



Agile Testing

**A 10 Minute Overview of Who, What, When, Why,
and How**

john.mufarrige@valtech.com





Agile Testing – Topics

- Roles (Who)
- Types (What)
- Process (When)
- Benefits (Why)
- Tools (How)
- Related Concepts



Agile Testing – Roles (Who)

➤ Who performs testing?

- Developer
- Dedicated Tester / QA Team
- Business Analyst / End User

➤ Who is interested in test results?

- Developer
- Development Manager
- Testing / QA Manager
- Business Analyst / End User



Agile Testing – Types (What)

- Unit Test
- Integration Test / Smoke Test
- Acceptance Test
- Performance Test
- Usability Test
- Reliability Test



Agile Testing – Types (What)

➤ How do you categorize tests?

- Granularity – At what level is the test applied?
 - Functions/methods, classes, modules, components, services, applications, frameworks
- Capability – What exactly are you trying to test?
 - Functional correctness, connectivity, performance, usability, fault tolerance
- Frequency – How often should the test be run?
 - Time-based: hourly, daily, weekly
 - Event-based: When code is modified/compiled, when code is committed to repository, at the end of an iteration
- Degree of Automation
 - Completely automated: no interaction required, test result summary can be emailed to interested parties
 - Completely interactive: tester manually executes each part of the test and inspects the results for verification

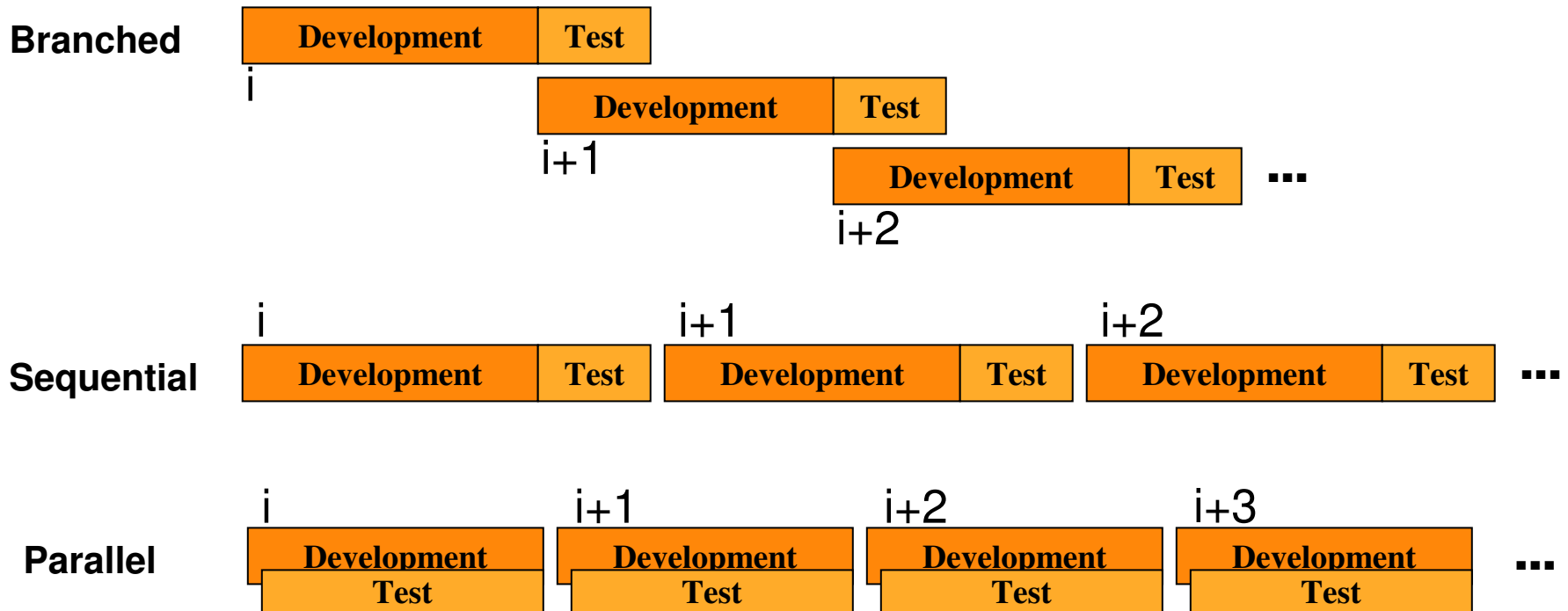
Agile Testing – Types (What)

Test Type	Granularity	Capability	Frequency	Automation	Notes
Unit Test	Fine grained, individual functions / methods	Functional correctness, method contract	Code modification, part of hourly CI process	Completely automated	Best support for regression testing
Integration Test / Smoke Test	Medium grained, components, layers	Connectivity, configuration	Configuration change, deployment	Batch / script initiated, results verified automatically	Vital for distributed systems
Performance Test	Typically medium to coarse grained, components, services	Load handling, throughput, response time	Verification of architecture or framework	Batch / script initiated, manual inspection of results	Should be done early if architectural choice is in question
User Acceptance Test	Coarse grained, services, use cases	Functional correctness	Delivered functionality, see process slide	Mixture of automated and interactive	Direct verification of requirements



Agile Testing – Process (When)

When should acceptance testing of iteratively-delivered functionality be performed? Three options:





Agile Testing – Process (When)

➤ Branched Testing

- Advantage: Iteration $i+1$ starts immediately after the end of iteration i .
- Disadvantages:
 - Defects are fixed at the expense of other deliverables that may be higher priority.
 - Code branches add complexity, risk, and potential delays.
 - It can be difficult to plan and schedule resources to anticipate defect levels.

➤ Sequential Testing

- Advantage: Iteration $i+1$ starts with fully tested and fixed codebase from iteration i .
- Disadvantages:
 - Defects are fixed at the expense of other deliverables that may be higher priority.
 - Iterations take longer and there is a considerable portion of time where zero progress is made.

➤ Parallel Testing

- Advantages:
 - Iterations are short and results delivered rapidly.
 - Defects are prioritized and scheduled so the highest priority work gets done first.
- Disadvantage: additional test staff may be needed to manage the parallel activities of executing acceptance tests for iteration i while designing acceptance test cases for iteration $i+1$.



Agile Testing – Benefits (Why)

- **Verification of requirements**
 - Acceptance Test
 - Performance Test
- **Code confidence / Ability to respond to change**
 - Unit Test
 - Integration Test
- **Design Quality**
 - Usability Test
 - Performance Test



Agile Testing – Tools (How)

➤ Unit Test

- xUnit: JUnit, HttpUnit, NUnit, cppUnit
- Agitar Agitator, Parasoft JTest

➤ Acceptance Test

- Watir, Selenium, Canoo, Abbott
- Parasoft WebKing, Mercury WinRunner

➤ Performance Test

- Grinder, JMeter
- Mercury LoadRunner, Empirix e-Load

➤ **This is not meant to be a list of recommended tools!**



Agile Testing – Related Concepts

- Continuous Integration (CI)
- Test Driven Development
- Executable Requirements
- Recording Defects as Tests
- Code Coverage
- Profiling