

Learning Objectives for Agile Testing

"Certification is the by-product; Learning is the product."

Agile Testing should:

- Compare and contrast agile testing with traditional software testing techniques and approaches. The learner should gain an understanding of the collaborative nature of agile testing and how business, development, and testing work together to produce high quality software.
- Give the learner both vocabulary and principles to be able to effectively perform agile testing as part of a whole-team concept – from initial test planning, through development iterations, to product release.
- Explore key agile testing techniques including early testing of software artifacts, test driven development (TDD), acceptance test driven development (ATDD), and exploratory testing. Approaches for automating appropriate types of testing should be discussed.

Although it takes months or years to learn how to be an effective agile tester, the contents set out below are designed to fit into three days of course materials and exercises, leading into more advanced learning in numerous testing topics. Instructors and self-learners are expected to include group activities of their own design and to incorporate their own, personal specialties that highlight particular aspects of agile testing. **[pre-requisites: Fundamentals of Agile Learning Objectives, understanding of common testing techniques and processes]**

The Agile Testing program is constructed around five major elements, with concepts and technique & application for each. An instructor or self-learner is welcome to add to these according to their own specialized interests.

1. **Agile Testing Mindset:** understand the history and mindset behind agile testing and how it impacts software quality during agile projects.
2. **Testing Techniques:** understand the types of testing and specific testing techniques typically used by business, development, and testing personnel during agile projects.
3. **Agile Testing Process:** understand the process by which agile teams plan, implement, perform, and report on testing activities.
4. **Test Automation:** understand why test automation is important, what tests are commonly automated, and who is typically involved in automation efforts.
5. **Beyond the Team :** understand how agile testers interact with management, integrate with other projects/groups, address policy and regulation, and operate in distributed environments.

How to read & use this document

This document is not a description of a course – it is a description of what a learner should encounter in courses, or study to learn, to gain a sound introduction to agile testing. These are the Fundamentals of agile testing.

These learning objectives are intended for those who are participating in agile teams and need to understand or perform software testing activities as part of their responsibilities. These roles may include: product owner, project manager, business analysts, software testers, and software programmers.

Watch, in each section, for *concept* elements, where a concept is introduced, and *technique & application* elements, where the learner gets to learn a technique, practice, or soak in a concept.

Duration

The time period set out to teach the agile testing learning objectives without test automation is two to three full days of instruction and activities. We encourage instructors to take more time to cover the material, to let the learner "soak" in the ideas for extra days, and to possibly split the learning objectives across multiple courses that may cover other material. We find that "soaking time" is key to conveying the mindset and approach of agile development and testing. Here are three ways you might extend the soaking time for your learners:

- Create a 4 or 5-day instead of 3-day course. Include more specific instruction on important agile testing techniques beyond those covered within this course in detail. Suggested topics include: test management, testing automation, load and performance testing, security testing, unit testing, usability testing, continuous integration.
- Split the material across multiple courses that cover other agile material (programming, project management, project planning)
- Separate material into classroom sessions and coaching sessions, with soaking on real projects and a coach to help steer into the agile approach along the way.

Depth of coverage

One difficulty that diligent instructors may encounter when preparing a 2-3 day course is balancing the need to cover the breadth of the agile testing topic while providing the details necessary to effectively educate course attendees. Wherever possible, group exercises should be used to drive home key learning objectives. Hands on use of tools and techniques are strongly encouraged as well.

1. Agile Testing Mindset

1.1. History of Agile Testing

1.1.1. Origins of agile testing

Many people who hear about agile testing for the first time assume that it was created as part of the agile movement. In actuality, much like agile itself, many of the agile testing techniques were espoused well before the Agile Manifesto was created.

The purpose of this LO is to anchor the ideas of agile testing in earlier work, giving the learners continuity from the past to the present.

1.1.2. Early thinking on agile and testers

Early agile teams and projects struggled with whether testing was a role or a person (or both) as agile began to become popular.

The purpose of this LO is to provide insight into the struggle associated with determining a tester's role within agile projects.

1.1.3. Agile testing vs. Traditional approaches

Agile testing is much different than testing performed during traditional software development approaches

The purpose of this LO is to provide insight into the major differences between agile testing and testing performed as part of traditional (phased-based) software development approaches in which testing is primarily performed by software testers who are often in their own organization and sometimes only involved late in the development lifecycle.

