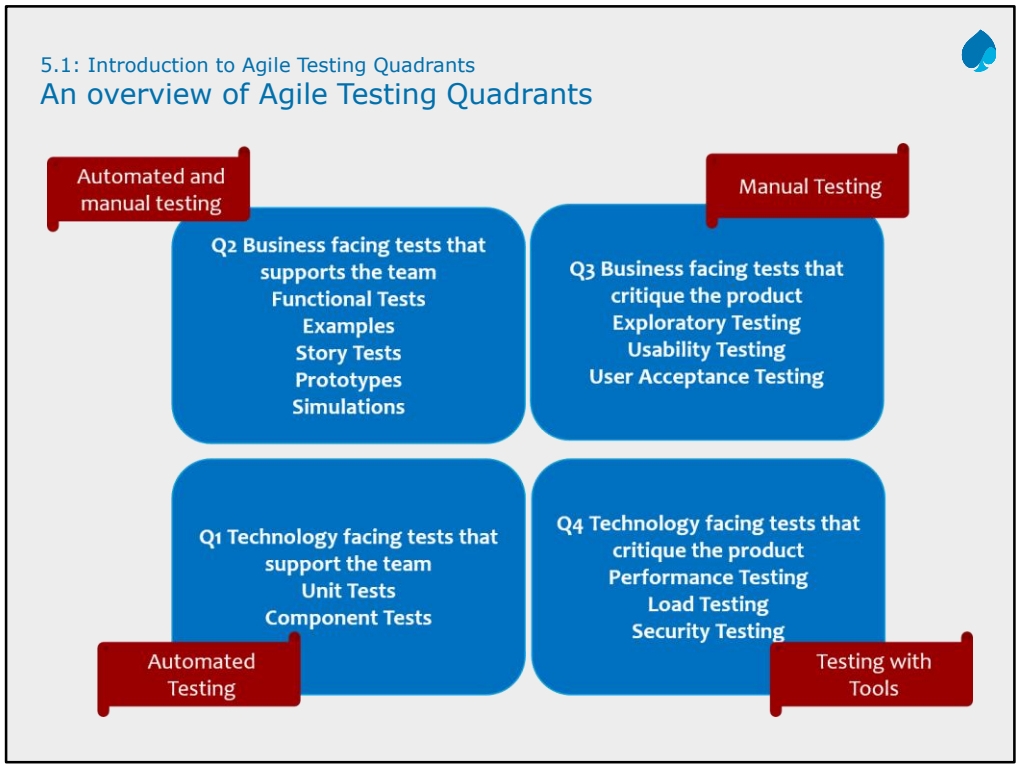

Introduction to Agile
Lesson 5: Agile Testing Quadrants and Agile Test Planning
Capgemini

Lesson Objectives

- An overview of Agile Testing Quadrants
- Agile Testing Quadrant 1, 2 & 3 Goals
- Agile Testing Quadrant 1, 2 & 3 Toolkit
- Test Planning in Agile Testing



Why do we test?

The answer might seem obvious, but in fact, it's pretty complex. We test for a lot of reasons: to find bugs, to make sure the code is reliable, and sometimes just to see if the code is usable. We do different types of testing to accomplish different goals. Software product quality has many components.

The Agile Testing Quadrants matrix helps testers ensure that they have considered all of the different types of tests that are needed in order to deliver value.

