Acceptance Testing –
What does it mean to you?

Fran O'Hara
Inspire Quality Services
www.inspireqs.ie
fran.ohara@inspireqs.ie

1

We provide Agile, Quality and Process Improvement Services such as

- **Consulting/Coaching**:
  - Strategic advice and hands-on Coaching/mentoring in areas such as agile/lean (Scrum, XP, Kanban), testing, process improvement, etc.

- **Training** public/inhouse:
  - Lean/Agile: Getting Lean through Kanban, Succeeding with Agile/Scrum, PMI's Agile Certified Practitioner, Agile Testing, Product Owner training, etc.
  - Testing (ISTQB Foundation and Advanced Test Manager/Analyst, Risk-based testing, Test design techniques, Testing for developers, TMap®, Peer Reviews, UAT, etc.)
  - Requirements/Business analysis
  - Software project management

- **Assessments**
  - Agile practices
  - Industry standards and models such as CMMI®, TPI®, TMMi®, etc.

2

Agenda

- **What is Acceptance testing?**
- Acceptance testing in traditional plan-driven lifecycles
  - V-model
  - Test strategies in differing contexts
- Acceptance testing in agile
  - Agile – a few relevant concepts
  - Agile test strategy
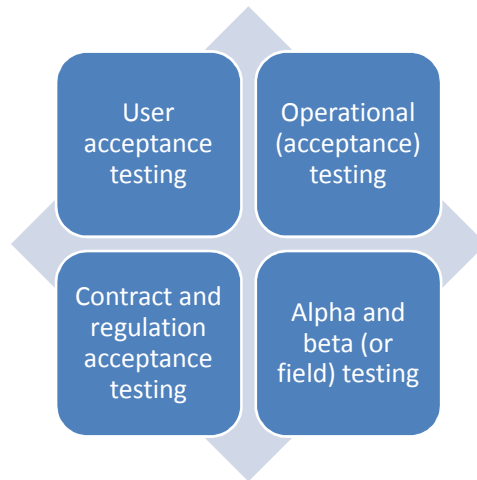   - Quadrant thinking and automation pyramid
- Summary & Conclusions

3

What is Acceptance Testing?

- **ISTQB : (user) acceptance testing:** Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.   [After IEEE 610]

- **User Acceptance Testing** - It's a form of testing to verify the system can support day-to-day business and user scenarios to validate rules, various workflows, data correctness, and overall fit for use and ensure the system is sufficient and correct for business usage - Wikipedia

- **Acceptance testing** is any testing done by one party for the purpose of accepting another party's work. – James Bach

4

Agenda

- What is Acceptance testing?
- **Acceptance testing in traditional plan-driven lifecycles**
  - V-model
  - Test strategies in differing contexts
- Acceptance testing in agile
  - Agile – a few relevant concepts
  - Agile test strategy
    - Quadrant thinking and automation pyramid
- Summary & Conclusions

5

V-Model

Early test design

Requirements

(User)
Acceptance test

Functional Spec.

System test

Reviews

Hi level design

Integration test

Lo level design

Unit test

Static Analysis

Code

Dynamic Testing

Static Testing

6

Typical forms of Acceptance Testing

- User acceptance testing
- Operational (acceptance) testing
- Contract and regulation acceptance testing
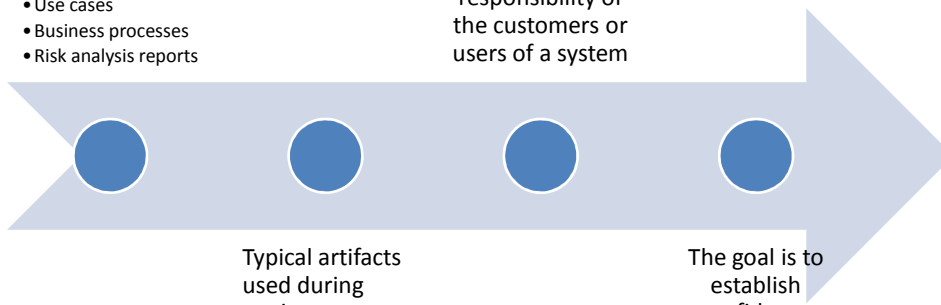- Alpha and beta (or field) testing

7

(User) Acceptance testing

Test basis:
- User/business requirements
- System requirements
- Use cases
- Business processes
- Risk analysis reports

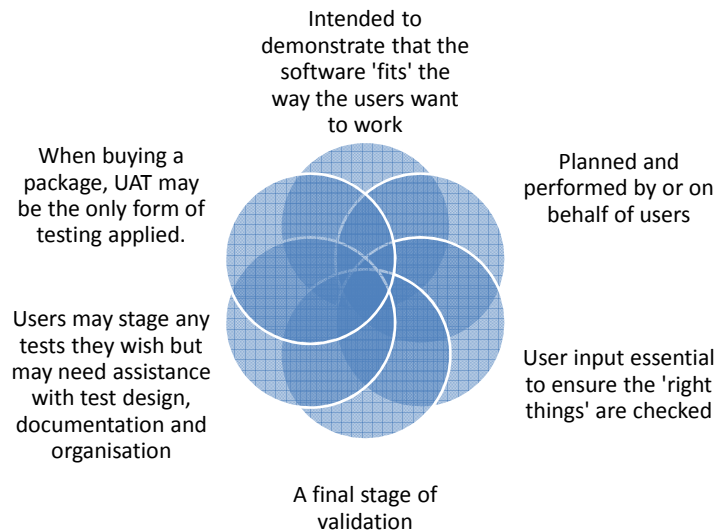Acceptance testing is often the responsibility of the customers or users of a system

Typical artifacts used during testing:
- Business processes on fully integrated system
- User procedures
- Forms
- Reports
- Configuration data

The goal is to establish confidence

8

User acceptance testing (UAT)

Intended to demonstrate that the software 'fits' the way the users want to work

When buying a package, UAT may be the only form of testing applied.