1.2. Mindset & Culture

1.2.1. Agile testing principals

The Principles behind the Agile Manifesto establish guiding principles for not only the agile movement but agile testing as a discipline.

The purpose of this LO is to help learners understand how the Agile Manifesto is realized within an agile testing process and approach.

1.2.2. Team-based quality

Quality is not "owned" by a particular role in agile. It is a property of software that the entire team must make sure is present before software is released to customers.

The purpose of this LO is to introduce the learner to the concept that quality is everyone's responsibility during agile projects and everyone is involved in software testing. Testers are often ideally suited to guide the team toward achieving its quality goals and its definition of "done" based on team definitions of the quality attributes for the product.

1.2.3. Building quality in

The role of a tester shifts in agile from that of quality gate keeper to a facilitator that supports the team through testing and critiquing the product.

The purpose of this LO is to shift the mindset of testers from that of an independent group responsible for gating the development process to a collaborative team member focused on improving the product and releasing value to the customer

1.2.4. Continuous improvement and feedback

Agile testing provides critical insights and feedback into the software process that can be used to drive team and quality improvements and assist the organization in making informed business decisions regarding software release. Automation of testing is a critical component in providing real-time feedback.

The purpose of this LO is to emphasize that agile testing is a critical feedback component when seeking to improve an agile process, reduce the cost of change.

1.2.5. Ingrain the agile testing mindset

Provide exercises or examples that solidify the mindset and culture of agile testing

The purpose of this LO is to have the learner experience situations in which the agile testing mindset is likely to be different, so the learner can internalize the difference experientially, not just in concept.

2. Testing Techniques

2.1. Categories of Testing

2.1.1. Agile testing quadrants or categories

Testing activities can be broken into various categories of testing based upon their purpose and value. Types of testing are often broken into categories that include: testing that supports project team development efforts, testing that looks at quality from a business perspective, testing that critiques the product, and testing that exercises the relationship between software and its deployment platform.

The purpose of this LO is to provide the Learner with a sound understanding of the purpose of various testing techniques so they can be applied appropriately and at the right time within an agile environment.

2.2. Subsystem Testing (aka Testing to Support Development)

2.2.1. Test-driven development (TDD)

TDD is a critical component of any agile development effort as it drives forward software design and implementation. Although TDD is fundamentally a design activity, it supports testing goals by producing automated unit and component tests that validate the quality of the code.

The purpose of this LO is to thoroughly understand the purpose and approach to successfully implementing TDD on agile projects.

2.2.2. Unit and Component Testing

Developer testing of individual software units and associated components is critical to detecting implementation defects within software. Unit and component tests are leveraged within TDD as well.

The purpose of this LO is to thoroughly understand the purpose and approach to successfully implementing unit and component testing on agile projects.

2.2.3. Other Developer Testing Techniques

Additional testing techniques beyond TDD for unit / component testing are often useful depending upon the circumstances and technologies associated with any particular project

The purpose of this LO is to provide the Learner with a sound understanding that it is up to the team to determine which testing techniques make the most sense to apply at all levels (e.g. component, integration, system).

2.3. Story and Feature Testing

2.3.1. Acceptance Test-driven Development (ATDD)

ATDD is a common technique for assuring that Stories are implemented in a manner that satisfies the exit criteria defined for Story completion. It is often use as a

technique to test Stories but in actuality includes the testing of key business processes and non-functional requirements as well.

The purpose of this LO is to thoroughly understand the purpose and approach to successfully implementing Acceptance Test-driven Development (ATDD) on agile projects.

2.3.2. Behavior-driven Development (BDD)

BDD is an alternative approach to ATDD that is sometimes used to test Stories, Business Process, and non-functional Requirements based upon an understanding of user behavior

The purpose of this LO is to thoroughly understand the purpose and approach to successfully implementing Behavior-driven Development on agile projects.

2.3.3. User Story Testing

Testing of User Stories is critical to successful development of software within an agile project. This testing is often performed using the techniques above but can be done in other ways as is appropriate or necessary.

The purpose of this LO is to thoroughly understand how User Stories are tested during software development.

2.3.4. Other Feature Testing Techniques

While the above techniques are the most common, there are a variety of other testing techniques that can be applied to test software features.

The purpose of this LO is to provide the Learner with an understanding that while particular testing techniques are more common than others, there are other techniques available that might make more sense in any particular situation.

2.4. Integration and System Testing

2.4.1. Structured Testing

Structured testing provides a mechanism for additional testing to be performed on Stories or Business Processes based upon a team's knowledge about the products structure and elements.

