Agile Testing and the Role of the Agile Tester

Declan Whelan
dwhelan@dpwhelan.com

Agenda

- What is Agility?
- Agile Testing Success Factors
- Agile/Scrum Overview
- What is Quality?
  - Context-Driven School of Testing
  - Session-Based Testing & Low-Tech Dashboards
- Agile Quality
  - Agile Testing Matrix
  - Agile Tester Activities
  - Test Automation

Notes on this Presentation

This presentation was originally delivered at the QUEST Toronto 2008 conference. *Prior to the conference it was discovered that information on session based testing was not properly attributed to James Bach (http://www.satisfice.com). Unfortunately an updated presentation was not available in time to be included on the USB sticks provided.*

This updated presentation correctly attributes this material to James and also includes some minor formatting changes.

Declan wishes to apologize to James for the omission and to thank him both for his understanding and his contributions to exploratory testing.

Agile Testing Success Factors

Testers are part of the team	• Be cathedral builders not stone cutters • Collective ownership
Agile testing mindset	• Drop the “Quality Police” mindset • Focus on team goals & customer value
Automate tests	• Automate tests wherever practical • Need rapid feedback
Look at the big picture	• Balance against developer focus on technical implementation • Use agile test matrix as guide

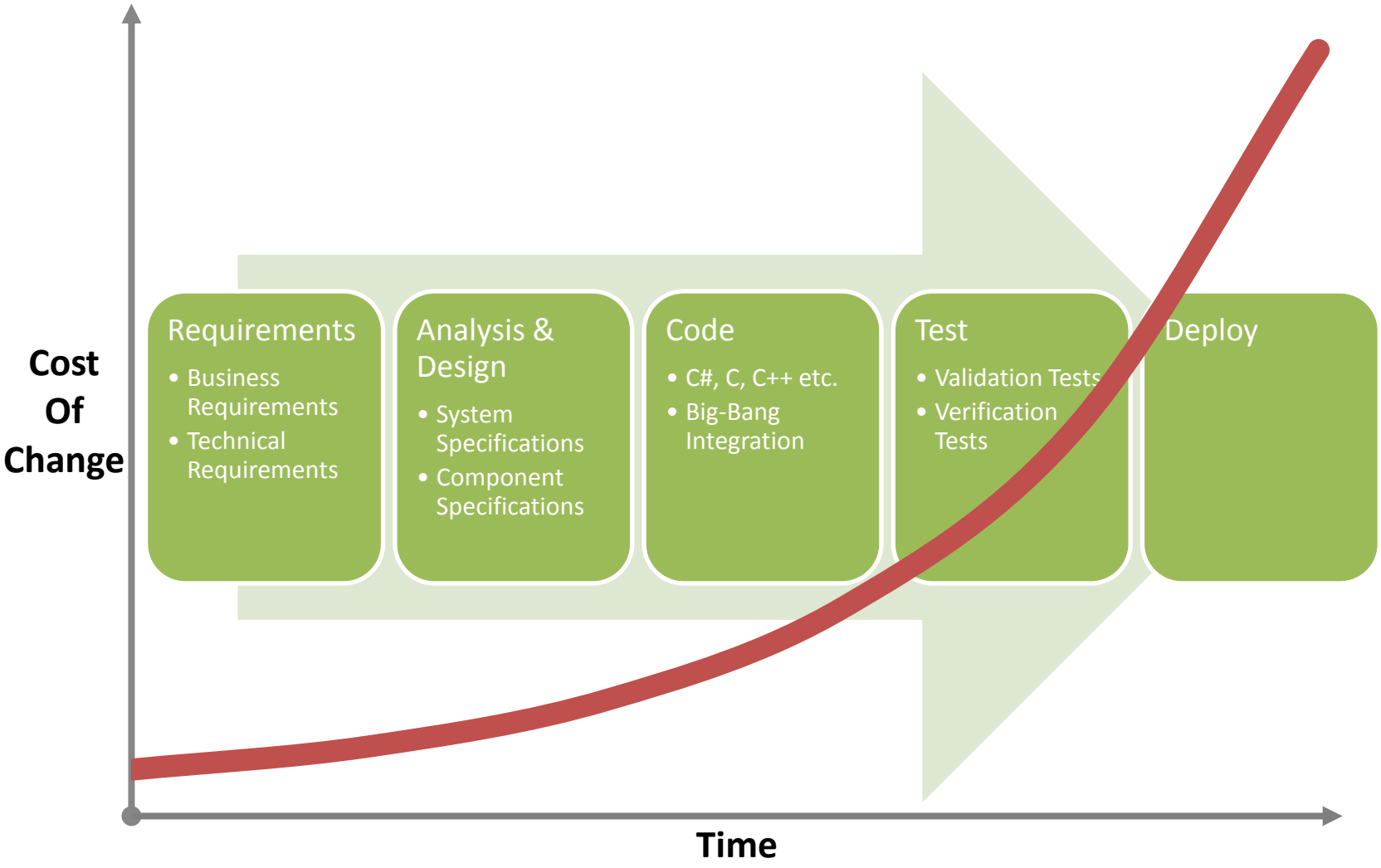
Source: http://www.agiletester.ca/

Agile Testing Success Factors

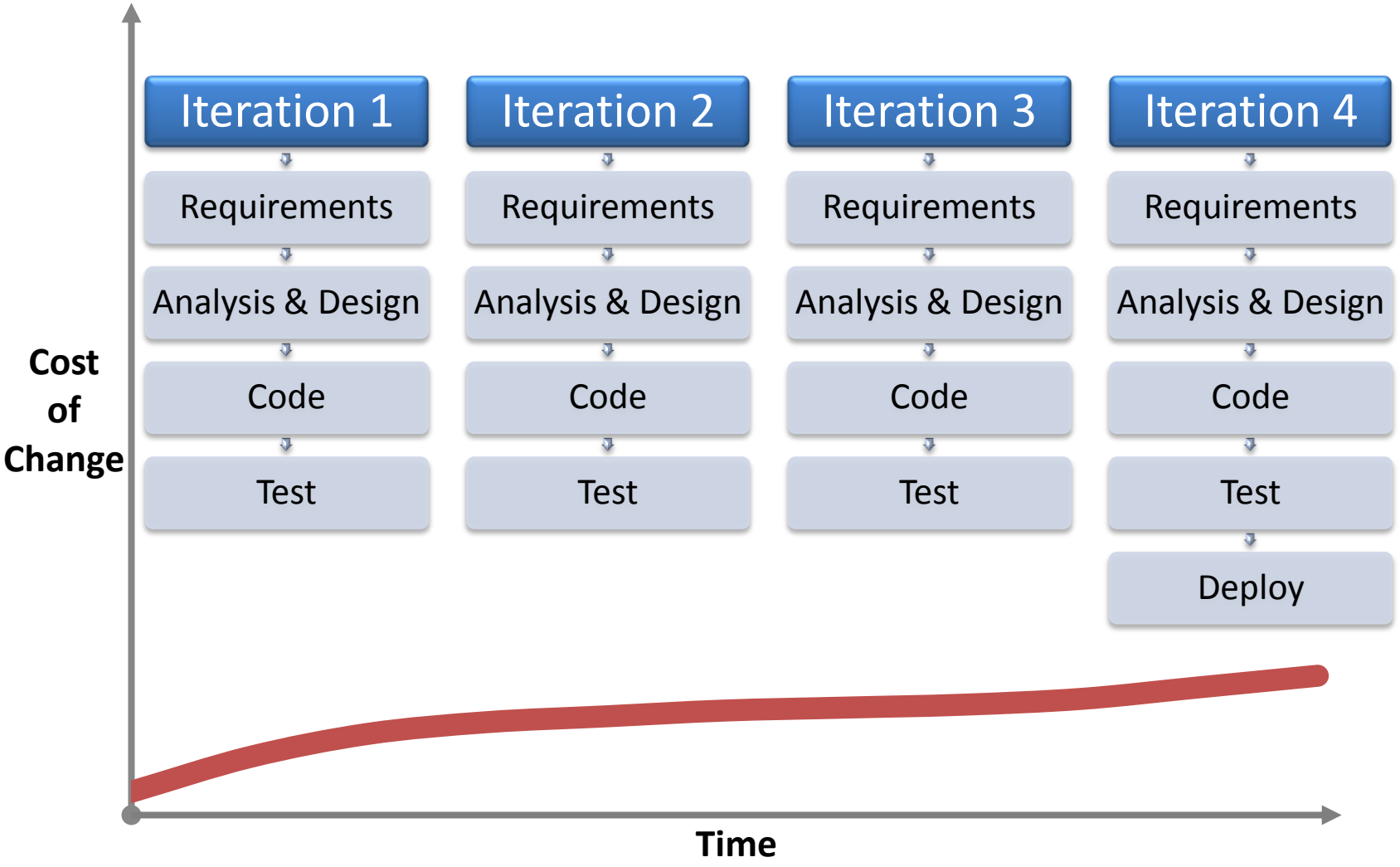
Foundation of critical practices	• Session-based testing, agile test environments • Informative workspace
Collaborate	• Collaborate with customers • Collaborate within team
Continually improve	• Team retrospectives • Personal training: reading, blogs, QAI, local QA groups

Source: http://www.agiletester.ca/

Traditional Approach - Waterfall



Agile Approach



What is Agility?

"Values, principles and practices that foster team communication and feedback to regularly deliver customer value through working software."

Agile Values

Individuals & Interactions	Processes & Tools
Working Software	Comprehensive Documentation
Customer Collaboration	Contract Negotiation
Responding to Change	Following a Plan