Planned and performed by or on behalf of users

Users may stage any tests they wish but may need assistance with test design, documentation and organisation

User input essential to ensure the 'right things' are checked

A final stage of validation

Inspire Quality Services   Slide 9

Prerequisites for User Acceptance Testing

Appropriate resources available

Business Requirements available

Application Code fully developed

Unit Testing, Integration Testing & System Testing should be completed

No Critical/High/Medium defects in System (Integration) Test

Regression Testing completed

All the reported defects should be fixed and tested before UAT

Traceability matrix for key test levels completed

UAT Environment ready

Exit criteria for System Testing met

Copyright © 2013 Inspire QS   10

Usability Testing

Asking users to report 'usability' problems during UAT is a weak form of usability testing

Can use usability heuristics on UI specs or prototypes or early versions of the GUI but effective usability testing should additionally involve users…

An approach:

- Focus the user experience rather than just specific features
- Select a representative sample of users to perform the tasks
- Provide the users with key tasks to perform (not scripts!)
- Observe them and use 'talking aloud' protocol
- Ideally whole team observes in live mode
- Feedback discussion/debrief with users and with team

11

Acceptance Testing

- **Operational Acceptance Testing (OAT)** Also known as operational readiness testing, this refers to the checking done to a system to ensure that processes and procedures are in place to allow the system to be used and maintained. This may include checks done to back-up facilities, procedures for disaster recovery, training for end users, maintenance procedures, and security procedures.

- **Contract and regulation acceptance testing** In contract acceptance testing, a system is tested against acceptance criteria as documented in a contract, before the system is accepted. In regulation acceptance testing, a system is tested to ensure it meets governmental, legal and safety standards.

- **Alpha and beta testing** Alpha testing takes place at developers' sites, and involves testing of the operational system by internal staff, before it is released to external customers. Beta testing takes place at customers' sites, and involves testing by a group of customers who use the system at their own locations and provide feedback, before the system is released to other customers. The latter is often called "field testing". – Wikipedia
- **End-to-end testing …**

12

Contract acceptance testing

Aims to demonstrate that the supplier's obligations are met

Similar to UAT, focusing on the contractual requirements as well as fitness for purpose
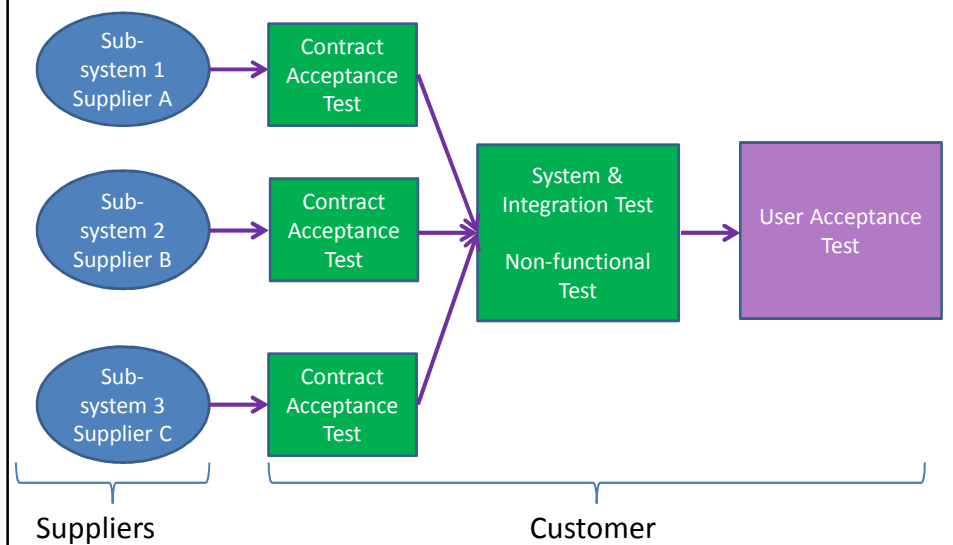
Contract should state the acceptance criteria

Stage payments may be based on successful completion.

Inspire Quality Services   Slide 13

A real world example - combination

Regulation acceptance testing

For example:

- FDA (U.S. Food & Drug Administration) regulate medical devices, pharmaceutical industry, etc.

  Software 'Validation' regulations include

  – Acceptance testing against requirements
  – Traceability
    - Between and to Requirements
    - Product risks based on safety (Hazards Analysis, FMECA, etc.)

  Clinical trials typically also required

Alpha and beta testing

- Often used by suppliers of packages/products (particularly shrink-wrapped)
- Where supplier wishes to receive feedback from actual or potential customers
- Alpha testing normally takes place on the supplier site
  – Performed by business/sales/support types
- Beta testing usually conducted by selected beta customers
  – Performed by users on their site
  – Similar to FOA/GA concept used for example in the telecommunications industry

Alpha and beta testing - intent

- To get market feedback on the product
- Are major features missing?
- Do new features 'miss the point'?
- Is product ready for release?
- Some supplier leave faults in the software to get bug reports returned to gauge:
  - where software is being used most
  - where users are most sensitive to faults.