The purpose of this LO is to provide the Learner with a sound understanding of Structured Testing techniques and approaches and how they are best applied to an agile project.

2.4.2. Exploratory Testing

Exploratory testing provides a mechanism for additional testing to be performed on Stories or Business Processes based upon a tester's intuition and knowledge about the product.

The purpose of this LO is to provide the Learner with a sound understanding of Exploratory Testing techniques and approaches and how they are best applied to an agile project.

2.4.3. Usability Testing

Usability testing is a formalized testing process for assessing the usability of a software application.

The purpose of this LO is to provide the Learner with a sound understanding of Usability Testing techniques and approaches and how they are best applied to an agile project.

2.4.4. Other Validation Techniques

Beyond Structured Testing, Exploratory Testing and Usability, there are other software testing techniques that can be used to validate the software meets customer needs. This include testing of Stories and Business Process descriptions during planning.

The purpose of this LO is to provide the Learner with an understand that while Structured, Exploratory, and Usability are common test techniques for validating customer needs, other technique (e.g. inspections, reviews, beta testing) can and will apply given any specific situation. In particular, static review / testing of stories and business processes should be performed during initial planning to assure Stories are defined properly and each story / feature fits within the product set.

2.5. Testing Non-functional Requirements

2.5.1. Load and Performance Testing

Load and Performance Testing is important whenever software must meet response time, availability, or computational speed requirements.

The purpose of this LO is to provide the Learner with an understanding of when and how Load and Performance Testing is best applied within an agile process.

2.5.2. Security Testing

Security testing is important whenever a software vulnerability has significant business consequences.

The purpose of this LO is to provide the Learner with an understand of how security testing is performed during an agile process and when the risks of software compromise are great enough to warrant security testing.

2.5.3. Other Non-functional Testing Techniques

There are other 'ilities' that are sometimes important give software's purpose and requirements.

The purpose of this LO is to provide the Learner with an overview of other relevant non-functional requirements (e.g. availability, robustness, fault-tolerance, safety) that are sometimes important enough to test for during an agile project.

3. Agile Testing Process

3.1. Roles and Responsibilities

3.1.1. Team-based Testing Approach

Testing during an agile project is team-oriented wherein it is common for every member of the team to provide some level of testing support.

The purpose of this LO is to provide the Learner with an understanding that within an agile project, the entire project team is responsible for test plans, test design, test cases, test automation, and test reporting.

3.1.2. Typical Product Owner Role in Testing.

Product owners typically provide guidance on acceptance criteria and sometimes create test cases that provide examples of what Stories are intended to accomplish.

The purpose of this LO is to provide the Learner with an understanding of the common test activities that a product owner is involved with during an agile project.

3.1.3. Typical Programmer Role in Testing

Software programmers typically build, automate, and run a variety of tests at a variety of levels as part of their development process. TDD and ATDD leverages this testing to improve design and development.

The purpose of this LO is to provide the Learner with an understanding of the role software programmer's play within an agile testing process.

3.1.4. Typical Tester Role in Testing

Software testers typically work hand-in-hand with the product owner and programmers to plan, execute, and report on the testing that is performed at all levels. Testers often are responsible for creating User Story and business process tests cases and performing exploratory testing. Testers participate in and may develop automated tests along with programmers or a dedicated test automation team.

The purpose of this LO is to provide the Learner with an understanding of the role software tester's play within an agile testing process

3.1.5. Typical Customer Role in Testing

Customers typically support testing through definition of Stories and associated acceptance criteria and by performing user acceptance testing (UAT) prior to release or when it makes sense after a series of iterations.

The purpose of this LO is to provide the Learner with an understanding of the role customer's play within an agile testing process.

3.2. Test Strategy and Planning

3.2.1. During Release/Theme Planning

Lightweight planning is typically part of the release/theme planning done prior to associated iterations.

The purpose of this LO is to provide the Learner with an understanding of how lightweight test strategy and planning is performed during release/theme planning and how decisions are made regarding what type of test documentation is needed and how much is enough.

3.2.2. During Iteration Planning / Kickoff

Test planning at iteration kickoff focuses on detailing acceptance criteria and examples for Stories.

The purpose of this LO is to provide the Learner with an understanding of how plans and tests are developed prior to implementation.

3.2.3. Light weight test plan documentation

Test planning in agile is different than in traditional development approaches as the goal is to provide the least amount of documentation needed to get the job done.