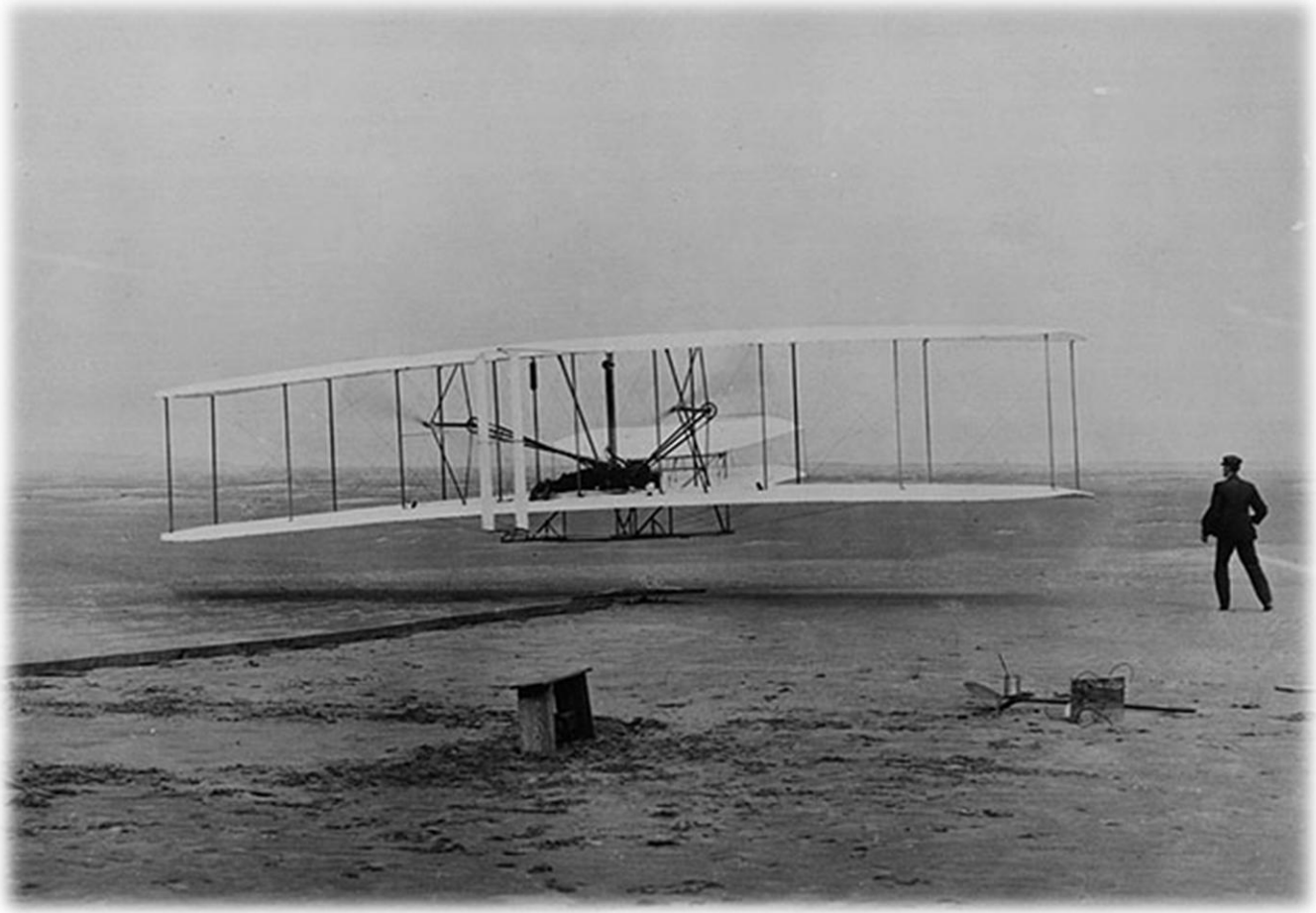
http://agilemanifesto.org/

Source: Brian Marick

Quarry

Lean

Engineering



XP – Customer Bill of Rights

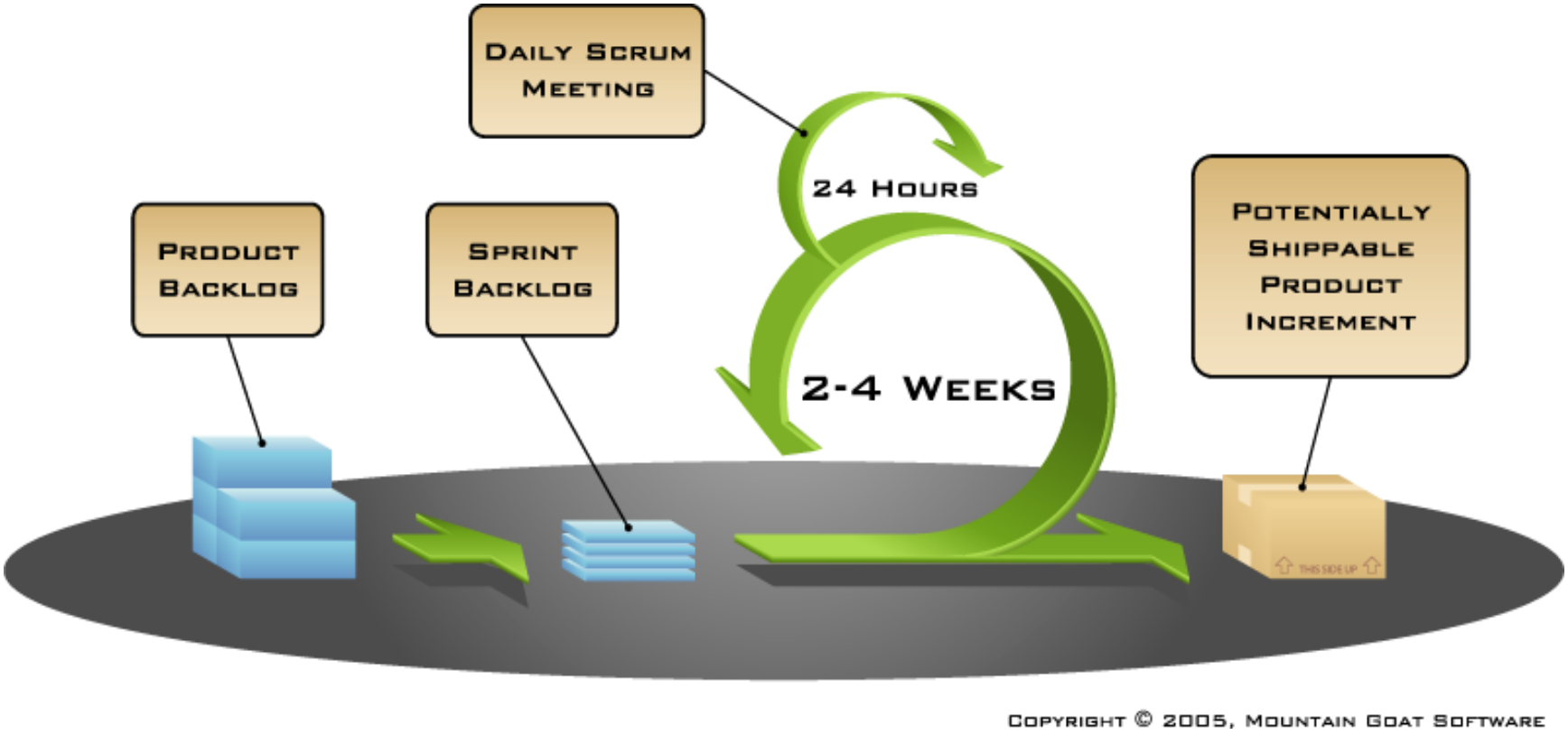
- *You have the right to an overall plan, to know what can be accomplished, when, and at what cost.*
- *You have the right to see progress in a running system, proven to work by passing repeatable tests that you specify.*
- *You have the right to change your mind, to substitute functionality, and to change priorities.*
- *You have the right to be informed of schedule changes, in time to choose how to reduce scope to restore the original date. You can even cancel at any time and be left with a useful working system reflecting investment to date.*

XP – Tester Bill of Rights