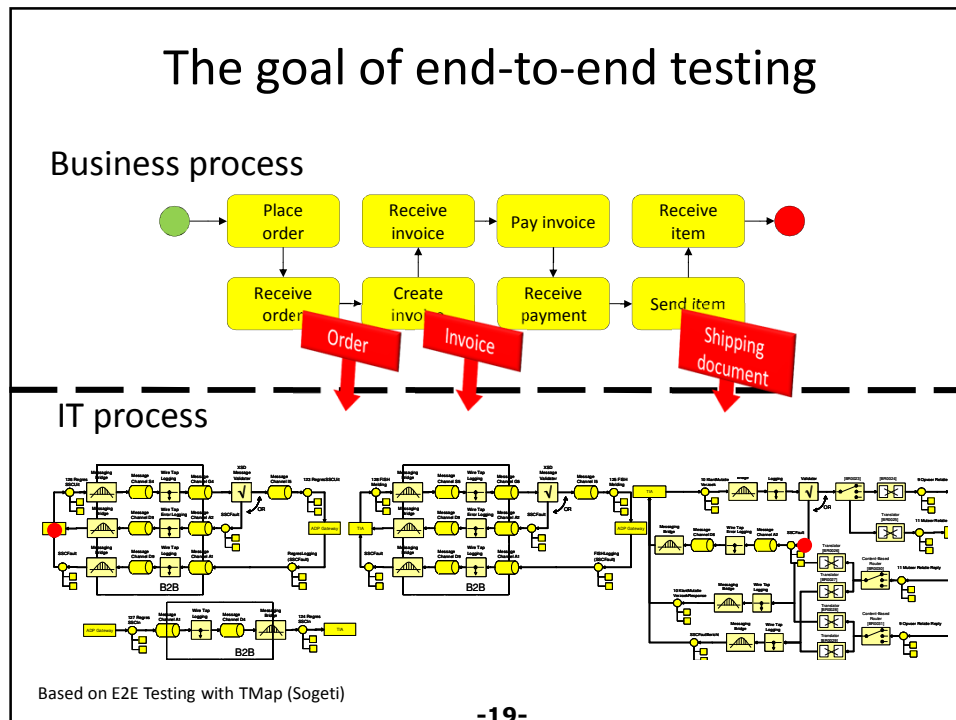Inspire Quality Services   Slide 17

Context: E2E testing and the V-model
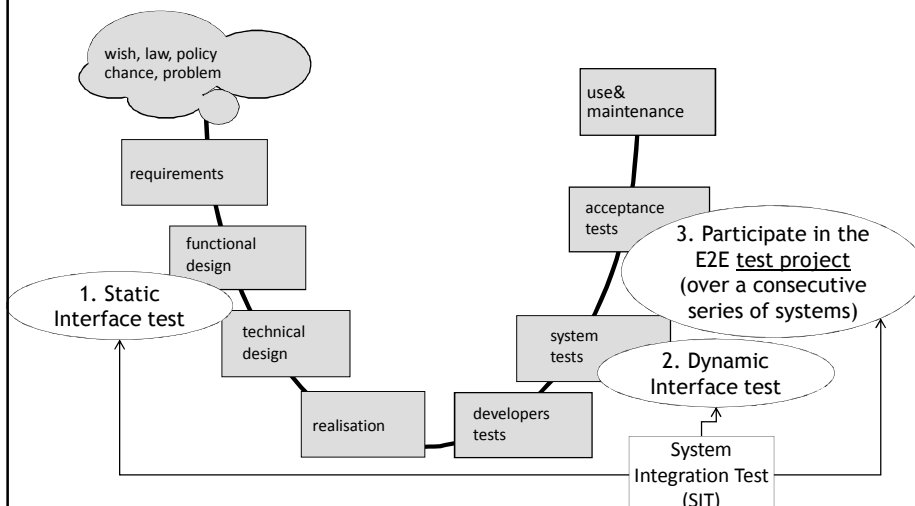
System Integration Test is executed in 3 steps
1. Static Interface test: by comparing the interface docs.
Moment of execution: design phase

The goal of end-to-end testing

Business process

IT process

Based on E2E Testing with TMap (Sogeti)

-19-
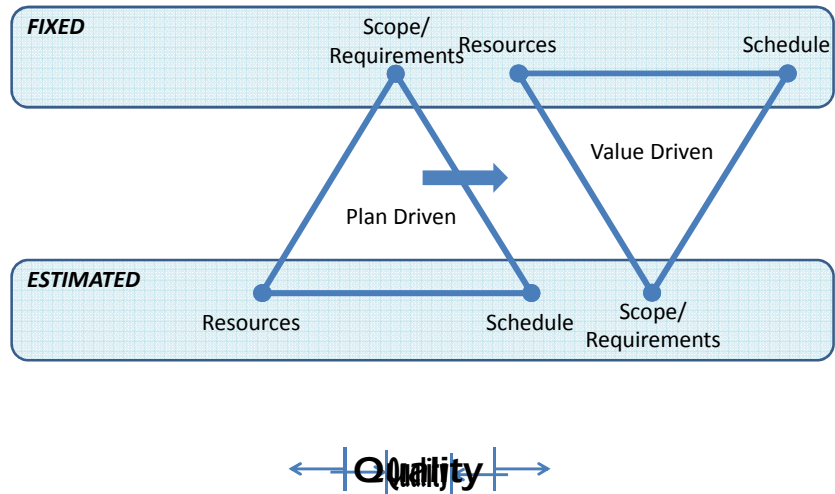
Agenda

- What is Acceptance testing?
- Acceptance testing in traditional plan-driven lifecycles
  - V-model
  - Test strategies in differing contexts
- **Acceptance testing in agile**
  - Agile – a few relevant concepts
  - Agile test strategy
    - Quadrant thinking and automation pyramid
- Summary & Conclusions

20

Flipping the Iron Triangle

FIXED

Scope/
Requirements   Resources   Schedule

Value Driven

Plan Driven

ESTIMATED

Resources   Schedule   Scope/
Requirements

Quality

21

The Life of an Iteration

1-4 WEEKS
Iteration

Product Backlog

StakeHolder
Demo/Review

Team
Retrospective
(CI)