The purpose of this LO is to provide the Learner with an understand of how to determine the amount of test documentation necessary for a given environment or situation.

3.2.4. Story Estimation

The scope (size) of stories should be estimated to provide information that an agile team achieves its target velocity while maintaining a sustainable development pace.

The purpose of this LO is to provide the Learner with an understanding of how software testing supports the story estimation process to assure that estimates include adequate time for testing as part of story development. Also the development of test infrastructure and supporting automation.

3.2.5. Acceptance Criteria

Story acceptance criteria is critical to defining “done” for stories and provide guidance for the types of tests to develop during ATDD / story testing.

The purpose of this LO is to provide the Learner with an understanding of how story acceptance criteria is created and how that criteria gets translated into acceptance tests.

3.3. Testing During Iterations

3.3.1. Developer / Tester Interactions

In agile, developers and testers work hand-in-hand to build and test stories.

The purpose of this LO is to describe the interactions and process by which developers and tests build and test software during iterations.

3.3.2. Tester / Customer Interactions

In agile, testers and customers work hand-in-hand to make sure acceptance criteria and acceptance tests are complete and appropriate for the stories that will be implemented and tested during each iteration.

The purpose of this LO is to describe the interactions and process by which testers and customers interact to define acceptance criteria and appropriate acceptance tests during each iteration.

3.3.3. Defect tracking and management

The amount of defect tracking that is performed during an agile project depends upon what works best for the team.

The purpose of this LO is to describe the key tradeoffs for determining which defects to track and which to rely upon team communication to correct without tracking.

3.3.4. Results reporting

Test reporting during agile projects depends upon what works best for the team.

The purpose of this LO is to describe the key tradeoffs between documented test results and team communication of those results.

3.3.4. Test metrics

Metrics collected to support test completeness and release readiness decisions

The purpose of this LO is to inform the Learner as to which metrics make sense to collect and report on for both test completeness and release readiness within an agile project.

3.3.5. Regression tests

Automated regression tests are essential to reducing the cost of change and providing real-time feedback during the development process.

The purpose of this LO is to provide the Learner with an understand of how to best leverage tests that have been automated during development within future iterations and releases.

3.3.6. Iteration wrap-up

Wrap-up activities during an iteration include a product demo, retrospective, and sometimes a User Acceptance Test.

The purpose of this LO is to provide the Learner with an understanding of the role that software testers play during iteration wrap-up activities.

3.4. Testing During Releases

3.4.1. Definition of a release / end game

A release process (aka “end game”) is performed whenever a decision has been made to release software to customer(s).

The purpose of this LO is to provide the Learner with an understanding of how a release decision is made and what testing activities are typically part of the release process.

3.4.2. User Acceptance Testing (UAT)

User Acceptance Testing is used within agile to gain customer feedback on a working piece of software before its release.

The purpose of this LO is to provide the Learner with a sound understanding of user acceptance testing (UAT) techniques and approaches and how they are best applied to an agile project.

3.4.3. Staging Environment Testing

A staging environment is often established to support testing of an application within an environment that closely resembles production.

The purpose of this LO is to provide the Learner with an understanding of the types of testing that is performed during the release process on a staging environment.

3.4.4. Post-release Testing

Testing after software release typically consists of testing “hot fixes” for critical defects identified in the field and on-going testing of bug fixes not fixed prior to release.

The purpose of this LO is to provide the Learner with an understanding of the types of testing that is performed post release and how continuous testing supports a continuous release process.

3.5. Test Environments and Infrastructure

3.5.1. Typical environments for test

Multiple environments are often necessary to support testing activities during iterations and the release process

The purpose of this LO is to provide the Learner with an understanding of the typical test environments that must be setup and maintained to support testing activities during iterations and releases.

3.5.2. Virtualization

Virtualization provides a mechanism (often automated) to support effective test environment setup, test execution, and test environment teardown during a testing process.

The purpose of this LO is to describe how virtualization can support an automated, effective testing process.

3.5.3. Testing the proper build

As builds are constantly being generated during an agile process, testing the proper build is critical to an effective testing process

The purpose of this LO is to discuss the best practices associated with choosing a build for test and keeping development and testing in synch during the process.

3.5.4. Test data management

Effective test data management is essential to all aspects of agile testing as the ability to select appropriate test data, set this data up, perform testing upon it, and reset any resulting changes is critical to an effective testing process.

