

## Table of Contents

CUSTOM EXERCISE GUIDELINES FOR ON SITE TRAINING.....	1
INTRODUCTION TO SOFTWARE TESTING.....	2
1 Day Tutorial .....	2
PROJECT MANAGEMENT FOR TEST PROJECTS.....	3
1 Day Tutorial .....	3
RISK BASED TEST MANAGEMENT FOR TESTERS.....	5
1 Day Tutorial .....	5
RISK-BASED ANALYSIS METHODOLOGIES FOR MANAGERS.....	6
1 Day Tutorial .....	6
PATH/SCENARIO DESIGN TECHNIQUES AND SELECTING THE MOST IMPORTANT TESTS.....	7
1 Day Tutorial .....	7
Path/Scenario Design Techniques Workshop .....	8
1+ Day Workshop.....	8
TEST CASE DESIGN TECHNIQUES FOR MANUAL AND AUTOMATED TESTING .....	9
1 Day Tutorial .....	9
Test Case Development and Tools Workshop.....	10
1+ Day Workshop.....	10
PRINCIPALS OF SOFTWARE TEST AUTOMATION.....	11
1 Day Tutorial .....	11
MANAGING THE TEST AUTOMATION EFFORT: STRATEGIES AND ARCHITECTURES.....	13
1 Day Tutorial .....	13
SECURITY TESTING FOR THE WEB AND HOW TO AUTOMATE IT.....	15
1 Day Tutorial .....	15

## CUSTOM EXERCISE GUIDELINES FOR ON SITE TRAINING

Any of the following materials can be used to customize the course exercises for your constituents. You will need to provide materials and non disclosure statements as early as possible before the onsite training date. The instructor will estimate what examples can be used within the scope of the day of training, and what examples will require add on workshops to work through.

- One or more examples of written requirements, specifications, and or change requests
- Sample business rule(s)
- One or more Use Cases, functional flow or process flows
- Sample data input screen(s) shots
- Sample file or data definitions (screen shot, excel spread sheet, or text file)
- Sample Schema, or data sets (screen shot, excel spread sheet, or text file)

## INTRODUCTION TO SOFTWARE TESTING

### 1 Day Tutorial

This tutorial is an introduction to current best practice test concepts and methods used by test professionals today. In addition to introducing and defining software testing terminology, the students get an introduction to best practice techniques.

#### Key topics:

- Test Planning practices
  - Software testing terminology
  - The testing lifecycle as a compliment to the development model
  - It all starts with requirements...
  - Positive and Negative testing
  - Static and Dynamic Testing
  - Testing Levels
  - Black box, clear box and gray box methods and techniques
  - Engineering Best Practices

#### Outline

- Introduction
- Life cycles
- Static and dynamic testing
- Test Methods
- Testing Practices
- Testing Levels
- Test development
  - Scenario development
  - Requirements to scenarios
  - Use Cases
  - Exercises throughout
- Test Metrics and SCurves

#### Students Learn

- Testing Terminology
  - Black Box, Gray Box and Clear Box Test Techniques, function testing and technical testing, etc
  - Positive and Negative Testing
- The testing life cycle, for traditional, RAD and Agile projects
- What methods to use for best test effectiveness
- Metrics that give testers and managers valuable information about the project

## PROJECT MANAGEMENT FOR TEST PROJECTS

### 1 Day Tutorial

#### Includes traditional and distance projects

Testing does not easily fit into the “do it and check it off” techniques of project management. If anything changes, testers must retest. If testers don't know exactly what changed, then they will want to retest everything; –so what can we do, uncheck all those “checks” and redo the schedule?

Traditional project management relies on formal documentation methods, which are both well planned and highly detailed. Today's short development cycles leave little time for planning and most of the details are dealt with as they appear. In addition, the challenges that arise in distance management are difficult to predict and require an adaptive, flexible approach to keep them on track.

This tutorial shows you how to use project management methods, tools and techniques to plan and manage your test effort, and be successful in the face of these challenges. Each topic is supported by one or more techniques, and numerous case studies are presented to illustrate events and methods.

