Writing and Running Tests

Course Summary

Description

This interactive workshop addresses management aspects of carrying out tests and reporting test results and test status. Since test execution typically occupies the greatest portion of test time, it is essential to leverage limited resources to maximize defect detection and ejection. That means isolating and reporting defects so they can be removed quickly. It also means analyzing testing results to enable focusing economically on areas with the biggest payback. Finally, it involves appropriate use of automated tools and other techniques to organize and handle often huge volumes of tests and test results. Exercises enhance learning by allowing participants to practice techniques with an actual case.

Objectives

At the end of this course, students will be able to:

- Use manual and automated techniques to structure and manage large volumes of testware.
- Write defect and status reports that help assure the important bugs are removed before delivery.
- Use methods to reliably keep testing efforts on track and economical.
- Use measures to monitor both testing of particular software and overall test process effectiveness.

Topics

- Defining Test Cases
- Testing Infrastructure--Technical
- Isolating and Reporting Defects
- Managing Test Execution

Audience

This course has been designed for systems and test managers, testers, analysts, and others who plan, oversee, and/or carry out testing of software products.

Duration

One day

Writing and Running Tests

Course Outline

I. Defining Test Cases
- A. The challenge—how much to write
- B. Too much, embedded keystroke procedure
- C. Too little, ad hoc exploratory error guessing
- D. Bigger challenge—writing right tests
- E. Traditional reactive test case approach
- F. Proactive Testing™ spots overlooked tests
- G. Test planning and design structure benefits
- H. Proactive risk-based test selection
- I. What a test case is
- J. Additional test case documentation
- K. Test case specification vs. data values
- L. Simple and complex tests
- M. Black box traceability coverage
- N. White box tests and degrees of coverage
- O. Other test case dimensions—load, duration
- P. Efficiently writing test cases
- Q. Test script format
- R. Capturing test data in a matrix
- S. Screen images and file layouts
- T. Exercise: Testing use case paths
- U. Exercise: Defining inputs/conditions

II. Testing Infrastructure--Technical
- A. Establishing the technical test environment
- B. Creating and maintaining test beds
- C. Hardware/software capabilities and versions
- D. Configuration management, essential tool
- E. Status-based defect tracking, also essential
- F. Tools for designing/generating test cases
- G. Tools to assist and monitor testing
- H. Automated test execution tools
- I. Automated capture/replay scripts
- J. Data-driven tests
- K. Action words automate without scripting
- L. Load testing tools, different type of test case
- M. Test management tools
- N. Issues governing automation of testing

III. Isolating and Reporting Defects
- A. Defect isolation vs. debugging
- B. Exercise: Writing an effective bug report
- C. Exercise: Defect isolation
- D. Categorizing defects, identifying trends
- E. Defect age
- F. Prioritizing, getting the important bugs fixed
- G. Defect density, mean times to fail and fix
- H. Defect statistics, degrees of precision
- I. Exercise: Test status reports

IV. Managing Test Execution
- A. Monitoring test progress and earned value
- B. Test cycles
- C. Improving accuracy of test estimates
- D. Projecting when it will be “good enough”
- E. Estimating remaining defects
- F. Seeding and pooling techniques
- G. Economical advantages of static reviews
- H. Measuring test detection effectiveness