is Testing Necessary?
- 1.2.1 Testing’s Contributions to Success
- 1.2.2 Quality Assurance and Testing
- 1.2.3 Errors, Defects, and Failures
  - Reasons behind Errors
- 1.2.4 Defects, Root Causes and Effects
  - Cost of Software Defects
  - Importance of Testing Early in SDLC phases

Table of Contents

Lesson 1: Fundamentals of Testing

- 1.3 Seven Testing Principles
  - Economic of Testing
  - Scope of Software Testing
  - Factors influencing Software Testing
- 1.4 Test Process
- 1.4.1 Test Process in Context
- 1.4.2 Test Activities and Tasks
- 1.4.3 Test Work Products
- 1.4.4 Traceability between the Test Basis and Test Work Products
- 1.5 The Psychology of Testing
- 1.5.1 Human Psychology and Testing
  - Attributes of a good Tester
  - Code of Ethics for Tester
- 1.5.2 Tester’s and Developer’s Mindsets

Table of Contents

Lesson 2: Testing Throughout the Software Development Life Cycle

- 2.1 Software Development Lifecycle Models
- 2.1.1 Software Development and Software Testing
- 2.1.2 Software Development Lifecycle Models in Context
- 2.2 Test Levels
- 2.2.1 Component Testing
- 2.2.2 Integration Testing
- 2.2.3 System Testing
- 2.2.4 Acceptance Testing
- 2.3 Test Types
- 2.3.1 Functional Testing
- 2.3.2 Non-functional Testing
- 2.3.3 White-box Testing
- 2.3.4 Change-related Testing
- 2.3.5 Test Types and Test Levels

Table of Contents

Lesson 2: Testing Throughout the Software Development Life Cycle

- 2.4 Maintenance Testing
- 2.4.1 Triggers for Maintenance
- 2.4.2 Impact Analysis for Maintenance
- 2.5 Test Case Terminologies
- 2.6 Test Data

Table of Contents

Lesson 3: Static Testing

- Types of Testing Techniques
- Differences between Static and Dynamic Testing
- 3.1 Static Testing Basics
- 3.1.1 Work Products that Can Be Examined by Static Testing
- 3.1.2 Benefits of Static Testing
- 3.2 Review Process
- 3.2.1 Work Product Review Process
- 3.2.2 Roles and responsibilities in a formal review
- 3.2.3 Review Types
- 3.2.4 Applying Review Techniques
- 3.2.5 Success Factors for Reviews

Table of Contents

Lesson 4: Test Techniques

- 4.1 Categories of Test Techniques
- 4.1.1 Choosing Test Techniques
- 4.1.2 Categories of Test Techniques and Their Characteristics
- 4.2 Black-box Test Techniques
- 4.2.1 Equivalence Partitioning
- 4.2.2 Boundary Value Analysis
- 4.2.3 Decision Table Testing
- 4.2.4 State Transition Testing
- 4.2.5 Use Case Testing
- 4.3 White-box Test Techniques
- 4.3.1 Statement Testing and Coverage
- 4.3.2 Decision Testing and Coverage
- 4.3.3 The Value of Statement and Decision Testing
- 4.4 Experience-based Test Techniques
- 4.4.1 Error Guessing
- 4.4.2 Exploratory Testing
- 4.4.3 Checklist-based Testing

Table of Contents

Lesson 5: Testing Management & Test Metrics

- 5.1 Test Organization
- 5.1.1 Independent Testing
- 5.1.2 Tasks of a Test Manager and Tester
- 5.2 Test Planning and Estimation
- 5.2.1 Purpose and Content of a Test Plan
- 5.2.2 Test Strategy and Test Approach
- 5.2.3 Entry Criteria and Exit Criteria (Definition of Ready & Done)
- 5.2.4 Test Execution Schedule
- 5.2.5 Factors Influencing the Test Effort
- 5.2.6 Test Estimation Techniques

Table of Contents

Lesson 5: Testing Management

- 5.3 Test Monitoring and Control
- 5.3.1 Metrics Used in Testing
- 5.3.2 Purposes, Contents, and Audiences for Test Reports
- 5.4 Configuration Management
- 5.5 Risks and Testing
- 5.5.1 Definition of Risk
- 5.5.2 Product and Project Risks
- 5.5.3 Risk-based Testing and Product Quality
- 5.6 Defect Management

Table of Contents

Lesson 6: Overview on Requirements & Requirement Gathering

- 6.1 Evolution of Requirements
- 6.2 Who provides the Requirements?
- 6.3 Challenges in Requirement Gathering
- 6.4 Why do we need good requirements?
- 6.4.1 Characteristics & Impact of bad Requirements
- 6.5 Requirement Engineering
- 6.6 Functional Vs Non-Functional Requirements
- 6.8 Non Functional Requirements: FURPS +
- 6.9 Stable and Volatile Requirements
- 6.10 Baselining Requirements
- 6.11 Requirements Traceability
- 6.12 Requirements Change

Table of Contents

Lesson 7: Use Case Testing

- 7.1 Use case modeling
- 7.2 Advantage of use cases
- 7.3 Actor
- 7.4 Goals and Requirements
- 7.5 Goals and scenarios
- 7.6 Naming Conventions
- 7.7 Alternate Path
- 7.8 Exceptions
- 7.9 Errors
- 7.10 Precondition & Post-condition
- 7.11 Good practices
- 7.12 Failure scenarios

References

Student material:

- Class Book (presentation slides with notes)
- Lab book

Book:

- Testing Computer Software – Cem Kaner
- Software Testing in the Real World – Edward Kit
- Effective methods for Software testing – William E. Perry
- Software Engineering -A Practitioner's Approach – Roger S. Pressman
- Software Testing Techniques – Boris Beizer

Web-site:

- http://www.softwaretesting.org
- http://www.onestoptesting.com/introduction/

Next Step Courses

Automation testing