### Students Learn

- How and where test projects fit into the overall development project
- What benefits a project management approach adds to the test effort, for local and remote test efforts
- How traditional project management techniques can be applied to the test effort and to distance management
- People planning, motivation, and control techniques for today's team driven environments
- What metrics to use on your test dashboard to control test progress, and predict difficulties; --and, keep project management well informed every step of the way

### Outline

- Foundations and Techniques
  - Effort estimation: How big is it?
    - Traditional (Planned) versus Agile efforts
    - Local versus remote (distance, off shore) efforts
- Test project planning
  - Requirements
    - Activity diagrams, Use cases
  - Risk Analysis and Reviews
    - Task identification and enumeration
- People
  - Identification of roles and responsibilities and assessing individuals' performance
  - Resource allocation (people, hardware, software, and facilities)
  - How teaming changes things in Project Management
    - Leading a team versus directing a group
  - How distance changes things
    - Schedule, reporting, and motivation tools
- Scheduling, tracking, and reporting techniques

- Scheduling
  - Identifying Critical Path flow: GANTT/PERT charts
  - Other project management techniques
- Project controls
  - Milestones, change management, risk mitigation
- Financial analysis and ROI
- Metric tracking and presentation
  - S-curves, velocity, Burn-downs
  - Bug metrics, risk mitigation
- Using automated project management tools

## RISK BASED TEST MANAGEMENT FOR TESTERS

### 1 Day Tutorial

Risk Analysis is a term that we hear regularly, but very few projects actually apply it as a Best Practice discipline. In order to qualify as “Best Practice”, the analysis and ranking process must meet the fundamental requirements for a reproducible and defensible practice. Further, Risk Management techniques are only useful if the test effort can manage its test assets effectively.

This seminar is designed to give the student a thorough grounding in Risk Analysis and Test Asset Management, as a “Best Practice” approach to software testing that will yield a significant and demonstrable ROI for the Test Effort. The methods and techniques taught in this seminar ensure the correct focus of the test effort, and provide testers and test managers with tools they can use to automate their test process and provide optimal test efficiency.

This tutorial is based on the book, *Software Testing Fundamentals*, authored by Marnie Hutcheson (Published by Wiley, 2003). It is a field guide aimed squarely at testers and test managers. It contains practical solutions, to real life testing challenges. It includes examples from both traditional and Agile/eXtreme projects. The seminar is filled with real world examples of how testers successfully managed their test efforts and demonstrated the value of testing.

This seminar contains working examples, and starter templates that you can take home and put to work.

Note: This seminar teaches the fundamental concepts that are the foundation of the one-day seminar: Identifying and Executing the Most Important Tests.

### Outline

- Introduction
- Fundamental methods and metrics for software and testing
- Test estimation and sizing
- The Test Inventory
- Tools and analysis techniques for test estimation and planning
- Identify and Enumerate Tests, fill out the Inventory, prioritize the tests, identify the most important tests, estimate the effort, track the test effort and evaluate it
- Analysis techniques for enumerating tests and test environments
- Risk analysis
- Establishing Test Priority
- Relating risk to test coverage
- Finishing the Test Estimation Effort
- Setting schedules, and goals
- Estimating the resources and time required to test the most important tests
- Negotiating for resources to conduct the test effort
- Adding value beyond the test effort

### Students Learn

- Building a test inventory, and automating it
- Establishing your own risk criteria, applying them to the test inventory and determining the risk associated with each test
- Picking the most important tests from the inventory
- Estimating the cost of running the most important tests, and how much this testing can save by finding bugs before the product is shipped

- Using worksheets to estimate the resources and time required to conduct a give test effort, and aid in negotiating for those resources

## RISK-BASED ANALYSIS METHODOLOGIES FOR MANAGERS

### 1 Day Tutorial

This seminar presents “Best Practice” risk-based methodology for planning, sizing, managing and successfully executing the test effort that is both defensible and reproducible. The focus of this seminar is on management of the software test effort using risk analysis as the basis. These methods can be applied to traditional test efforts and more importantly, they are useful in keeping up with and adding measurable value to Agile and eXtreme efforts. The seminar is filled with real world examples of how testers successfully managed their test efforts and demonstrated the value of testing.

