# xTesting Requirements

# Essential Techniques for Agile and Waterfall Teams

## Course Overview

This course is for agile and waterfall teams who want to improve their ability to meet stakeholder expectations. It focuses on three ways this can be achieved by:

* checking that the solution requirements **will** satisfy stakeholder needs;
* checking that the team **is** implementing the solution accurately; and
* checking that the team **has** implemented a suitable solution.

The relationship between software requirements and testing is a central theme of the course. Concepts such as Requirements Based Testing, Test Driven Requirements and Behaviour Driven Development (BDD) are explored and fully defined.  
  
The course explains the different approaches that agile and waterfall teams can follow, and highlights where a common approach works for both. Participants will leave the course with a better understanding of how to capture stakeholder expectations and how to check that a software solution meets their expectations.  
  
Previous versions of this course were known as *Requirements Based Testing* and *Requirements Verification*.

## Course Features

* Clarifies the relationship between software requirements and testing.
* Offers proven, practical strategies for meeting stakeholder expectations.
* Suitable for graduates, staff moving into a new role, or experienced staff who need to update their skills, attend a refresher, or simply get some new ideas.
* Available as an instructor-led on-site course or as a facilitated virtual course with collaborative exercises based on a virtual canvas.

## Participant Benefits

* Understanding of the role of software requirements and how these relate to software testing.
* Understanding of how the agile life cycle resolves issues with the waterfall life cycle related to requirements and testing.
* Ability to identify and document software requirements using natural language, use cases, user stories and formal requirements statements.
* Able to apply practical strategies to ensure that solution requirements will satisfy stakeholder needs; the team is implementing the solution accurately; and that the team has implemented a suitable solution.
* Ability to develop a state chart model of a software solution and use the model to generate test scenarios.
* Hands-on experience gained from discussions, examples and short exercises based on a case study.

## Who Should Attend?

* Those who are currently members of agile or waterfall teams who want to deliver software solutions that meet the expectations of their stakeholders, such as Business Analysts, Software Developers, Test Analysts and Quality Assurance Staff
* Those who want further develop their careers as agile or waterfall team members, such as, Business Analysts, Software Developers, Test Analysts
* Those who need to plan and manage delivery of software solutions that meet stakeholder expectations such as Product Owners, Product Managers, Scrum Masters, Program Managers, Project Managers and Project Sponsors
* Those who need to approve Requirements Specifications, Test Plans and Test Specifications; as well as those who manage software development activities such as Product Owners, Product Managers, Program Managers, Project Managers, Test Managers, Engineering Managers and Project Sponsors
* Those who want to gain an understanding of requirements based and test driven testing, such as, Chief Information Officers (CIO), Business Architects, Enterprise Architects and Data Architects

## Course Duration

* 2 days full time

## Course Agenda

### Requirements Based Testing

* Requirements at different levels
* Objectives of software testing
  + Checking software meets its expectations
  + Identifying failures by exploring risks
* Building confidence in the reliability of the software
* Requirements Based Testing
  + Test Driven Requirements
  + Behaviour Driven Development (BDD)

### Software development life cycles

* The Waterfall life cycle
  + Traditional test levels
  + Requirements Validation - checking that the solution requirements will satisfy stakeholder needs
  + System Testing - checking that the team is implementing the solution accurately
  + Acceptance Testing - checking that the team has implemented a suitable solution
* The Agile life cycle
  + The Agile Testing Quadrants
  + Ensuring that the solution requirements will satisfy stakeholder needs
    - Incremental delivery
    - Three Amigos
    - Conversations
  + Ensuring that the team is implementing the solution accurately
    - Implementation immediately follows understanding
    - User story testing
  + Checking that the team has implemented a suitable solution
    - Scenario based testing
    - Exploratory testing
    - Quality attribute testing

### Software requirements

* Software features
  + Capabilities
  + Constraints
  + Components vs. features
  + Describing software features
    - Natural language
    - Use case diagrams
    - User stories
    - Formal requirements statements
  + Comparing approaches
* Describing detailed requirements

### Checking software meets its expectations

* Waterfall life cycle issues
  + Test planning
  + Iterative nature of testing
* Agile life cycle improvements
  + Incremental delivery
  + Three amigos
  + Conversations
  + Implementation immediately follows understanding
* Software testing
  + Test cases
  + Testing and the waterfall life cycle
    - System Testing
    - Acceptance Testing
  + Testing and the agile life cycle
    - User story testing
    - Checking quality attributes are fit for the intended purpose
    - Scenario based testing
    - Exploratory testing
  + Software testing summary
    - Components vs. features
    - Stand alone vs. end to end tests
* Additional techniques
  + The limitations of software testing
  + Workshops
  + Prototyping
  + Reviews
    - Types of review
    - Document reviews
    - Source code reviews
    - Reviews and "error seeding"
    - Reviews and test case design

### Model based testing

* Modelling business object life cycles
* State charts
  + States
  + Transitions
  + Mapping to features
  + Decisions and conditions
  + Sub-states
* Using state charts to check that the team is implementing the solution accurately
* Using state charts to check that the solution requirements will satisfy stakeholder needs

### Scenario based testing

* Comparing feature and test scenarios
* Using scenarios to check
  + The end to end integration of features is accurate
  + The end to end implementation of business rules is accurate
  + The team has implemented a suitable solution
  + The solution provides value
* Generating scenarios from state charts
  + Scenario steps
  + Scenario coverage

### Automated testing

* Managing change
  + "Freezing" the specification
  + Version control
  + Change and the Product Backlog
* Regression testing
  + Change and regression testing
    - Checking that the team is STILL implementing the solution accurately
    - Checking that the team has STILL implemented a suitable solution
  + Full regression test
  + Impact analysis and partial regression test
  + Test automation
  + Automated vs. manual testing
  + Test automation tools
    - Cucumber
    - Graph Walker

### Exploratory testing

* What is Exploratory Testing?
* Exploratory vs manual and automated testing
* Types of exploratory testing
* When is exploratory testing performed?
* Exploratory testing and scenarios