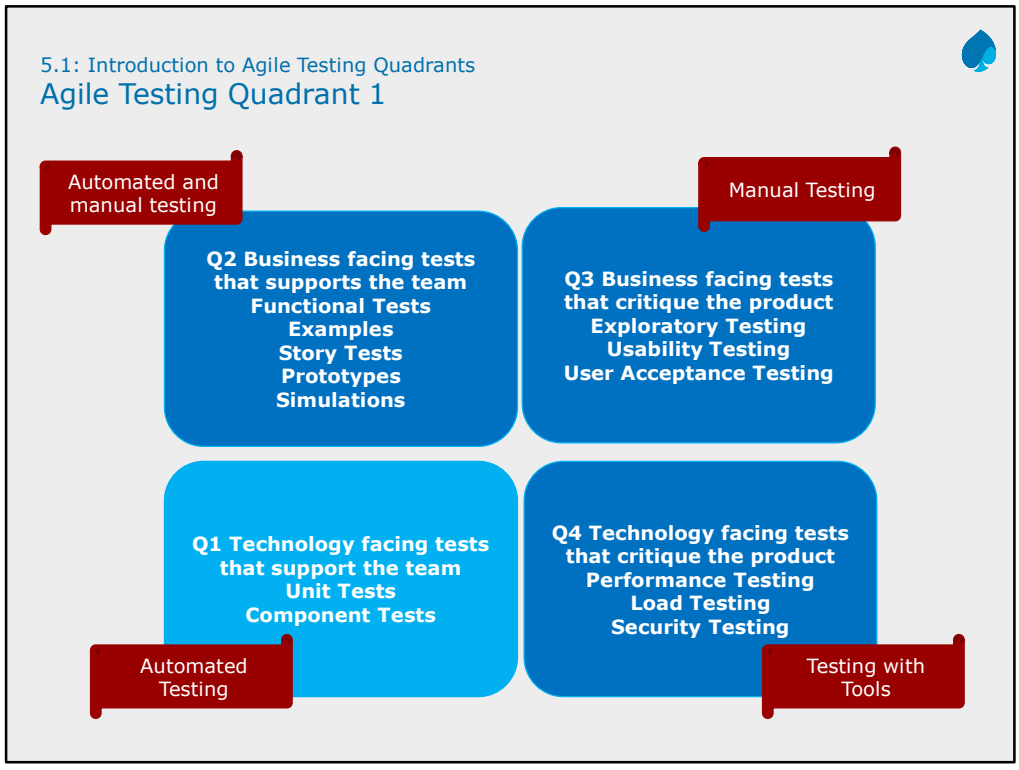
The above diagram of the agile testing quadrants that shows how each of the four quadrants reflects the different reasons we test. On one axis, we divide the matrix into tests that support the team and tests that critique the product. The other axis divides them into business-facing and technology-facing tests.

The order in which the quadrants are numbered is nothing to do with the sequence in which these different types of testing are performed. The time at which various types of tests are performed is dependent on the project risk factor, the product expectations, the type of project i.e. whether the team is working on legacy code or on a greenfield project and most importantly when resources are available to do the testing.

5.1: Introduction to Agile Testing Quadrants

An overview of Agile Testing Quadrants (Cont.)

- Agile development is supported by a bundle of concrete practices, covering areas like requirements, design, modeling, coding, testing, project management, process, quality etc.
- Agile expert Lisa Crispin explains the four Agile testing quadrants and how they can guide managers and development teams in creating a test strategy
- Undoubtedly Agile Testing is gaining its popularity in the testing world with its ability to deliver high quality products that keep the customers satisfied
- Project Managers and teams new to Agile development, the idea of planning and executing all the testing activities within short iterations and release cycles is intimidating
- It is imperative to understand the Agile testing quadrants and how it can help you perform Agile testing better
- Another advantage of the quadrants is that they can explain the whole testing in common language that is easy to understand


5.1: Introduction to Agile Testing Quadrants
Agile Testing Quadrant 1
Automated and manual testing
Manual Testing
Q2 Business facing tests that supports the team
Functional Tests
Examples
Story Tests
Prototypes
Simulations
Q3 Business facing tests that critique the product
Exploratory Testing
Usability Testing
User Acceptance Testing
Q1 Technology facing tests that support the team
Unit Tests
Component Tests
Q4 Technology facing tests that critique the product
Performance Testing
Load Testing
Security Testing
Automated Testing
Testing with Tools