#### Outline

- Risk analysis methodologies
- Risk Identification and classification
- Risk prioritization and ranking
- Calculating costs and probability using risk based techniques
- Risk reporting
- Monitoring and controlling risks
- Contingency planning and mitigation

#### Students Learn

- How to establish their own risk criteria, apply them to the test inventory and determine the risk associated with each test
- How to use the test inventory and risk analysis in test planning
- How to use worksheets to estimate the resources and time required to conduct a given test effort, and aid in negotiating for those resources
- How to estimate the cost of running the most important tests, and how much this testing can save by finding bugs before the product is shipped
- How to use risk-based selection techniques for test design
- How to report testing progress, status, and identify problem areas
- How to use the results of testing to improve subsequent efforts
- How to add value to the product and demonstrate that value to management

## **PATH/SCENARIO DESIGN TECHNIQUES AND SELECTING THE MOST IMPORTANT TESTS**

### **1 Day Tutorial**

This course applies techniques introduced in the Risk Based Test Management course. Students should be familiar with inventories, risk ranking techniques, test metrics, planning, sizing, and test estimation in order to get the most benefit from this seminar.

#### **Outline**

- Test Design
- Analytical Test Design
  - Path Analysis
  - Logic flows and Use Cases
- Risk analysis and Ranking
- Test reporting and measurement
- Managing Test Execution and Reporting Test Progress: Introduction to S-Curves
  - S-Curves – How they work
- Automation Techniques and Tools
  - Publishing Results
  - Status
  - Task assignments
  - High Function Web Sites
- Reporting and Tracking Bugs
- Analyzing Bug Statistics
- Tracking and Reporting the Test Effort
- Beyond the test cycle: Adding more value

#### **Students Learn**

- Risk-Based selection techniques for scenario and test case design
- How to select, build, and manage test scripts and test suites to maximize efficiency and reuse
- When and how to add automation to improve the performance of the test effort
- Techniques for tracking bugs and getting them removed
- How to add value to the product and demonstrate that value to management

## Path/Scenario Design Techniques Workshop

### 1+ Day Workshop

Available as an add on workshop with PATH/SCENARIO DESIGN USING LOGIC FLOW ANALYSIS AND SELECTING THE MOST IMPORTANT TESTS

#### Purpose

Provide mentoring as students apply new techniques to their real world problems

#### On Site Customers

Any of the following materials can be used in the workshop:

- Use Cases, functional flows, Storyboards
- The parent requirements, specifications, and or change requests, business rule(s), Use Cases, functional flow or process flows
- Sample file or data definitions (screen shot, excel spread sheet, or text file)

#### Students Gain

- Hand on experience applying the test scenario design techniques to your software systems
- Students exercise their in house standards and guidelines and identify difficulties
- Practice analyzing requirements, use cases, and other static designs
- Find solutions to problems raised by formal analysis

# TEST CASE DESIGN TECHNIQUES FOR MANUAL AND AUTOMATED TESTING

## 1 Day Tutorial

Verification and Validation is not something you do just at the end of development. V and V is part of each phase of development and deployment. --Or at least it should be. This course teaches you test design techniques you can use at each phase of development and test. Static test design techniques are used to design tests starting with the requirements, and moving through scenario development, functional design, detailed design, and into coding. We then explore dynamic test design techniques that take us from unit testing, through module integration, system testing and into User Acceptance and Beta testing. Exercises accompany each technique.