1 DAY
daily stand-
up meeting

Test &
Develop
Increment

Update &
Groom Product
Backlog

Iteration
Planning
Meeting

Sprint
Backlog

VISION
+
TEAM

(potentially releaseable)
PRODUCT
+
IMPROVED TEAM

22

PLAN-DRIVEN, WATERFALL METHOD

NEED + RESOURCES → Requirement → Design → Implementation → Verification → Maintenance → PRODUCT RELEASE + RESOURCES

AGILE METHOD

Stability — Change — Stability — Change — Stability — Change — Stability — Change — Stability — Change

VISION + TEAM → IMPROVED TEAM

(potentially releasable) PRODUCT
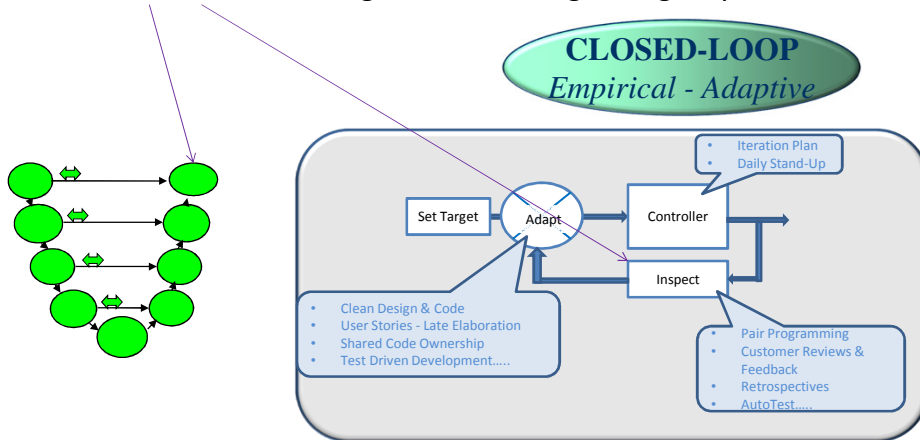
23

Validation in traditional versus agile

- Verification – checking we are building the system right
- Validation – checking we are building the right system

CLOSED-LOOP
Empirical - Adaptive

- Iteration Plan
- Daily Stand-Up

Set Target — Adapt → Controller → Inspect

- Clean Design & Code
- User Stories - Late Elaboration
- Shared Code Ownership
- Test Driven Development…..

- Pair Programming
- Customer Reviews & Feedback
- Retrospectives
- AutoTest…..

24

The Major Agile/Lean Methods

- **Scrum (1995) – PM Oriented**
  - Timeboxing
  - Prioritized backlog
  - Daily standup meetings
  - Demo after each iteration
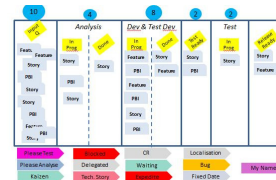 - Correct the process through lessons learned

XP (1999) – Engineering Oriented
- (A)TDD, refactoring, pair programming, continuous integration, simplicity, whole team, planning game, …
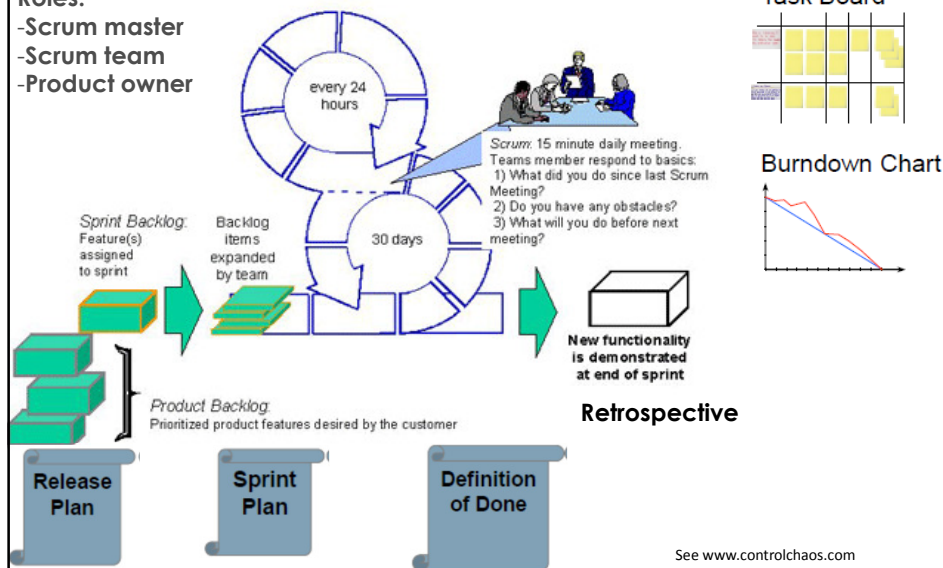
Kanban(2010) – Continuous Improvement
  Visualize
  Reduce WIP
  Manage Flow
  Make process Policies Explicit
  Nurture effective feedback loops
  Improve Collaboratively (using scientific method)

25

Scrum

Roles:
-**Scrum master**
-**Scrum team**
-**Product owner**

Task Board

Scrum: 15 minute daily meeting. Teams member respond to basics:
1) What did you do since last Scrum Meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?

every 24 hours

30 days

Sprint Backlog: Feature(s) assigned to sprint

Backlog items expanded by team

New functionality is demonstrated at end of sprint

Burndown Chart

Retrospective

Product Backlog: Prioritized product features desired by the customer

Release Plan

Sprint Plan

Definition of Done

See www.controlchaos.com

26

Scrum Phases? – but beware

- Planning phase steps
  - Product backlog prioritized and ready?
    - At least for first sprint or two!
  - Architecture defined?
    - Versus emergent!?.... 'architectural vision'
  - Release & Test Planning
- Development iterations ← **Most focus**
  - Build quality software/documentation
- Implement phase steps ('End Game')
  - System integration testing
    - But integrate early as much as possible
  - Final performance testing
  - UAT/Beta....