- You have the right to bring up issues related to quality and process at any time.
- You have the right to ask questions of customers and programmers and receive timely answers.
- You have the right to ask for and receive help from anyone on the project team, including programmers, managers and customers.
- You have the right to make and update your own estimates for your own tasks and have these included in estimates for stories.
- You have the right to the tools you need to do your job in a timely manner.
- You have the right to expect your project team, not just yourself, to be responsible for quality.

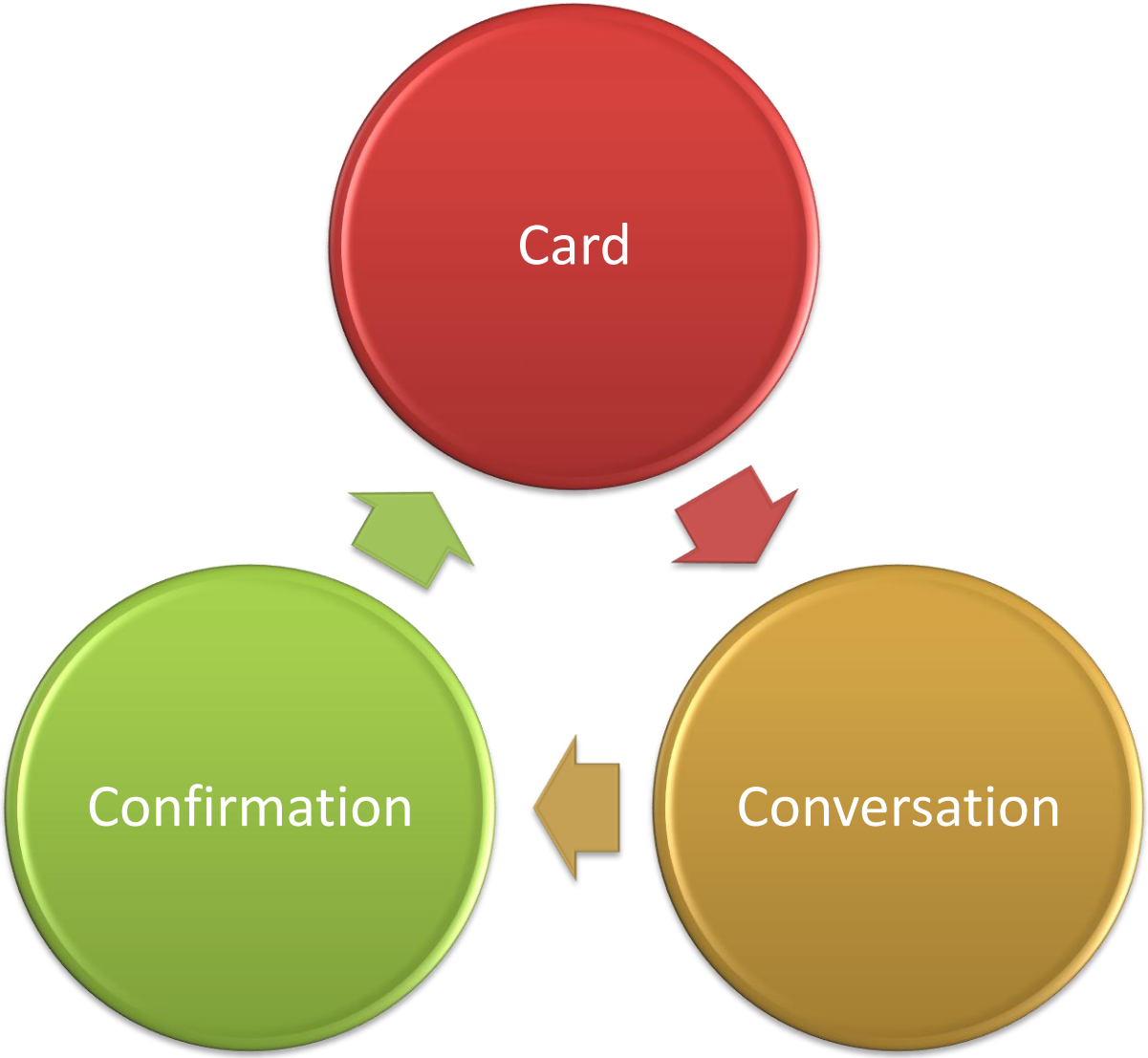
Source: http://home.att.net/~lisa.crispin/XPTesterBOR.htm

Scrum Overview



http://www.mountaingoatsoftware.com/scrum_figures

Story



What is a Story?