### Outline

- The testing lifecycle as a compliment to the development model
- Software testing terminology review
  - Positive and Negative testing
  - Static and Dynamic Testing
  - Testing Levels --It all starts with requirements
  - Black box, clear box and gray box methods and techniques
- Test Design Techniques
  - Black box Techniques
    - Turning Requirements into scenarios
      - Decision Tables (How many possible combinations exist?)
      - Modeling: Activity diagrams and Use cases (Introduction only)
    - Data Design techniques for Test Cases:
      - Boundary Value analysis
      - Equivalence class partitioning
      - All Pairs reductions (Introduction only)
      - Matrix data reductions (Introduction only)
  - Clear and Gray box techniques
    - Logic flow diagrams and basis paths (Introduction only)
    - Unit and Module Integration testing Statement and Branch coverage

### Students Learn

- How to apply each technique in a series of exercises
  - Black Box test design techniques
  - Modeling, Inspections and other forms of review
  - Clear Box test design techniques
  - Gray Box test design

## Test Case Development and Tools Workshop

### 1+ Day Workshop

Available as an add on workshop with TEST CASE DESIGN TECHNIQUES FOR MANUAL AND AUTOMATED TESTING

#### Purpose

Provide mentoring as students apply new techniques to their real world problems

#### On Site Customers

Any of the following materials can be used in the workshop:

- Sample data input screen(s) shots
- Sample file or data definitions (screen shot, excel spread sheet, or text file)
- Sample Schema, or data sets (screen shot, excel spread sheet, or text file)
- The parent requirements, specifications, and or change requests, business rule(s), Use Cases, functional flow or process flows

#### Students Gain

- Hand on experience applying the test scenario design techniques and the test case design techniques
- Students exercise their in house standards and guidelines and identify difficulties and
- Practice developing automation frameworks that work for you
- Practice Test Case Template Design and Development
- Practice Bug Reporting and measuring techniques

# PRINCIPALS OF SOFTWARE TEST AUTOMATION

## 1 Day Tutorial

Shorter and shorter code release cycles mean that there is more testing that should be done and less time to do it. Test automation is a requirement if the test effort is going to provide adequate test coverage for every release. But where should you start? There are many kinds of test automation, which ones are right for you? What should you automate? What tools should you use and once you have chosen, how do you get the job done? How do you measure what you got and demonstrate the value that your test automation has added? Even if you already do automated testing, are you getting the best return for your investment? How do you know?

This course prepares participants to answer all of these questions as it explains how to plan, build, execute, and measure (especially how to show ROI) automated software testing.

### Outline

- Introduction
  - Testing VS Test Automation –
  - What automation can gain and what it costs
- Tool evaluation and selection (both classes)
  - Picking the tool and support
  - Build or buy?
  - Management commitment
  - Product support and training
- Automation team roles
  - Internal Test Team
  - Education and new process support – Developing skills for test automation
  - Contractors and External Providers
  - Team Building
- Architectures – the arrangement of all test assets so they work well
  - Script attributes
  - **Scale, reuse, multiple versions, platform dependence/ independence**
  - Naming conventions
  - Test Asset Management
    - Library, warehouse, and results Inventories;
- Process automation and metric analysis
  - The Testing Dashboard
  - Bugs
  - Test Time:
  - Test Outcome versus Time Remaining
- Automation standards and guidelines
  - Build and apply your test framework
  - Naming conventions
  - Reusability, modularity, data matrix
- Planning the test automation process (both classes)
  - What to automate first

- Don't automate too much too soon – the system is still evolving
- Go for the quick wins –
- Estimating scope cost and timelines
- Running the automated test effort
  - Building the test environments
    - Data, architecture, security, network considerations
  - Running the tests, analyzing what happened, and reporting results
  - Making the ship-no-ship decision
  - Maintenance and re-use

### Students learn

- Factors necessary to succeed with test automation
- What you can automate, what it will cost, and what return on investment you can expect in terms of efficiency gains and bugs shipped
- What testing can be automated at each phase of software development
- How to set realistic goals
- What to automate first, last and not at all
- How to find and evaluate automation tools
- What kind of talents and activities need to be part of an automation test team
- How to develop and use a defensible consistent strategy for planning, developing, executing, reporting, and maintaining automated tests
- How to develop automated test scripts that are re-usable, extensible, flexible, and valuable
- Must have metrics for test automation, including ROI