Waterfall ⟷ Agile

27

Evolving from sequential to iterative/incremental!

A

Sprint 1 | Sprint 2

| Code | | Code | Code & Bug Fix |

Test

B

Sprint 1 | Sprint 2

| Code | Code & Bug Fix | | Code | Code & Bug Fix |

Test | Test

C

Sprint 1 | Sprint 2

| Code & Bug Fix | | Code & Bug Fix |

Test | Test

Task Board

28

'Acceptance' Testing in Agile

An acceptance test is a formal description of the behaviour of a software product, generally expressed as an example or a usage scenario. ..

- in many cases the aim is that it should be possible to automate the execution of such tests by a software tool, either ad-hoc to the development team or off the shelf.
- Similarly to a unit test, an acceptance tests is generally understood to have a binary result, pass or fail;
- For many Agile teams acceptance tests are the main form of functional specification; sometimes the only formal expression of business requirements. ..

Also known as

- The terms "functional test", "acceptance test" and "customer test" are used more or less interchangeably.
- A more specific term "story test", referring to user stories is also used, as in the phrase "story test driven development".

- Agile Alliance

29

Done

- What does "Done" mean for the project?...
  - Design doc completed for maintenance purposes
  - Code checked in and coding standard checked by tool
  - Builds
  - Unit tests complete successfully
    - 80% code branch coverage on unit tests
  - 100% Boundary Value coverage
  - Acceptance tests passed
  - Within acceptable defect levels
  - Non functionally tested (performance, security…?)
  - Integration tested ………Etc.
  - Accepted by product owner
  - Product documents updated
  - Sales materials updated.....

30

Examples of Definition of Done

Working software & demo
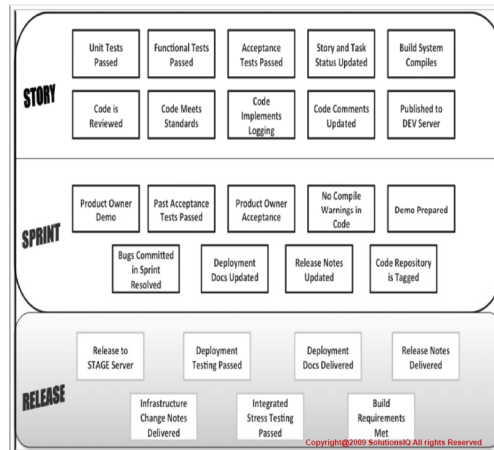- ✓ Unit test
- ✓ Code review
- ✓ Installer

Tests
- ✓ Functional
- ✓ Performance
- ✓ Regression

Documentation
- ✓ User docs/Online help
- ✓ Internal design docs
- ✓ Release notes
- ✓ API documents

Example1

Example2

31

How agile changes things

- Whole Team Approach - collaboration
- Coding and testing are integrated rather than distinct phases
- Early and frequent feedback
- TDD/ATDD practices
- Test-infected developers, better automation strategies, better designed tests

