

TEST02 : Software Testing Techniques for User Acceptance and System Integration Testing

Description :

Offers an alternative perspective of software testing that bridges the 'testing gap', Builds confidence in those who need to develop UAT and SIT strategies and plans, Develops the skills required to write effective UAT and SIT test specifications and Provides a factual introduction to exploratory testing

Instructor :



Mr.Phil Robinson

CERTIFIED SCRUM MASTER

ที่มีประสบการณ์และอยู่ในอุตสาหกรรมดิจิทัลมาอย่างยาวนาน

Training Date : **16-17 June 2025**

fee : **15,000 ฿** (ราคายังไม่รวม Vat 7%)

Days & Duration : **2 Day(s) | 12 Hour(s)**

Time : **09:00:00 - 16:00:00**

Language : **English**

Venue : **Online**

Type : **Online**

Category : **Software Testing**

Objectives :

There are many training courses offering a solid grounding in the theory and practice of software testing but software testers attending these courses often struggle to apply what they have learnt when they return to the workplace. This gap between theory and practice is known as the 'testing gap'. This two-day course is designed to bridge the 'testing gap'. It is aimed at software testers who need a practical approach to User Acceptance Testing (UAT) and System Integration Testing (SIT) that can be applied to software projects in the real world.

The course is presented by an experienced software tester who shares a wealth of practical tips and recent project experience with participants during the course.

Course Features

- Presents a practical, non-academic approach to UAT and SIT
- Based on a proven framework for planning UAT and SIT
- Highlights the importance of exploratory testing to complement traditional specification based testing
- Detailed discussion of date based testing with practical solutions to common problems and issues
- Presented by an experienced tester who has been responsibility for the successful completion of UAT and SIT projects

Target Group :

- Those who want develop their careers as UAT and SIT test specialists such as Test Engineers, Tester Analysts, Quality Assurance Staff, Agile Teams, Business Analysts, Business Systems Analysts, Systems Analysts, Functional Analysts
- Those who need to develop plan and manage the UAT and SIT test effort such as Test Managers, Quality Assurance Staff, Project Managers, Program Managers Product Owners, Product Managers, Scrum Masters
- Those who who want to gain an understanding of UAT and SIT techniques such as Test Managers, Project Managers, Program Managers Software Engineers, User Representatives, Product Owners, Product Managers, Scrum Masters, Requirements Engineers, Requirements Analysts, Human Factors Specialists, Software Developers, Process Engineers, Software Engineering Process Group (SEPG) Staff, Methodologists, Process Improvement Staff

Benefits :

- Offers an alternative perspective of software testing that bridges the 'testing gap'
- Builds confidence in those who need to develop UAT and SIT strategies and plans
- Develops the skills required to write effective UAT and SIT test specifications
- Provides a factual introduction to exploratory testing

Course Outline :

Course Agenda

The Problem With Software Testing

- Problems With Confusing Terminology and Popular Myths
- The 'Testing Gap'

- Product vs. Project Life Cycles
- Views of Quality
- Software Testing Objectives

A Framework for Better Understanding Software Testing

- Definition of Test Levels
- Understanding the Integration Test Level
 - Component Integration
 - Feature Integration
- Identifying Test Items
- Understanding the Acceptance Test Level
- Testing is Not Black and White
 - Alternatives to the Traditional 'Black-Box' - 'White-Box' View
 - Defining the Test Basis
- Who Executes the Tests?
- Automating Test Execution
 - Two Compelling Drivers For Test Automation
 - When Is Manual Test Execution Impractical?
 - Comparing Automated and Manual Testing
- A Practical Software Testing Framework
- Test Case Definition

Planning User Acceptance Testing (UAT) and System Integration Testing (SIT)

- Using the Software Testing Framework for Test Planning
 - Typical UAT Strategy
 - Typical SIT Strategy
 - Developing Custom Strategies
- Risk-Based Testing
 - Potential Failures
 - Risk Analysis
 - Mitigation Strategy
- Expanding a Test Strategy into Full UAT and SIT Test Plans