## MANAGING THE TEST AUTOMATION EFFORT: STRATEGIES AND ARCHITECTURES

### 1 Day Tutorial

Test Automation promises a lot, but without proper planning and preparation, these promises do not come to fruition. An automation effort should start by analyzing the automation opportunities in its own development process, and selecting the best candidates. --Not just by buying a tool and trying to use it. But all too often, this is what happens. This course shows you how to develop a strategy and a plan to select the automation that will work in your environment and succeed with it.

This course is aimed at managers and test leads who must get more testing done in less time, and demonstrate their results. It addresses issues like: How do you know if it's working or not? How do you fix it if it's not working? How do you manage it? How do you measure your ROI?

### Students Learn

- Formulate a test automation strategy, select the tools to support it, plan the implementation, and measure the automation effort to demonstrate efficiency gains and ROI.
- Formulate and setup standards, procedures, metrics and controls to succeed with test automation
- Build and train test automation and performance test teams
- Create exemplary automated tests that are maintainable and reusable
- Detect problems early and resolve them while they are still small
- Get a positive ROI from test automation

### Outline

1. Introduction
  - Process
  - Program
  - Dashboard and metrics
2. Defining a test automation strategy and plan
  - Strategy
  - Plan
3. Test tool evaluation and selection
  - Categories of automated tools that can be used to aid testing
  - Automation tools for each step of the process
  - Build it vs. Buy it: automation strategies approaches
4. Developing skills and relevant test automation roles
  - Automation test standards, framework,
  - Versions, naming conventions and change management
5. Building a performance test team
  - Special considerations
    - Tools and black boxes
    - Physical and software environments and admin control
    - Trend analysis metrics

6. Selecting which tests to automate and converting from manual to automated testing
  - Test selection p 232
    - Most important tests
    - Breadth tests, general sampling
    - Most important functions
    - Most important Data
    - Test that are easiest to automate
    - Tests with the fastest payback
    - The tests that are run most often
  - Estimating scope cost and timelines
    - Starting Automation at the right time
    - Don't automate too much too soon – the system is still evolving
    - Go for the quick wins –
    - Soak and performance tests – quick to set up and impossible to do manually
  - Planning for the test environments –
    - Test data, architecture, security, networks, etc.
7. Process automation and metric analysis
  - Running the tests and reporting results
  - Making the ship-no-ship decision
  - Calculating ROI of automation
8. Long term maintenance considerations and re-use

## SECURITY TESTING FOR THE WEB AND HOW TO AUTOMATE IT

### 1 Day Tutorial

Would you recognize security vulnerability in your system or application if you saw it? Most testers and developers don't recognize security vulnerabilities when they see them. So it isn't surprising that most test groups don't perform any security specific testing at all. Security testing is easy to add to your test strategy, and it's a good candidate for automation.

This seminar brings you up to speed on today's security realities and shows you just how insecure our Internet/Intranet systems are. You will find out what hackers are doing, and how they are doing it. This seminar shows testers what to test for and how to test it to ensure their system is able to withstand current attack strategies and how to connect to ongoing information streams that will keep them on top of security vulnerabilities as they are discovered.

### Outline

- Introduction
  - The good news about security vulnerability'
  - What's at risk
  - Reporting
- What to test for and how to test it
  - The 4 main security vulnerabilities and how to design tests for each
  - Buffer Overflow
  - Cross Site Scripting (XSS)
  - SQL Injections
  - Encoded strings
  - Other Security Issues
- How to automate your security tests

### Students Learn

- How hackers break into our systems.
- What tests you need to perform to detect weaknesses in your Web server, Database Server, your Portals, and your applications.
- How to automate these tests and use them for more than just security testing
- How to report your results so that management understands the value of your effort
- Where to find security resources, BLOGS, and newsletters that will help you stay informed about current Security Issues