  Always working software

32

Agile Test Strategy

- Risks
  - Similar product risks
  - Regression risk with high level of change
- How many test levels?
  - XP appears to advocate two as part of a predefined test strategy
    - Unit and (Story-based) Acceptance testing - both automated as part of Test Driven Development
    - Is system test no longer required?
    - What does 'acceptance testing' mean now?
  - Automation reduces regression risk
  - Developers doing testing reduces risk of poor quality code
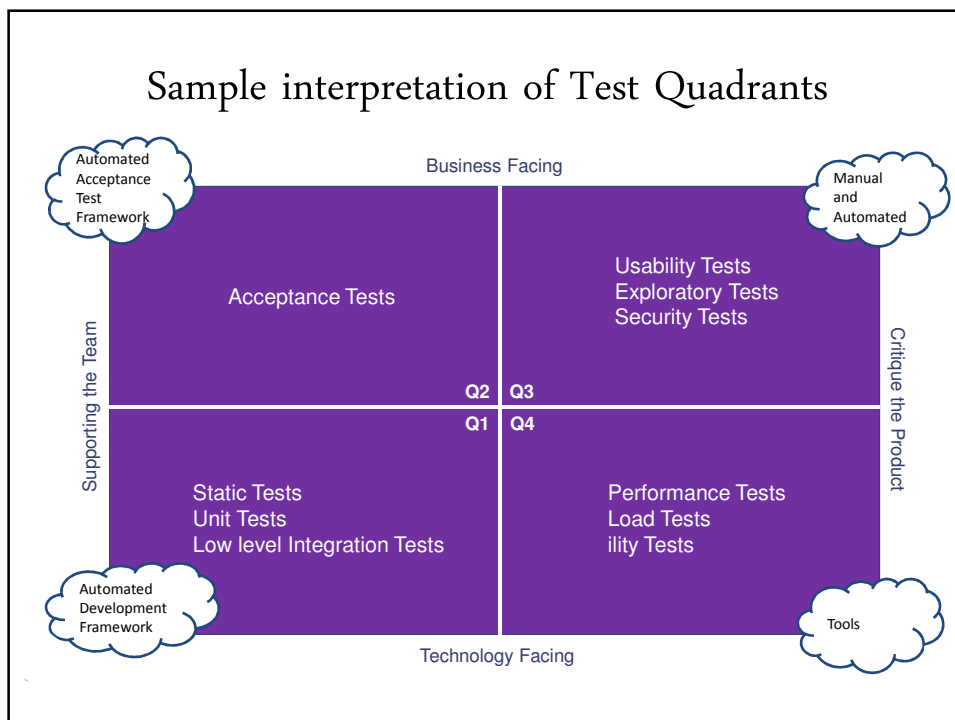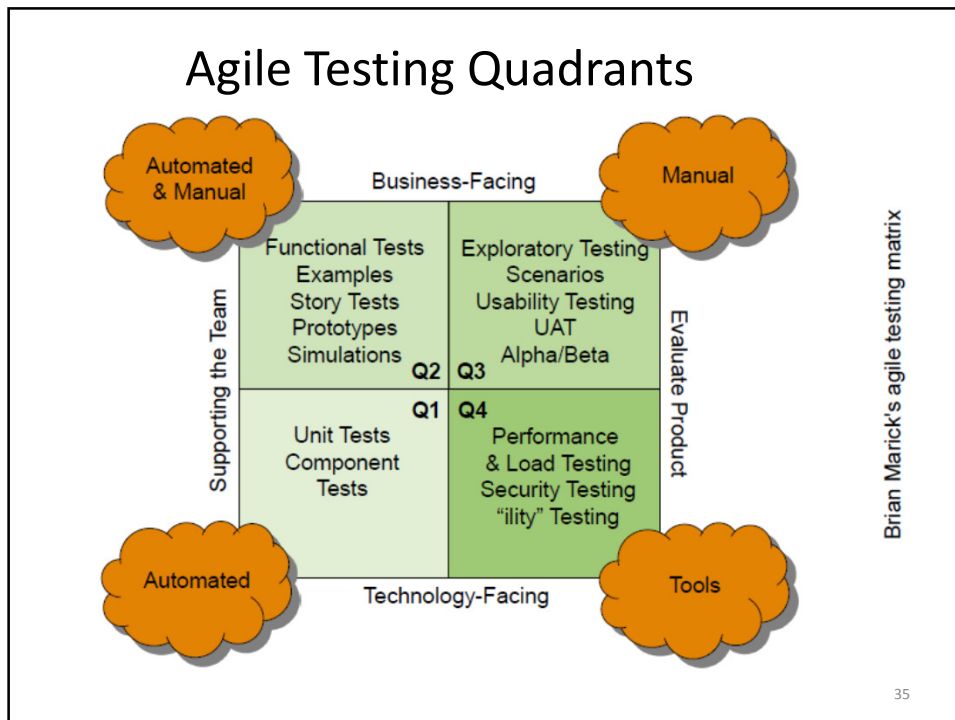 - But how can a test strategy/approach be method rather than product based?
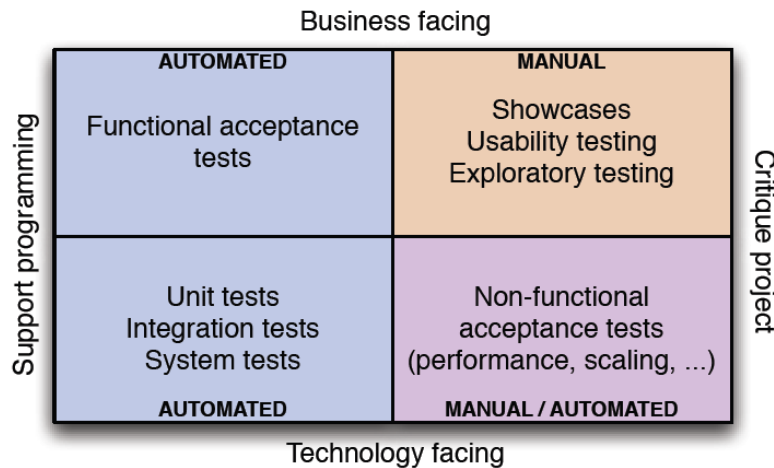
33

'Acceptance' Testing – is it enough?

- May not be…context/risk/strategy issue…
  - May not be fully automated – partial regression strategy needed
  - Expand to fuller 'system' tests
    - Functional testing
    - Non-functional testing – performance, usability, etc.
  - May still need more user story interaction tests, end-to-end business scenario focused User Acceptance test, etc.
  - System integration testing issues
  - Etc.
- Strategy and scheduling issue
  - Risk-driven, adaptive

34

Agile Testing Quadrants

Sample interpretation of Test Quadrants

More interpretations….

Business facing

AUTOMATED	MANUAL
Functional acceptance tests	Showcases Usability testing Exploratory testing
Unit tests Integration tests System tests	Non-functional acceptance tests (performance, scaling, …)
AUTOMATED	MANUAL / AUTOMATED