- Functionality that provides value
- Written in business (domain) language
- Ideally < 2 days work
- "As a <*role*>, I want to <*feature*> so that <*benefit*>"

Independent

Negotiable

Valuable

Estimable

Small

Testable

Story Card

- The “headline” for some new functionality.
- It is an “invitation to a conversation”.
- Written in business (domain) language.
- It is not a specification … specifications are documented using story tests.
- Stories can be used to drive additional work such as building test scripts/frameworks

Story Conversation

- Exchange of thoughts, opinions, and feelings.
- Takes place over time, particularly when the story is estimated.
- Conversations also occur at the iteration planning meeting when the story is scheduled for implementation.
- Supplement with documents as needed.

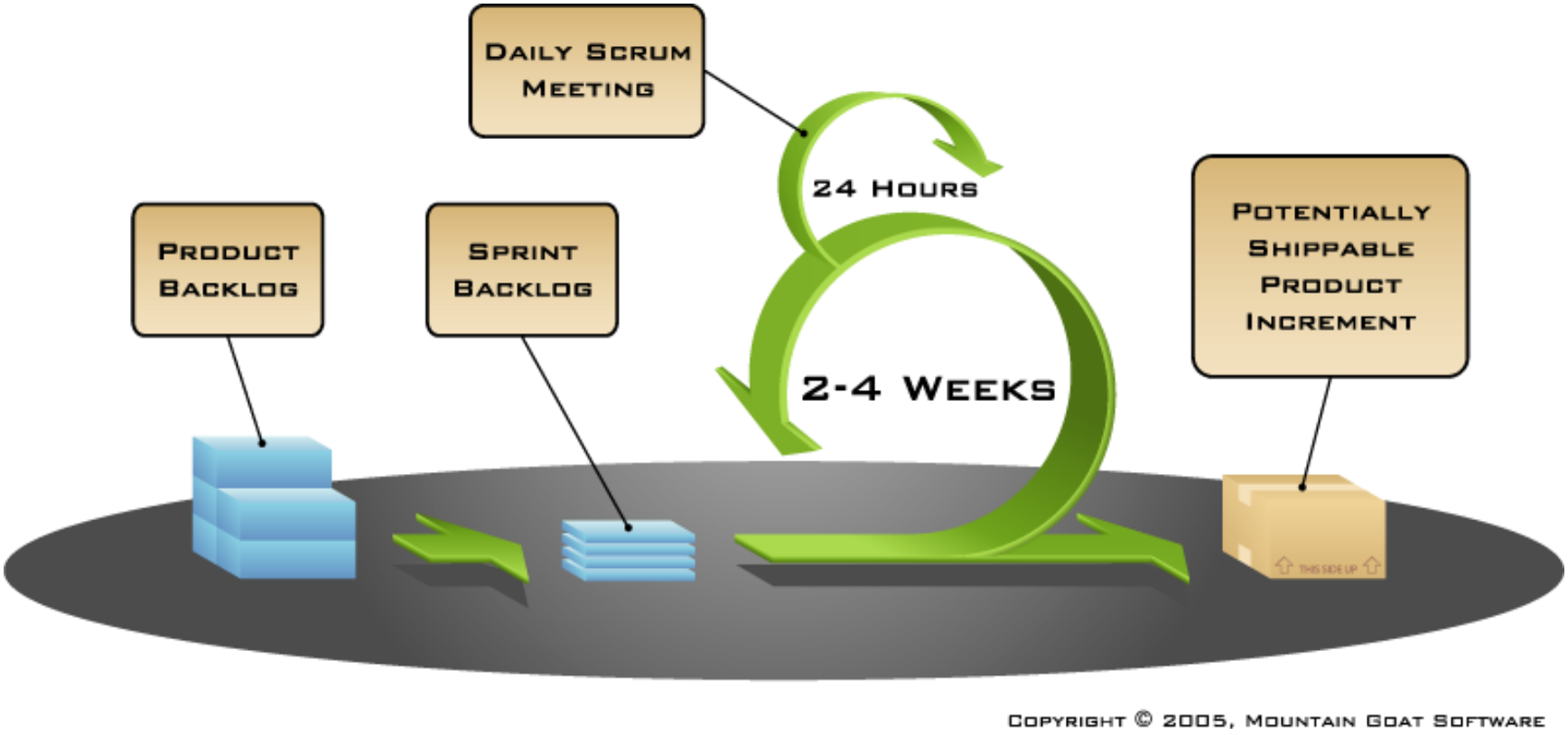
Story Confirmation

- Story tests are acceptance tests for the product owner
- Product owner specifies the story tests but will collaborate with team to create them
- Team can add additional tests
- Most tests can and should be automated (e.g. FIT, xBehave, xSpec, stress tests, load tests)

Task

- Some atomic unit of work that can be “done”.
- Development estimate includes design, development, unit test, refactoring, check in
- Usually maintained on a whiteboard or bulletin board for wide visibility
- Add testing tasks for non-functional “ility” testing
- Typically 1-4 hours effort