The purpose of this LO is to discuss the best practices associated with managing test data during an agile process.

3.6. Working on Distributed Teams

3.6.1. Distributed Team Communication

Distributed teams are a fact of life in most organizations and must be dealt with to make agile testing initiative successful.

The purpose of this LO is to provide the Learner with an understanding of how communication can be most effective on distributed teams.

3.6.2. Distributed Team Coordination

Distributed teams are a fact of life in most organizations and must be dealt with to make agile testing initiative successful.

The purpose of this LO is to provide the Learner with an understanding of how testing activities can be coordinated when the team is distributed.

4. Test Automation

4.1. Test Automation Strategy

4.1.1. Automation Pyramid

Automated testing can be performed at various levels within a software application. An automation pyramid or structure describes these various levels and discusses the approach and likelihood of automating tests within each of them.

The purpose of this LO is to provide the Learner with a comprehensive understanding of the various types of testing that can be automated and how decisions get made regarding what to automate during an agile project.

4.1.2. Planning for Automation

Defining the approach, tools and timings for automation through the project.

The purpose of this LO is to provide the Learner with general knowledge regarding how to plan out an agile test automation effort: what to automate, when to automate, who will be doing automation and who will be using automation during the project.

4.1.3. Automation Frameworks

Frameworks provide test infrastructure for automating various types and levels of tests.

The purpose of this LO is to provide the Learner with general knowledge regarding various types of test automation frameworks so they can effectively choose which frameworks make sense for their particular application based on testing requirements and timelines.

4.1.4. Selecting Tests for Automation

It is typically infeasible and not cost effective to automate all tests that are created and/or run.

The purpose of this LO is to provide the Learner with an understanding of how to decide which tests that get created and/or run during an agile project should be automated vs. tested manually.

4.1.5. Supporting process

Test automation is performed at various points during agile project iterations and release cycles. When test automation is performed and for what purpose must be understood.

The purpose of this LO is to provide the Learner with an understanding of when it makes sense to automate tests during development iterations and release cycles.

4.2. Continuous Integration

4.2.1. Automated Builds

Incremental automated builds are essential for reducing the cost of change and providing rapid feedback on quality to the development team.

The purpose of this LO is to provide the Learner with an understanding of how automated builds are set up to support a continuous testing process.

4.1.2. Automated Test Cycles

Integrating automated testing into a build environment assures that software changes are tested early and often during the development process.

The purpose of this LO is to provide the Learner with tips and techniques for integrating automated tests into an incremental build process such that software is validated during the entire development process.

4.1.3. Code Analysis / Metrics

Code analysis and quality metrics can provide additional insights into an applications quality and release readiness of the software.

The purpose of this LO is to provide the Learner with an understanding of code analysis and code metrics for measuring the quality of software applications.

4.3. Automating Unit/Component Testing

4.3.1. Unit and component testing frameworks (supports TDD as well)

Frameworks exist that provide support for developing and running automated unit and component tests during software iterations.

The purpose of this LO is to discuss common unit and component test frameworks and how they can be used to effectively automate tests.

4.3.2. Demonstration

Demonstrate an existing unit and component test framework.

The purpose of this LO is to give the Learner hands-on experience with a commonly used unit and component test framework to further their understanding of how such frameworks are used to support development activities.

4.3.3. Hands-on Unit Testing

Specify and automate unit tests

The purpose of this LO is to give the Learner hands-on experience specifying and implementing automated unit tests within a common commercial or open source unit testing framework.

4.4. Automating Story and Feature Testing

4.4.1. ATDD and BDD testing frameworks

Frameworks exist that provide support for developing and running automated story and other types of feature tests during software iterations.

The purpose of this LO is to give the Learner hands-on experience with a commonly used story or feature test framework to further their understanding of how such frameworks are used to support development and testing activities

4.4.2. UI testing tools

Tools exist for exercising software through its user interface to test features and combinations of features.

The purpose of this LO is to give the Learner hands-on experience with a commonly used UI test tool to further their understanding of how testing can be performed through software's user interface.

4.4.3. Demonstration

Demonstrate existing story and UI testing tools.

The purpose of this LO is to give the Learner hands-on experience with commonly used story and feature testing tools.

4.4.4. Hands-on Story and Feature Testing

Specify and automate Story and/or Feature tests

The purpose of this LO is to give the Learner hands-on experience specifying and implementing automated Story and/or Feature tests within a common commercial or open source testing framework.