5.1: Introduction to Agile Testing Quadrants

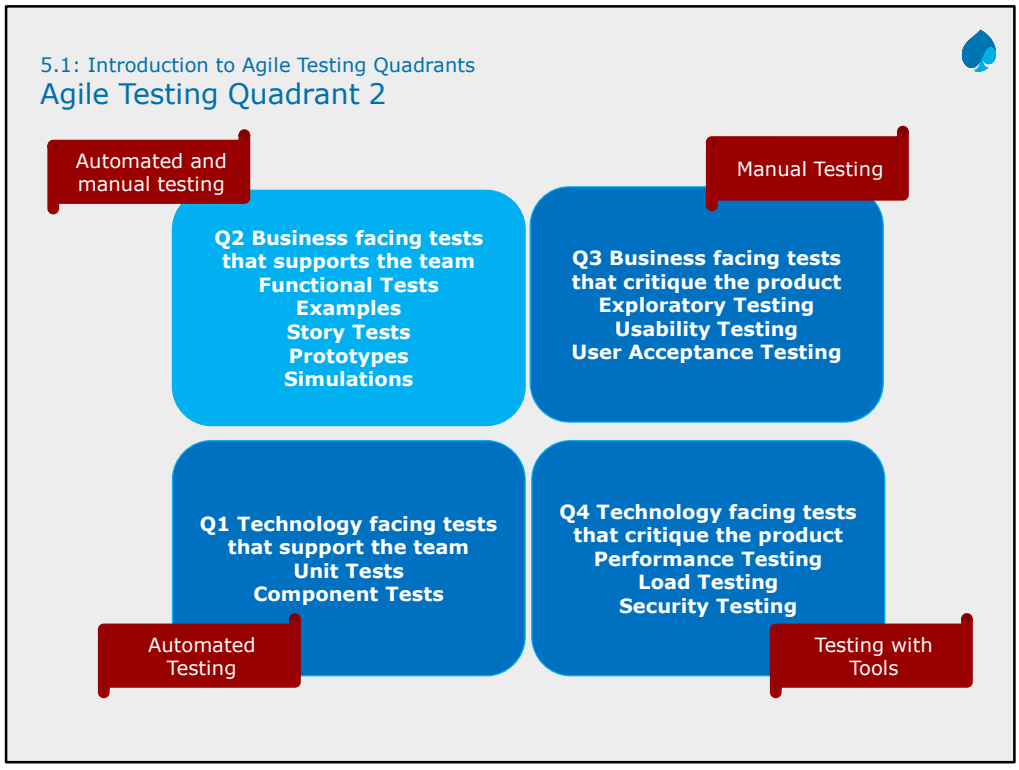
Agile Testing Quadrant 1 Goals

- The lower left quadrant represents test-driven development, which is a core agile development practice
- Unit testing verifies that the functionality of a smallest unit of the system, such as an object or method
- Component testing verifies the behavior of a larger part of the system, such as a group of classes that provide some service
- Both types of tests are usually automated with a member of the xUnit family of test automation tools
- We can refer to these tests as programmer tests, developer facing tests, or technology-facing tests
- They enable the programmers to measure what Kent Beck has called the internal quality of their code

5.1: Introduction to Agile Testing Quadrants

Agile Testing Quadrant 1 Toolkit

- Source code management
  - Version control
- Integrated development environment (IDE)
  - Compile, debug, build GUI, refactor : Eclipse, IntelliJ Idea, NetBeans
- Build tools
  - CruiseControl, Hudson. TeamCity
- Unit test tools
  - xUnit ,Mocking tools


5.1: Introduction to Agile Testing Quadrants
Agile Testing Quadrant 2
Automated and manual testing
Manual Testing
Q2 Business facing tests that supports the team
Functional Tests
Examples
Story Tests
Prototypes
Simulations
Q3 Business facing tests that critique the product
Exploratory Testing
Usability Testing
User Acceptance Testing
Q1 Technology facing tests that support the team
Unit Tests
Component Tests
Q4 Technology facing tests that critique the product
Performance Testing
Load Testing
Security Testing
Automated Testing
Testing with Tools