Scrum Overview



http://www.mountaingoatsoftware.com/scrum_figures

Scrum Team

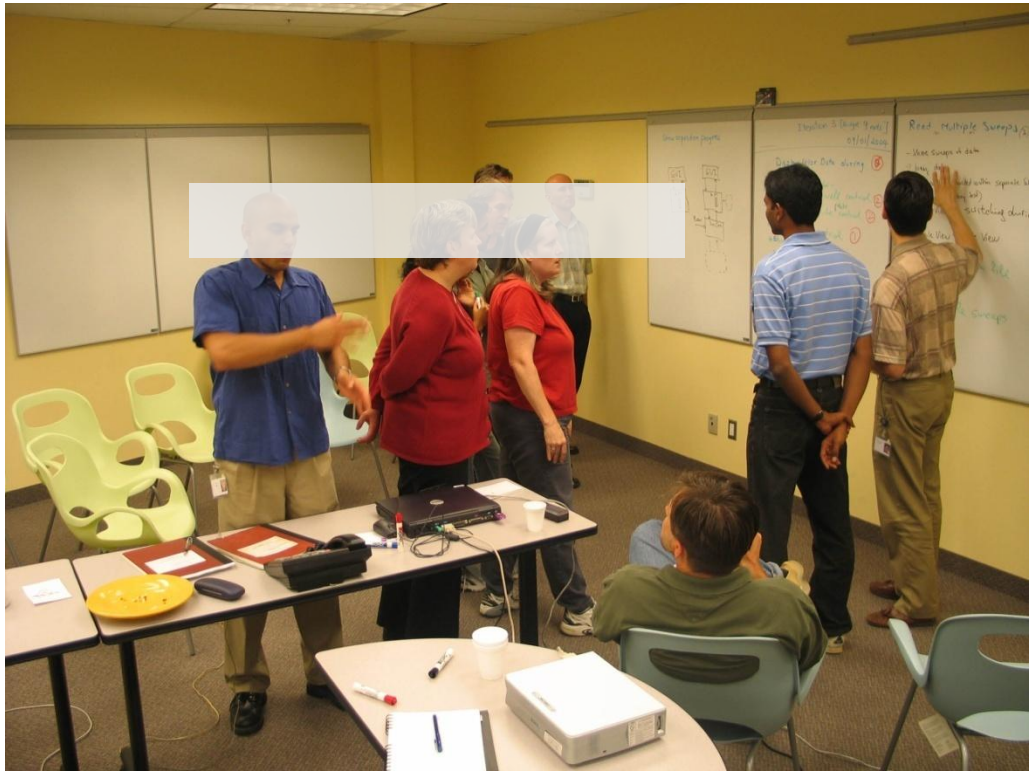
Product Owner	ScrumMaster	Team
• Feature definition • Release dates • Single decision point • Accepts or rejects work • ROI	• Represents management • Removes obstacles • Ensures Scrum process • Servant leader	• Self organizing • Cross-functional • Estimates • Tracks • Gets ‘er done

Product Backlog

- Master list of all “features”
- High priority features are split into “stories” achievable within an iteration.
- Each “story” is prioritized and scoped.

Sprint Planning Meeting

- Highest priority stories are reviewed.
- Team selects stories
- Team breaks stories down into tasks & re-estimates.
- Team commits to next iteration's deliverables.



Sprint Backlog

Daily Scrum

- Each team member describes:
  - What they did
  - What they plan to do
  - Obstacles
- ScrumMaster tracks and resolves obstacles
- 10 – 15 minutes

Sprint Demo

- Team demonstrates working software to product owner
- Product owner accepts or rejects completed work
- Result should be potentially shippable

Sprint Retrospective

- Team meets to review:
  - What is working?
  - What is not working?
- Team adds tasks for immediate actions for working better

What is Quality?

Jerry Weinberg:

"Value to some person(s)."

Testing Schools of Thought

- How much has been tested?
- Objectivity
- Branch of CS/Math

Analytic

- Controlled process
- Formal rules
- Standards & certification
- Conformance focus

Factory

- Focus on QA vs. testing
- Process watchdog
- Risk driven

Quality

Agile Helps Set Context

- Skilled mental activity
- People set context
- Testing provides information
- Value driven

Context Driven

Source: http://www.qaiasia.com/bpw/innerpages/documents/Process_Improvement/Model_and_Framework/PI_007-shrinivas_kulkarni.pdf

Context-Driven Principles

1. The value of any practice depends on its context.
2. There are good practices in context, but there are no best practices.
3. People, working together, are the most important part of any project's context.
4. Projects unfold over time in ways that are often not predictable.
5. The product is a solution. If the problem isn't solved, the product doesn't work.
6. Good software testing is a challenging intellectual process.
7. Only through judgment and skill, exercised cooperatively throughout the entire project, are we able to do the right things at the right times to effectively test our products.

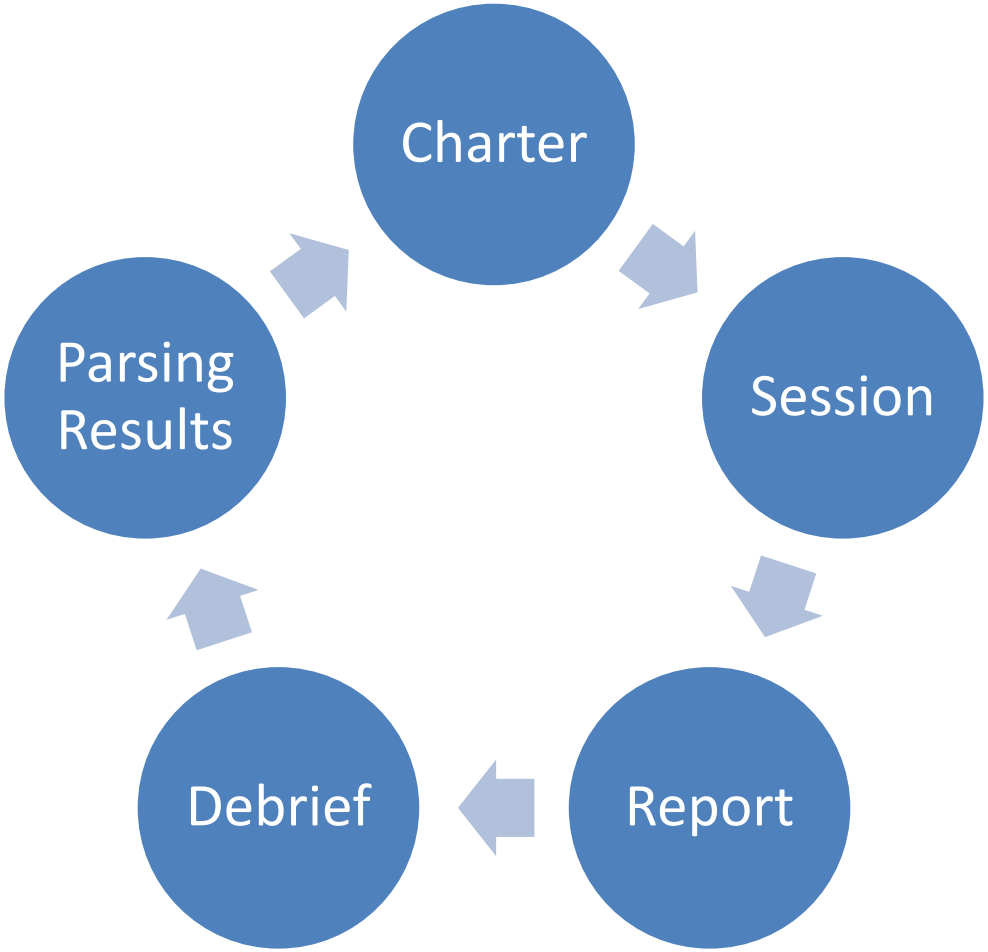
Source: http://www.context-driven-testing.com