Support programming

Critique project

Technology facing

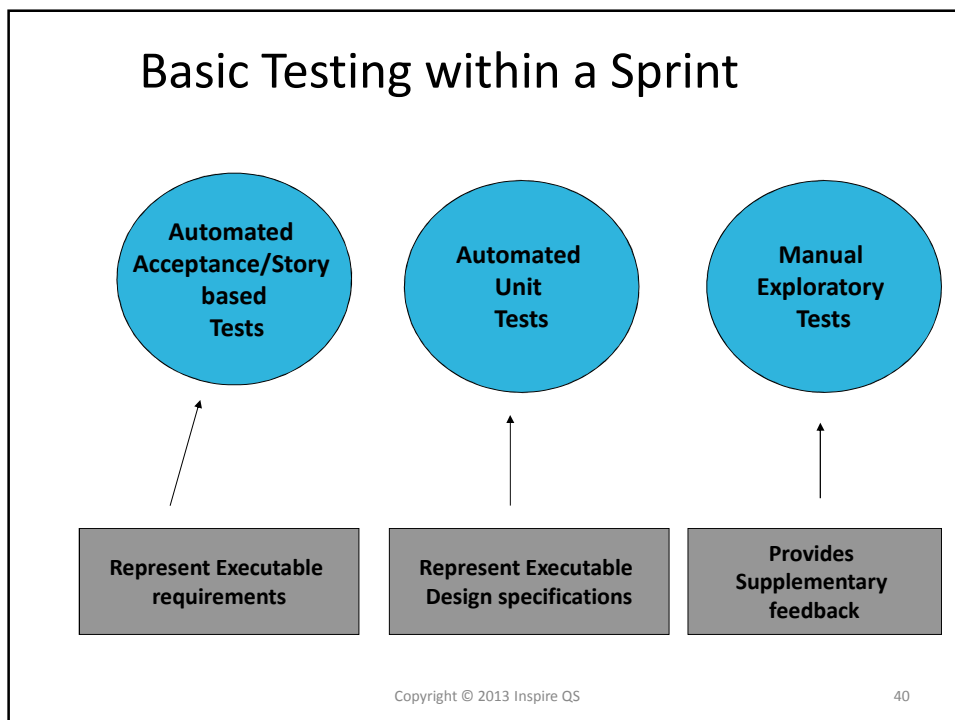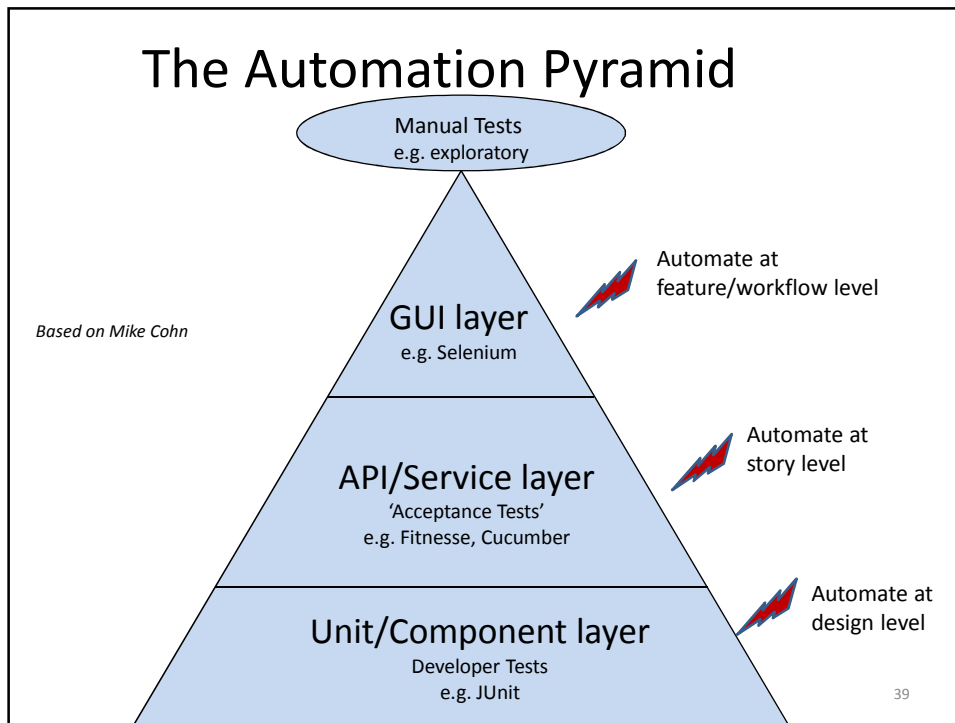From 'Maintainable Acceptance Tests', Janakiram/Humble, Agile 2012 Dallas
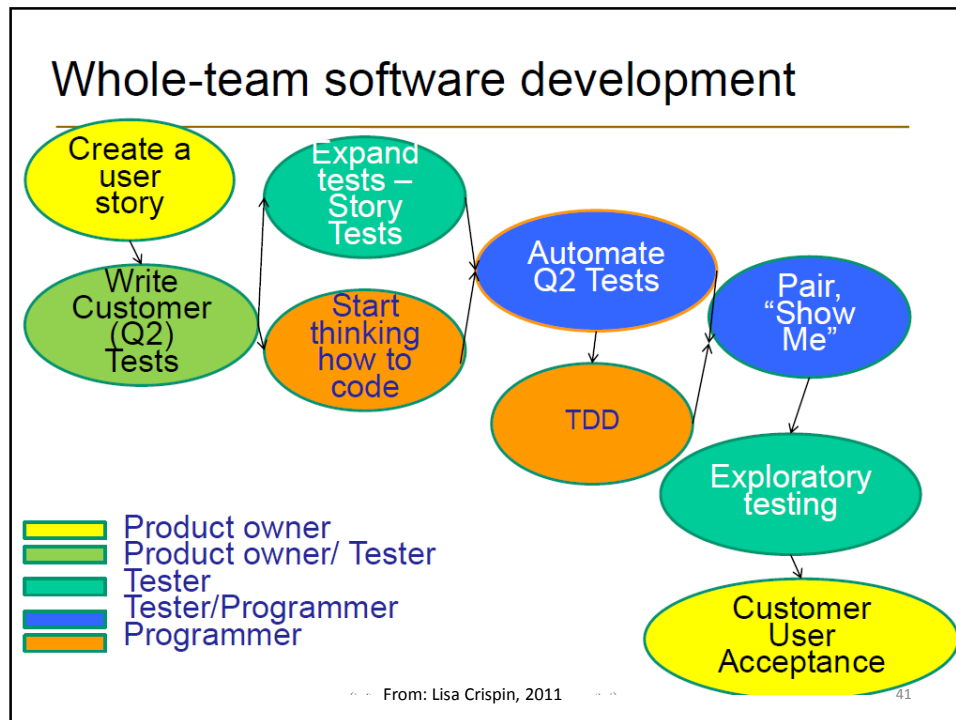
37

More interpretations….