5.1: Introduction to Agile Testing Quadrants

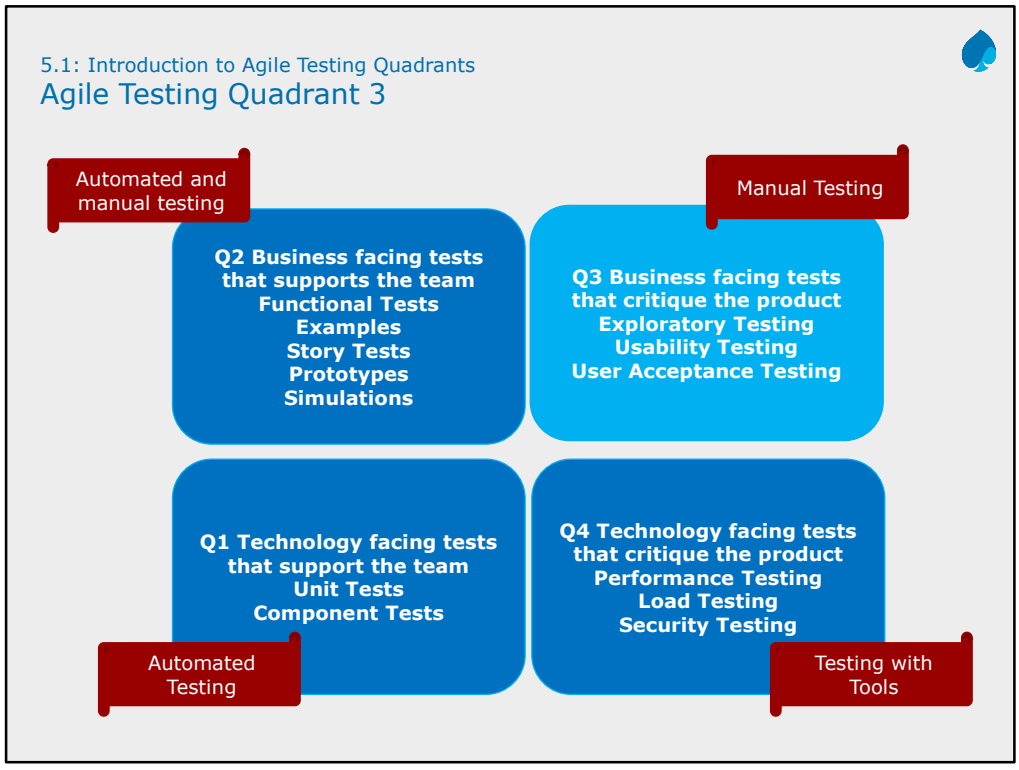
Agile Testing Quadrant 2 Goals

- This quadrant focuses on eliciting the requirements
- The tests in Quadrant 2 support the work of the development team as well, but at a higher level
- These business-facing tests, also called customer-facing tests and customer tests, define external quality and the features that the customers want
- With agile development, these tests are derived from examples provided by the customer team
- They describe the details of each story. Business-facing tests run at a functional level, each one verifying a business satisfaction condition
- The objective of this quadrant is to obtain enough requirements so that coding can commence without any hiccups

5.1: Introduction to Agile Testing Quadrants

Agile Testing Quadrant 2 Toolkit

- Checklists
  - Mind Maps
  - Brainstorming
  - Mockups / Paper prototypes
  - Flow Diagrams
  - Whiteboards (Physical and Virtual)
  - Fit/FitNesse
- BDD frameworks
  - Cucumber, easyB, nbehave, rspec
- GUI test tools/libraries/frameworks
  - Selenium
  - Robot Framework
  - SWAT
  - QTP


5.1: Introduction to Agile Testing Quadrants
Agile Testing Quadrant 3
Automated and manual testing
Manual Testing
Q2 Business facing tests that supports the team
Functional Tests
Examples
Story Tests
Prototypes
Simulations
Q3 Business facing tests that critique the product
Exploratory Testing
Usability Testing
User Acceptance Testing
Q1 Technology facing tests that support the team
Unit Tests
Component Tests
Q4 Technology facing tests that critique the product
Performance Testing
Load Testing
Security Testing
Automated Testing
Testing with Tools