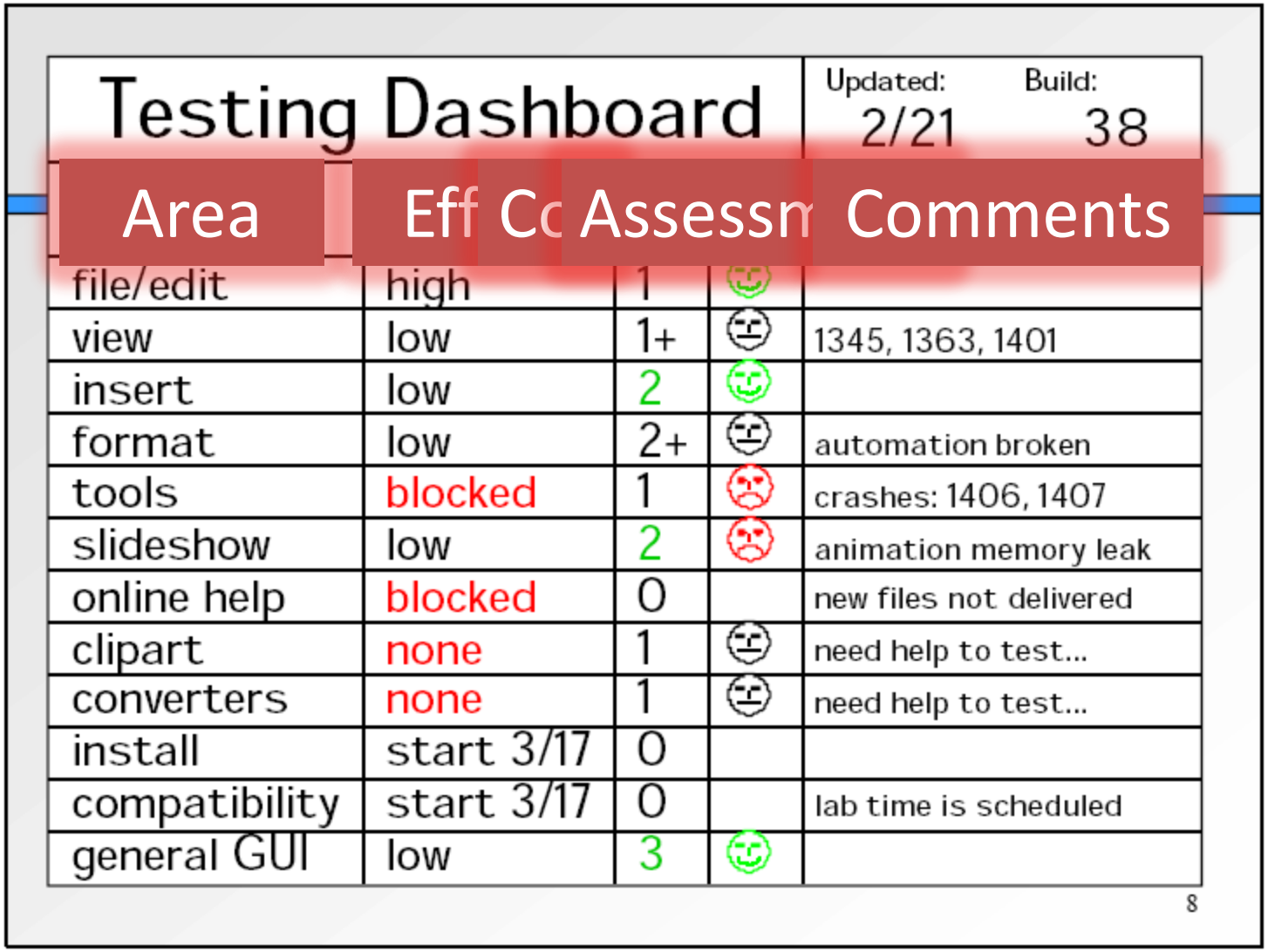
Session Based Testing

A method specifically designed to make exploratory testing auditable and measurable on a wider scale.

Source: http://en.wikipedia.org/wiki/Exploratory_testing

Session Based Structure





Source: http://www.satisfice.com/rst.pdf

Agile Quality – A Team Deliverable

Agile Practice	Benefits
Whole Team	• Quality is not just a tester responsibility • Quality is more than just testing • Testing role shifts to quality infusion throughout project life cycle
Continuous Integration	• Developers cannot check in code with failing tests
Continuous Testing	• Avoids long delays with “big-bang” testing after the “final build” • Bugs found closer to when they are introduced making them easier to fix

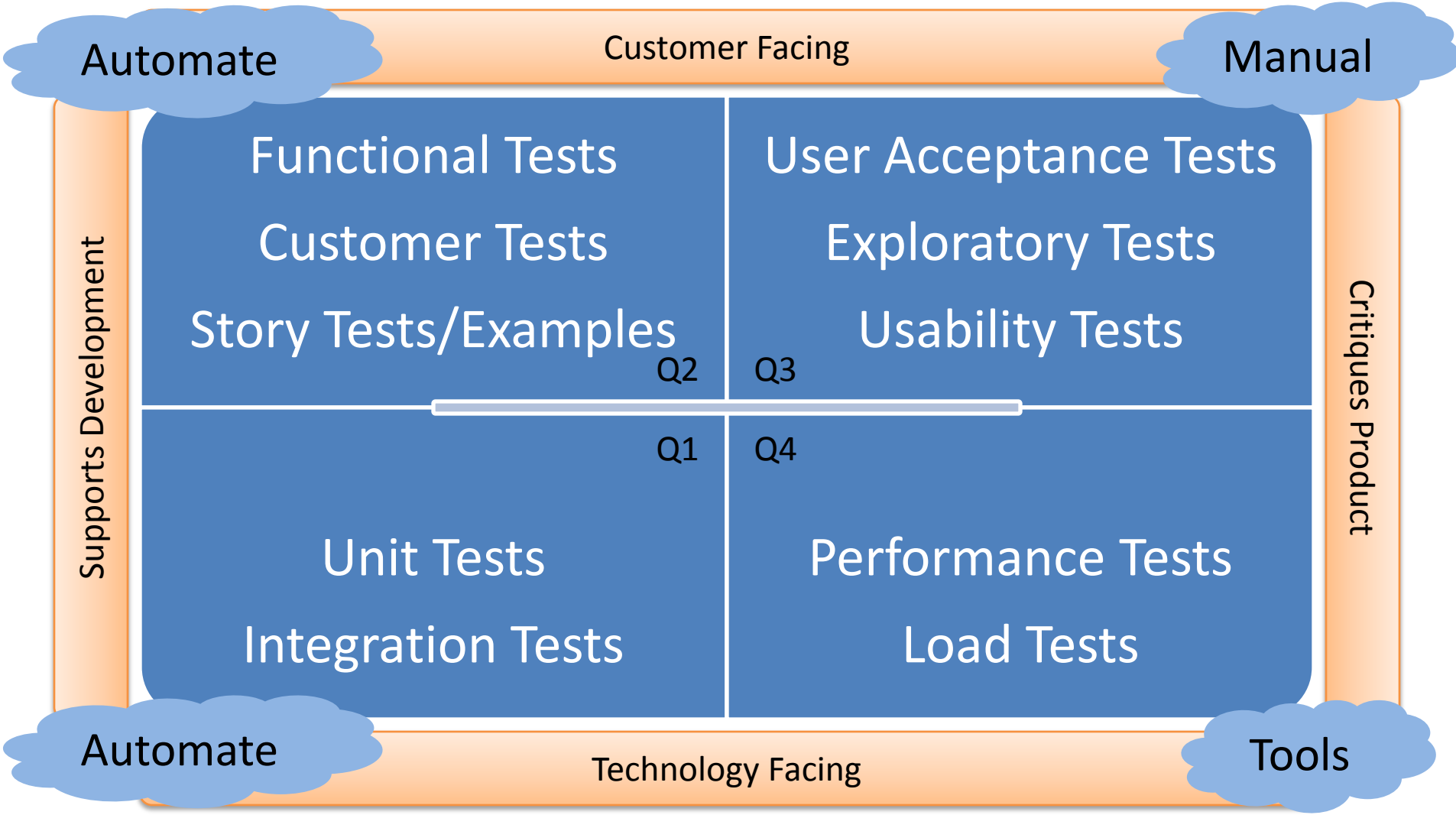
Agile Testing Challenges

- Team may not value testers
- Testers may not value team
- Unclear role of testers on team
- Testing is often squeezed as deadlines approach
- Developers and testers are often in different operational silos
- Team may not have the skills or domain expertise to develop/test effectively