'Some people use the term 'acceptance tests' to describe Q2 tests, but we believe that acceptance tests encompass a broader range of tests that include Q3 and Q4.
Acceptance tests verify that all aspects of the system, including qualities such as usability and performance, meet customer expectations. '

– from 'Agile Testing', Crispin/Gregory

38

The Automation Pyramid

Manual Tests
e.g. exploratory

Based on Mike Cohn

GUI layer
e.g. Selenium

Automate at
feature/workflow level

API/Service layer
'Acceptance Tests'
e.g. Fitnesse, Cucumber

Automate at
story level

Unit/Component layer
Developer Tests
e.g. JUnit

Automate at
design level

39

Basic Testing within a Sprint

Automated
Acceptance/Story
based
Tests

Automated
Unit
Tests

Manual
Exploratory
Tests

Represent Executable
requirements

Represent Executable
Design specifications

Provides
Supplementary
feedback

40

Whole-team software development

Create a user story

Expand tests – Story Tests

Automate Q2 Tests

Pair, "Show Me"

Write Customer (Q2) Tests

Start thinking how to code

TDD

Exploratory testing

Customer User Acceptance

Product owner
Product owner/ Tester
Tester
Tester/Programmer
Programmer

From: Lisa Crispin, 2011

41

But is this enough?

Keep track of the big picture
- Consider how each story affects rest of application
- Does it affect other stories, other systems?
- Are there non-functional implications?
- What about end-to-end tests?
- …..Think about the testing quadrants…

42

Maintaining Context

PRIORITY

GRANULARITY

43

Sprints and Testing Strategy

| Sprint 1 Dev + Test* | → | Sprint 2 Dev + Test* | → | Sprint 3 Dev + Test* | ... |

| Additional testing | | Additional testing | | Additional Testing |
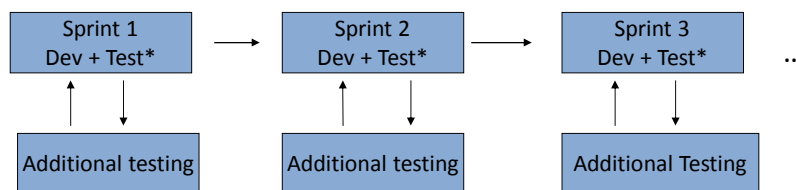
*Sprint test = Automated Unit & Acceptance, Manual Exploratory

Within a Sprint may need to perform additional testing as part of a defined but adaptive testing strategy e.g.:
- Feature/'epic' or workflow level testing
- Combination/feature interaction testing
- Business cycle & end-to-end scenario testing – exercising multiple stories, end of month processing, etc.
- Performance testing
- Usability testing
- Security testing
- System integration testing
- Note: Ideally any testing needed should be included within the Sprint rather than being deferred....

Evolve to fully Working Software!! 44

Large / Multiple Teams

From: Janet Gregory 2011
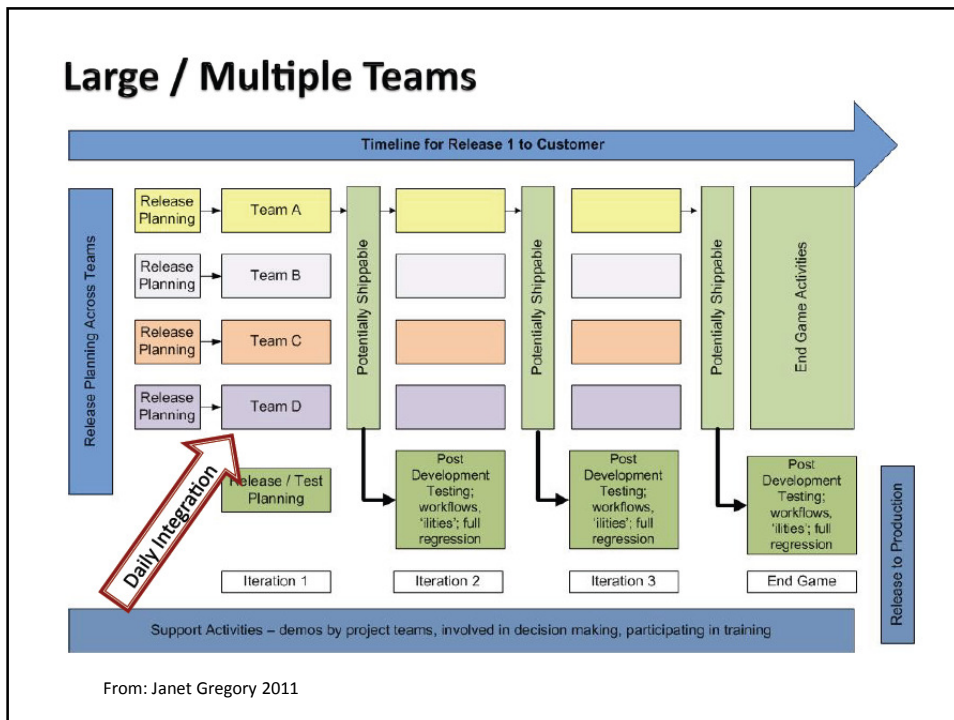
Acceptance Testing

WRAP-UP

Conclusions

- Adapt test strategy (and acceptance testing) to your context e.g.
  - Lifecycle
    - Sequential
    - Iterative/incremental e.g. Agile/lean
  - Organisational
    - IT
    - Product development
    - Outsourcing
  - Domain area
    - Regulated - Safety critical, Financial Services, …
    - Web, embedded, …
  - Product risks
- Based on above, agree (local) definition of terms and disseminate!

47

Inspire
Quality Services

Any other questions/issues?

Fran O'Hara
Inspire Quality Services

www.inspireqs.ie
fran.ohara@inspireqs.ie

48