5.1: Introduction to Agile Testing Quadrants

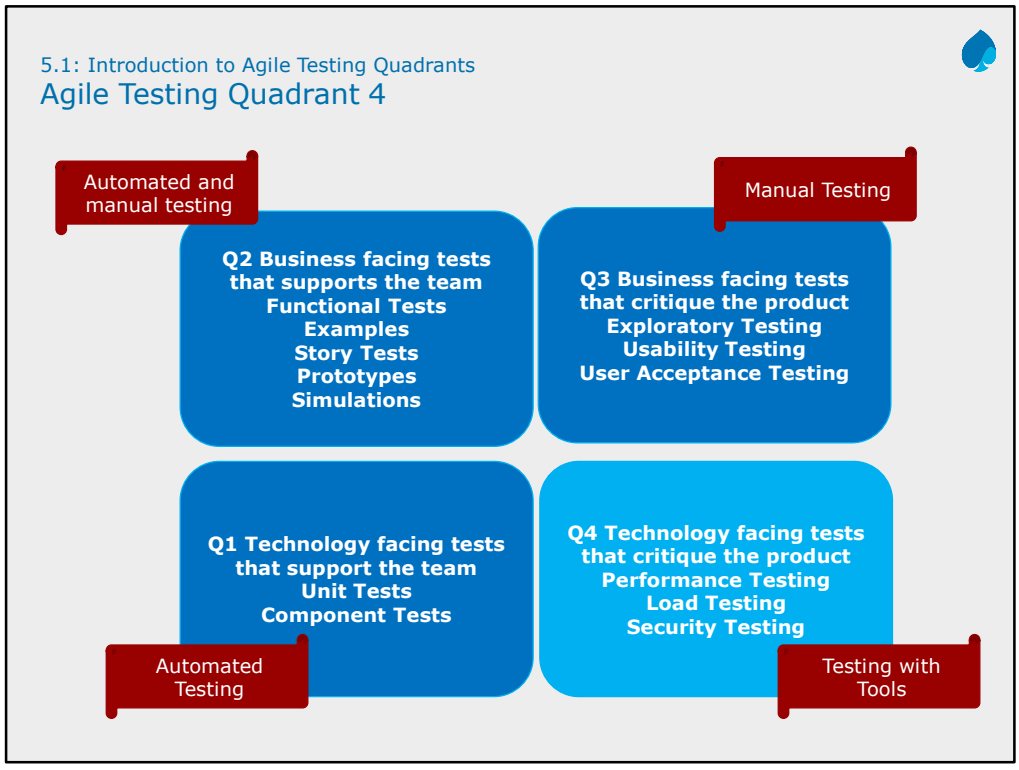
Agile Testing Quadrant 3 Goals

- Quadrant 3 classifies the business-facing tests that exercise the working software to see if it doesn't quite meet expectations or won't stand up to the competition
- When we do business-facing tests to critique the product, we try to emulate the way a real user would work on the application
- This is manual testing that only a human can do
- User Acceptance Test (UAT) gives customers confidence when they see requirements are met
- The benefits of UAT significantly outweigh the investments
- Exploratory testing is central to this quadrant
- During exploratory testing sessions, the tester simultaneously designs and performs tests, using critical thinking to analyze the results
- This offers a much better opportunity to learn about the application than scripted tests

5.1: Introduction to Agile Testing Quadrants

Agile Testing Quadrant 3 Toolkit

- Generate test data
  - e.g. PerlClip, Ruby script
- Emulators
  - Duplicate system behavior
  - e.g. mobile devices


5.1: Introduction to Agile Testing Quadrants
Agile Testing Quadrant 4
Automated and manual testing
Manual Testing
Q2 Business facing tests that supports the team
Functional Tests
Examples
Story Tests
Prototypes
Simulations
Q3 Business facing tests that critique the product
Exploratory Testing
Usability Testing
User Acceptance Testing
Q1 Technology facing tests that support the team
Unit Tests
Component Tests
Q4 Technology facing tests that critique the product
Performance Testing
Load Testing
Security Testing
Automated Testing
Testing with Tools

5.1: Introduction to Agile Testing Quadrants

Agile Testing Quadrant 4 Goals

- Technology-facing tests in Quadrant 4 are intended to critique product characteristics such as performance, robustness, and security
- This quadrant is responsible to deliver the ultimately finished product
- The application is made to deliver the expected value and non-functional qualities with help of this quadrant
- Kind of tests in this quadrant
  - Non-functional tests such as stress and performance testing
  - Security testing with respect to hacking and authentication
  - Infrastructure testing
  - Data migration testing
  - Scalability testing
  - Load testing
  - Maintainability Test
  - Compatibility Test
  - Data Migration Testing
  - Recovery Testing

5.1: Introduction to Agile Testing Quadrants

Agile Testing Quadrant 4 Toolkit

- Monitoring tools examples
  - jConsole
- Application bottlenecks, memory leaks
  - jProfiler
  - Database usage
- Commercial load test tools
  - Loadrunner
  - Silk Performer
- Open source test tools
  - jMeter
  - The Grinder
  - jUnitPerf
- Performance test providers
  - Multiple sites

5.2: Test Planning in Agile Testing

Introduction to Test Planning in Agile Testing

- On Agile projects the teams don't depend upon heavy documentation about what testers need to do
- Testers works with the agile team hand in hand so that the testing efforts are visible to all in the form of testing cards
- In release planning the agile team defines the purpose of the release, scope and assumptions
- Teams do quick analysis of these risks and plan test approach to address those issues
- In test planning we also consider and plan important aspects of testing that is automation, test environment, test data etc.
- In agile testing we need our test plans to be light weight and satisfy our needs
- The customer may require the test plan at the time of release for a compliance reasons

Test Planning in Agile Testing

In traditional software development methodologies a good amount of importance is given to the test plan documentation. The test plans are intended to outline the objective, scope, approach and focus of the software testing efforts for stakeholders. The completed document help people outside the testing group understand the "WHY" and "HOW" of the product validation.

5.2: Test Planning in Agile Testing

Introduction to Test Planning in Agile Testing (Cont.)

- The test plan should contain information about testing issues that are specific to this release or project
- It should contain risk analysis and assumptions identified
- The test plan should outline the critical success factors that your customers has identified
- Keep only that information which people need to know about testing and remove any extraneous information
- The main advantage of test planning is planning itself
- It allows you to consider and address issues such as test data requirements, infrastructure
- Test planning is a risk mitigation strategy

Summary

- In this lesson, you have learnt
  - Agile Testing Quadrants and their implementation
  - Depending upon the project requirements, scope, risks and priorities, the quadrants need to be chosen and implemented
  - Appropriate tools also need to be selected to assist and carry out the testing Implementing the quadrants help in collaborating the efforts of programmers, testers and customers
  - Knowing when to use the quadrants and how to implement them together is to be decided by the test team

Add the notes here.