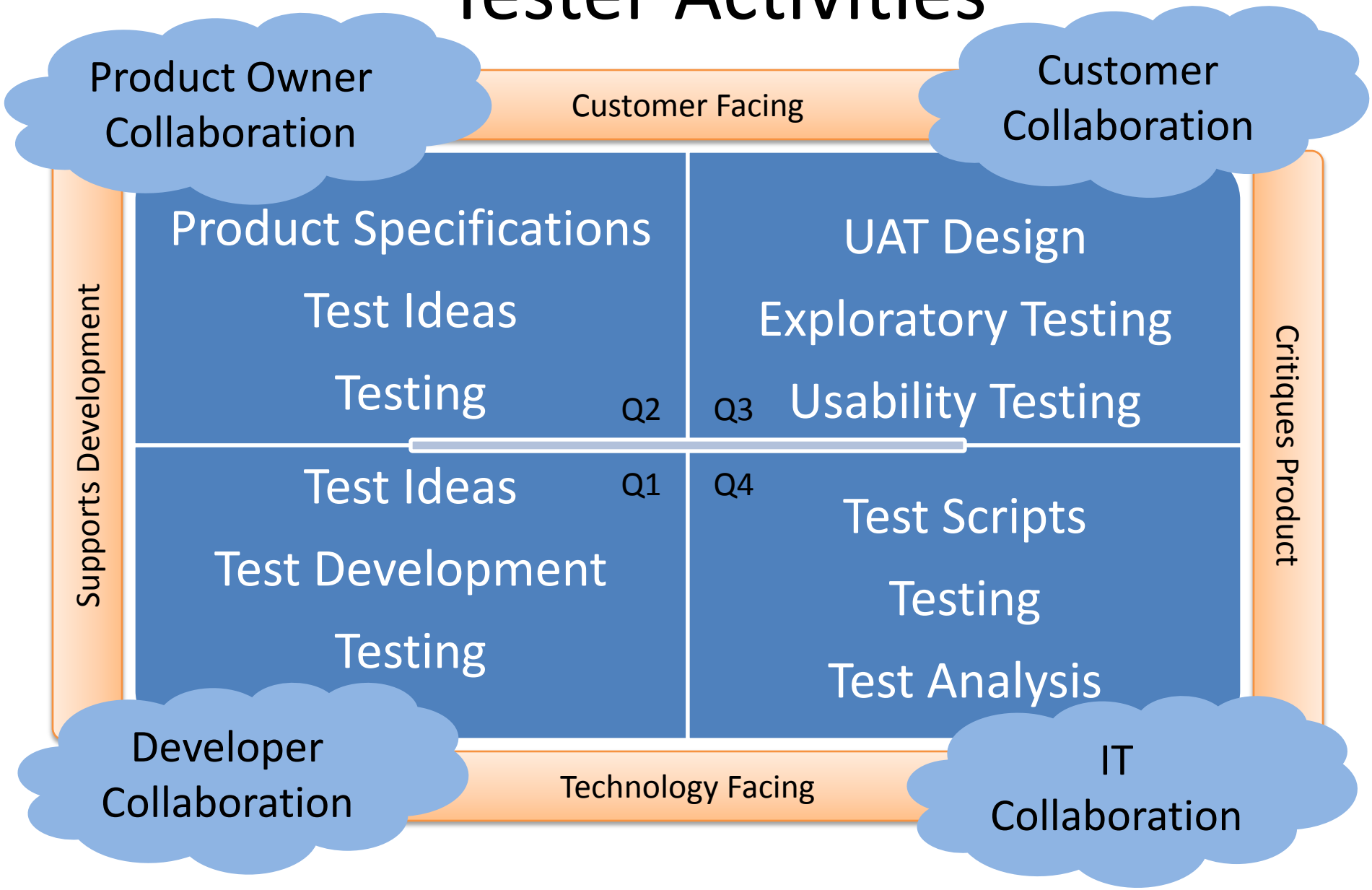
Agile Testing Approach

- Testers are first class citizens on agile teams and part of the “whole team” supporting customers, business stakeholders, developers and other team members
- Testers support quality infusion through entire team and product cycle
- Test tasks and stories are planned and executed like development tasks and stories
- Automate where possible and use session-based testing for exploratory testing
- Communicate through information radiators