Feature Testing

- What are Features?
- Features vs. Components
- What is Feature Testing?
 - Testing Features in Isolation
 - Testing Component Integration
 - Feature vs. Component Testing
- When is Feature Testing Performed?
 - The Role of Feature Testing During SIT
 - The Role of Feature Testing During UAT
- Modelling Features
 - Why Model Features?
 - How to Model Features
 - Natural Language
 - Use Case Diagrams
 - User Stories
 - Flow Diagrams

- Developing a Feature Test Specification
 - Testing Features in Isolation and Sequentially
 - Testing Between Boundaries and On Boundaries
 - Test To Fail (Negative Tests)
 - Testing Business Rules and Decision Outcomes
 - Testing Form Actions
- Feature Testing Tools and Test Automation
- Planning Feature Testing

End-to-End Testing

- What Is End-to-End Testing?
 - End to End Scenarios and Scenario Based Testing
 - Testing Feature Integration
 - End-to-End Verification of Business Rules
- When Is End-to-End Testing Performed?
 - The Role of End-to-End Testing During SIT
 - The Role of End-to-End Testing During UAT
- Modelling Business Object Life Cycles
- State Charts
 - Modelling States and Changes of State
 - Modelling Interactive and Batch Features
 - Modelling Decisions and Conditions
 - Modelling Batch Features
 - Dealing With Duplicate Transitions and Grouping Into Sub-States
- Developing an End-to-End Test Specification
 - Generating Scenarios
 - Scenario Steps
 - Client Stereotypes
- End-to-End Testing Tools and Test Automation
- Planning End-to-End Testing

Date-Based Testing

- What Is Date-Based Testing?
 - Events
 - Dates and Online Features
 - Dates and Batch Features
- System Clocks
- Changing the System Clock
 - Proxy Clock
 - Real Time Clock(s)
 - Date API
 - Date Utility
 - Other Issues
- Developing a Date-Based Test Specification
 - Date Partitions
 - Synchronising Scenarios
 - Developing a Test Schedule
- Date-Based Testing Tools and Test Automation
- Planning Date-Based Testing

Exploratory Testing

- What Is Exploratory Testing?
- Exploratory vs. Scripted Testing?
- When Is Exploratory Testing Performed?
- Exploratory Testing Skills
- Exploratory Testing Tools and Test Automation
- Planning Exploratory Testing

Managing the UAT and SIT Effort

- Tracking Progress
- Managing Issues
 - Tracking Product Quality
 - Triaging Issue Reports
 - Tracking Product Quality
- Test Completion Criteria
- Finding a Voice for Product Confidence

Payment Condition :

Payment can be made by:

1. Cash or Credit Card or Bank Cheque payable to "สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ" (a post-dated cheque is not accepted) on the first day of the service or within the last day of the service.
2. Account transfer and send the proof of the payment (the deposit slip) **via email** ttd@swpark.or.th
 - ธนาคารกรุงเทพ สาขาอุทยานวิทยาศาสตร์
Saving Account Number: 080-0-00001-0
Account Name: สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ
 - ธนาคารกรุงไทย สาขาตลาดไท
Saving Account Number: 152-1-32668-1
Account Name: สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

Notes:

- Withholding tax (3%) is exempt.
- Should you need to withdraw, you must send the notice of the withdrawal in writing no later than 7 working days before the commencement date. The cancellation less than 7 days will be subject to a fine of 40% of the fee.
- Software Park Thailand reserves the rights to cancel courses due to unforeseen circumstances.

Contact Person :

For more information, contact our course coordinator on:

เสกสรรค์ สังกุข (อีซู)

Mr. Seksun Sungsook

☎ : +662 583 9992 Ext. 81421

☎ : +6681 913 1828

✉ : seksun.sun@nstda.or.th



You are encouraged to use the course schedule as a guide to plan your training. The schedule is accessible at www.swpark.or.th for more information.

