

Proactive Test Process Management

Course Summary

Description

Your testing process is the way testing is done in your organization and involves much more than merely procedures and paperwork, such as the people and organizational issues essential for support and success. This interactive course describes key elements a testing process needs in order to be effective and efficient and explains how Proactive Testing can go well beyond traditional reactive testing to deliver higher quality systems quicker and cheaper. Managing the testing process involves defining, monitoring, and actively guiding/adjusting how individual testing projects plan, design, organize, direct, and control testing of specific systems under test (SUT). Techniques are shown to identify testing and necessary supportive tasks; estimate required time, effort, and resources; keep testing on track; and monitor and evaluate the testing process itself. Exercises enhance learning by allowing participants to practice techniques.

Objectives

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

- Know what REAL vs. Presumed Process identification and elements a testing process should include.
- Manage the testing process vs. managing individual testing projects.
- Understand a structured Proactive Testing process that yields better software and also cuts development time.
- Effectively and efficiently use-busywork methods for planning, designing, and organizing tests.
- Cross-project components, roles, and tasks to provide adequate testing infrastructure environment.
- Understand issues and techniques to gain support within the organization for suitable testing.
- Measure and monitor both testing of particular software and overall test process effectiveness.

Topics

- Defining the Testing Process
- Managing the Test Environment
- Planning and Designing Testing

Audience

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

Prerequisites

There are no prerequisites required for this course.

Duration

One day

Proactive Test Process Management

Course Outline

I. Defining the Testing Process

- A. Defining and evaluating your current process
- B. Relation of testing projects to testing process
- C. Commonly-recognized test process issues
- D. REAL vs. Presumed Processes
- E. "How much process?" is mistaken question
- F. Reactive testing—out of time, but not tests
- G. Current status of defect injection, detection
- H. Quality goals and the testing process
- I. Models of test process maturity
- J. Documentation of and from the process
- K. Managing testing Key Process Areas
- L. "Good practices" limitations on using alone
- M. Life cycle involvement, iterative approaches
- N. Static and dynamic testing, issues
- O. CAT-Scan Approach to find more errors
- P. Independent test group issues and strengths
- Q. Planning and designing tests
- R. Test execution and status reporting
- S. Defect categorization and analysis
- T. Customs, beliefs, management practices
- U. Measuring and managing the testing process
- V. Secrets for overcoming resistance to testing
- W. Proactive Testing better, quicker, cheaper
- X. How testing can save the developer's time
- Y. Caution: "Emperor's New Clothes" risk
- Z. Recognizing your REAL hidden issues

II. Managing the Test Environment

- A. Establishing the testing infrastructure
- B. Maintaining test beds, databases, labs
- C. Automated tools for testing
- D. Configuration management, defect tracking
- E. Standards, policies, and procedures
- F. Roles, staffing, training, and career paths

III. Planning and Designing Testing

- A. Risk elements, relation to testing
- B. Proactive vs. reactive risk analysis
- C. IEEE Standard for Test Documentation
- D. Master Test Plan counterpart to project plan
- E. Risk-based way to define test units
- F. Letting testing drive development
- G. Preventing major cause of overruns
- H. True agility, value vs. busywork
- I. Regression tests, avoiding minefield effect
- J. IEEE Standard on Unit Testing
- K. Graphical technique to simplify integrations
- L. Integration test plans prevent schedule slips
- M. Avoiding the major cause of estimate errors
- N. Top-down vs. bottom-up sizing techniques
- O. Providing for ancillary and support tasks
- P. Critical Path dependencies, concurrencies
- Q. Controlling test execution relative to plan
- R. Earned value
- S. Measuring and improving test effectiveness