Course Outline

Risk Management Methods for Test Projects

Course Summary

Description

Testing is our primary means of reducing risks related to systems and software. All testing is risk-based in that there are more potential tests than we have time and resources to run, so we choose those that address higher risks. The difficulty is adequately identifying the potential risks, for which a number of structured techniques are described. Risk management involves obtaining, directing, and controlling resources for continuously planning, strategizing, identifying, analyzing, prioritizing, preventing, responding to, mitigating, monitoring, controlling, and reporting risk. Key types of risks and identification approaches are distinguished. Powerful Proactive Testing techniques are demonstrated that enhance conventional reactive testing risk analysis to enable testing higher risks not only more but earlier. Exercises aid learning by allowing participants to practice applying practical methods to realistic examples.

Objectives

After taking this course, students will be able to understand:

- The elements of risk and risk response, and their relation to software development and testing.
- Importance of distinguishing business, management, functional, and technical risk causes and effects.
- Alternative methods for identifying, analyzing, classifying, and prioritizing the several types of risks.
- Proactive Testing methods that enhance conventional reactive testing risk identification, response.
- Monitoring, evaluating, adjusting, and reporting on risk activities, findings, and results.

Topics

- Nature and Importance of Risk
- Project Management Risks
- Conventional Risk Analysis
- Proactive Risk-Based Testing

Audience

This course has been designed for analysts, designers, programmers, testers, auditors, users, and managers, who rely on, plan, oversee, and/or carry out testing of software products.

Prerequisites

There are no prerequisites for this course.

Duration

One day

Course Outline

Risk Management Methods for Test Projects

Course Outline

I. Nature and Importance of Risk

Describe classic risk management concepts and elements. Introduce risk identification and quantification approaches and issues. Explain importance of distinguishing causes and effects.

A. Murphy's Law; O'Brien's Law
B. Relation of risk to software and testing
C. Classical risk management
D. Impact and probability risk elements
E. Classical risk mitigation, contingency plans
F. Costs, ease of detecting, controlling vs. harm
G. Quantifying qualitative risk, tricks and traps
H. Types, classifications of risks
I. Threats, vulnerabilities, triggers
J. Risk identification analysis methods
K. Business, functional, and operational risks
L. Direct and indirect forms of injury
M. Management and technical risks
N. Business Effect Risks
O. Effects vs. causes risk identification
P. Monitoring, controlling, reporting risks

II. Project Management Risks

Introduce the aspect of risk that is most written about, highlighting issues and relevance to risk analysis for testing. Critically evaluate common risk categorizations and checklists.

A. Causes vs. Effects Trap
B. Most frequently encountered risks
C. Changing requirements and scope creep
D. Lack of management support, priorities
E. Typically unrecognized interrelationships
F. How testing can reduce these risks
G. Project management risk relevance to testing
H. Software risks--or just poor management
I. Monitor and manage risks like a project
J. Evaluating Risk Factor Lists
K. Traditional checklists for project managers
L. Features/quality, resources, schedule risks
M. Implications for software QA/testing
N. SEI software risk taxonomy
O. Product engineering
P. Development environment
Q. Program constraints
R. Taxonomy-based questionnaire

III. Conventional Risk Analysis

Describes and gives practice with the two most common bases for identifying software risks and the two most common methods for evaluating and prioritizing those risks.

A. Testing riskier features, components more
B. Costs, additional testing resources vs. fixed
C. Rating each risk's impact and likelihood
D. Defining and rating feature risks
E. System, development practices risk checklists
F. Design analysis vs. prioritizing test cases
G. Common subjective risk judgment methods
H. Features risk ratings
I. Challenges of reliably rating risks
J. Setting objective criteria
K. Prioritization--rating vs. ranking
L. Defining, ranking component risks
M. Reporting, gaining agreement on risks
N. Features vs. component risks

IV. Proactive Risk-Based Testing

Introduce additional, more powerful approaches for identifying commonly-overlooked risks, starting with large, then medium-sized, then small risks. Apply efficient simplified prioritization drilling down on highest risks so they can be tested not only more but earlier.

A. Why typical risk-based testing is reactive
B. Proactive Testing Life Cycle
C. Structured model of test planning
D. Multiple levels, points of risk analysis
E. Prioritization demands knowing the choices
F. Proactive Master Test PlanningRisks
G. Identifying overlooked project-specific risks
H. Refocusing on tests that reduce the key risks
I. Letting testing drive development
J. Gaining user, manager, developer support
K. Detail Test Plan Risks
L. Test Design Specification Risks
M. Test Case Specification Risks
N. Proactive risk-based testing and checklists
O. Differentiating user and technical views
P. Risks of not testing some things
Q. Metrics to monitor effectiveness, efficiency
R. Reporting risk status, expected and actual
S. Categorizing actuals, improving over time