4.5. Automation Support for Integration and System Testing

4.5.1. Database setup and teardown

Effective test automation often includes automating the manual processes associated with setting up and resetting test data.

The purpose of this LO is to provide the Learner with an understanding of how test automation can be used to automate the setup and teardown of test data sets.

4.5.2. Tools to support Exploratory Testing

While Exploratory Testing is inherently a manual testing process, tools can be leveraged to assist in the testing process.

The purpose of this LO is to provide the Learner with an understanding of how tools can assist an Exploratory Testing process.

4.5.3. Demonstration

Demonstrate the use of tools to support Exploratory Testing

The purpose of this LO is to provide the Learner with a hands-on understand of how tools can support Exploratory Testing.

4.5.3. Hands-on Exploratory Testing

Specify and perform Exploratory testing

The purpose of this LO is to give the Learner hands-on experience specifying and performing Exploratory Testing using existing commercial or open source tools to support the process.

4.6. Automating Non-functional Testing

4.6.1. Load and Performance tools

Tools exist to support both load testing and performance analysis of software.

The purpose of this LO is to provide the Learner with an understanding of how load and performance testing is performed during an agile project.

4.6.2. Security analysis tools

Tools exist to support security testing of applications.

The purpose of this LO is to provide the Learner with an understanding of how security analysis is performed on software and the ways in which tools can support this process.

4.6.3. Tools for performing other types of non-functional testing

Tools exist to support other types of non-functional testing

The purpose of this LO is to provide the Learner with an understanding of other tools that are available for testing other non-functional requirements beyond load and performance, and security.

4.6.4. Demos

Demonstrate the use of tools to support Non-functional Testing

The purpose of this LO is to provide the Learner with a hands-on understanding of how tools can support non-functional testing.

5. Beyond the Team

5.1. Cultural Challenges

5.1.1. Adjusting to a Whole Team Philosophy

Working in a whole team approach takes a completely different mindset than working in a traditional process.

The purpose of this LO is to give the Learner an appreciation for the differences between working as a “whole team” and working in a traditional team structure and the ways in which this transition can be made.

5.1.2. Comfort Level with Ambiguity

Ambiguity surrounding requirements and release definition is something many individuals new to agile are not used to.

The purpose of this LO is to provide the Learner with an understanding of the challenges associated with the ambiguity that a learning-driven process has up front and how to best cope with this change.

5.1.3. Management Discomfort around Planning

Management is sometimes not comfortable with the lack of definition for the scope of a planned release.

The purpose of this LO is to provide the Learner with an understanding of the reasons why management is not comfortable with an agile project planning process and how to overcome that discomfort over time.

5.1.4. Resistance to Change

Any type of change is often difficult for some people to overcome.

The purpose of this LO is to provide the Learner with an understanding that moving to agile is a change process and will likely result in some resistance to change from various parts of the organization.

5.2. Organizational Structure

5.2.1. Project vs. Matrix

A common recommendation is for organizations that are embracing agile to organize their staff around products / projects instead of functional responsibilities.

The purpose of this LO is to provide the Learner with the pros and cons of various organizational structures when implementing agile.

5.2.2. Role of Test Managers in Agile

When an organization has decided to organize around products / projects, test managers are often left to wonder what their role within this new structure will be.

The purpose of this LO is to provide the Learner with an understanding of the role that a test manager plays within an organization that is moving toward a product / project based staff organizational model.

5.3. Addressing Governance and Policy Requirements

5.3.1. Satisfying Corporate Policies

Organizations often have a variety of corporate policies and procedures that must be adhered to during software development activities.

The purpose of this LO is to provide the Learner with an understanding of the various corporate governance, policy, and procedures that are often required during software development activities.

5.3.2. Documentation for Regulatory Requirements

Regulatory compliance often demands particular documentation be developed as part of any software development process.

The purpose of this LO is to provide the Learner with an understanding of how regulatory requirements can still be fulfilled when following an agile development process.

5.4. Integrating with other Products or External Technologies

5.4.1. Release Trains

A release train is a process for coordinating agile and IT projects.

The purpose of this LO is to provide the Learner with an understanding of how to coordinate agile projects and associated testing activities when the project must interact / interface with other projects and IT technology release schedules.

5.4.2. Shifting Priorities based upon External Interfaces

Dependencies upon external systems and products must be considered during agile projects.

The purpose of this LO is to provide the Learner with an approach to dealing with external dependencies during an agile project.