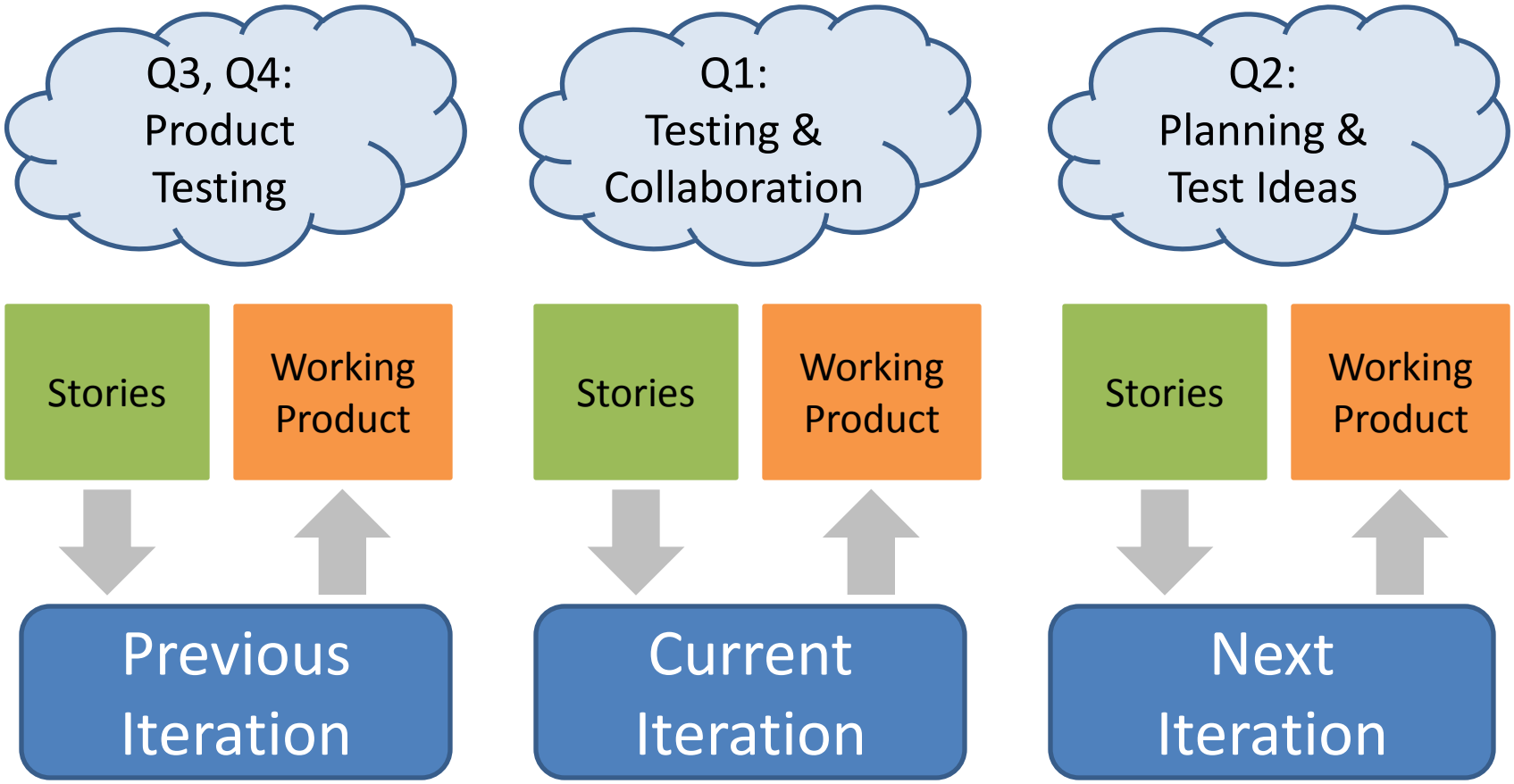
Brian Marick's Agile Testing Matrix



Tester Activities



Agile Testing Iterations

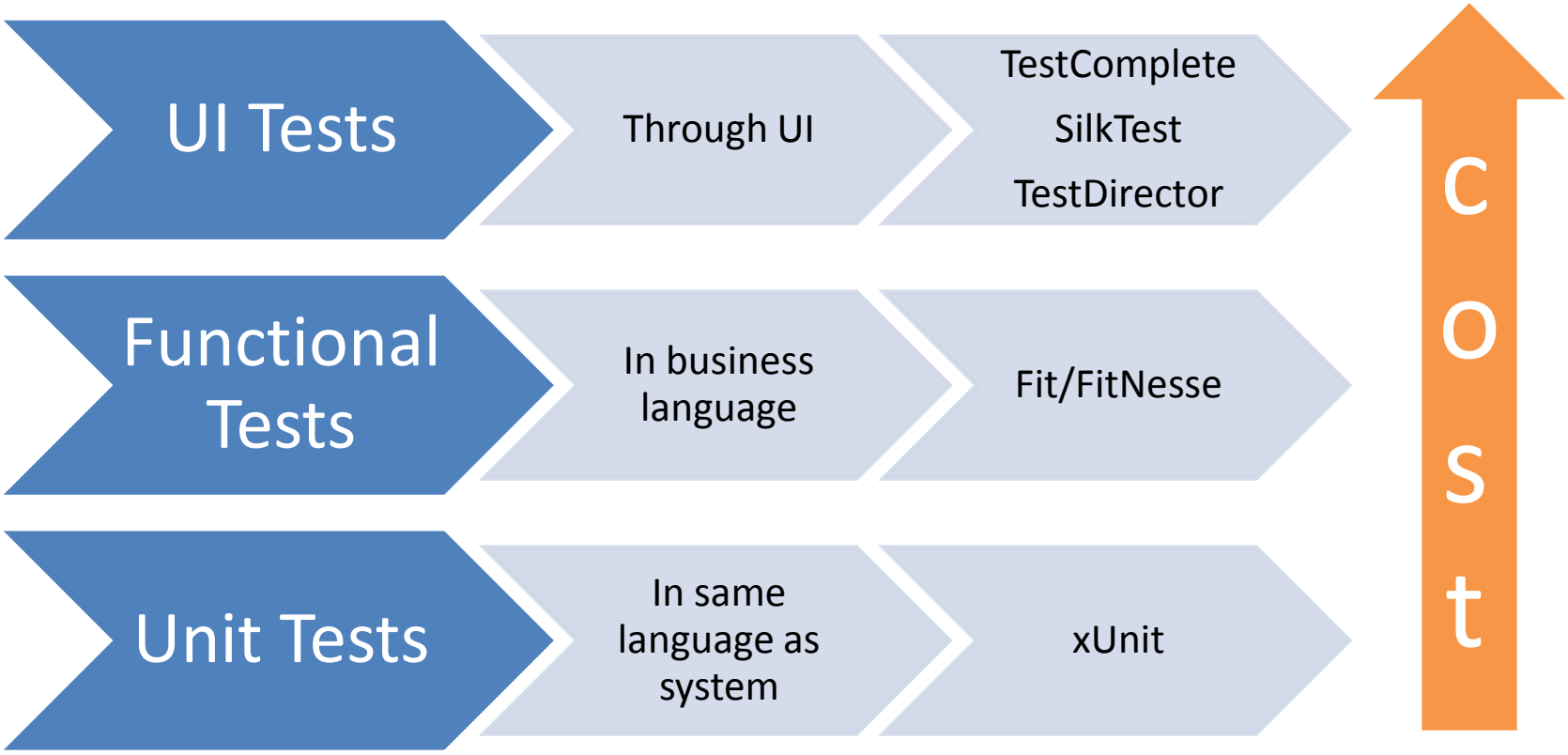


Why Automate Tests?

- Provides safety net
- Supports rapid iterations
- Provides footholds to keep notching upward
- Provides rapid feedback
- Focuses effort on what is valuable
- Frees people to do their best work

Need to balance automation costs against delivered value

Types of Automated Tests



Agile Testing Success Factors

Testers are part of the team	• Be cathedral builders not stone cutters • Collective ownership
Agile testing mindset	• Drop the “Quality Police” mindset • Focus on team goals & customer value
Automate tests	• Automate tests wherever practical • Need rapid feedback
Look at the big picture	• Balance against developer focus on technical implementation • Use agile test matrix as guide

Source: http://www.agiletester.ca/

Agile Testing Success Factors

Foundation of critical practices	• Session-based testing, agile test environments • Informative workspace
Collaborate	• Collaborate with customers • Collaborate within team
Continually improve	• Team retrospectives • Personal training: reading, blogs, QAI, local QA groups

Source: http://www.agiletester.ca/

Agile Testing Reading List

- *Agile Project Management*
  Jim Highsmith; 2004

- *Lean Software Development*
  Mary & Tom Poppendieck; 2003

- *Extreme Programming Explained 2nd Edition*
  Kent Beck, Cynthia Andres; 2004

- *Testing Extreme Programming*
  Lisa Crispin; 2002

- *Agile Testing: The Tester Role on an Agile Project*
  Lisa Crispin, Janet Gregory; Can pre-order on Amazon

Web Sites

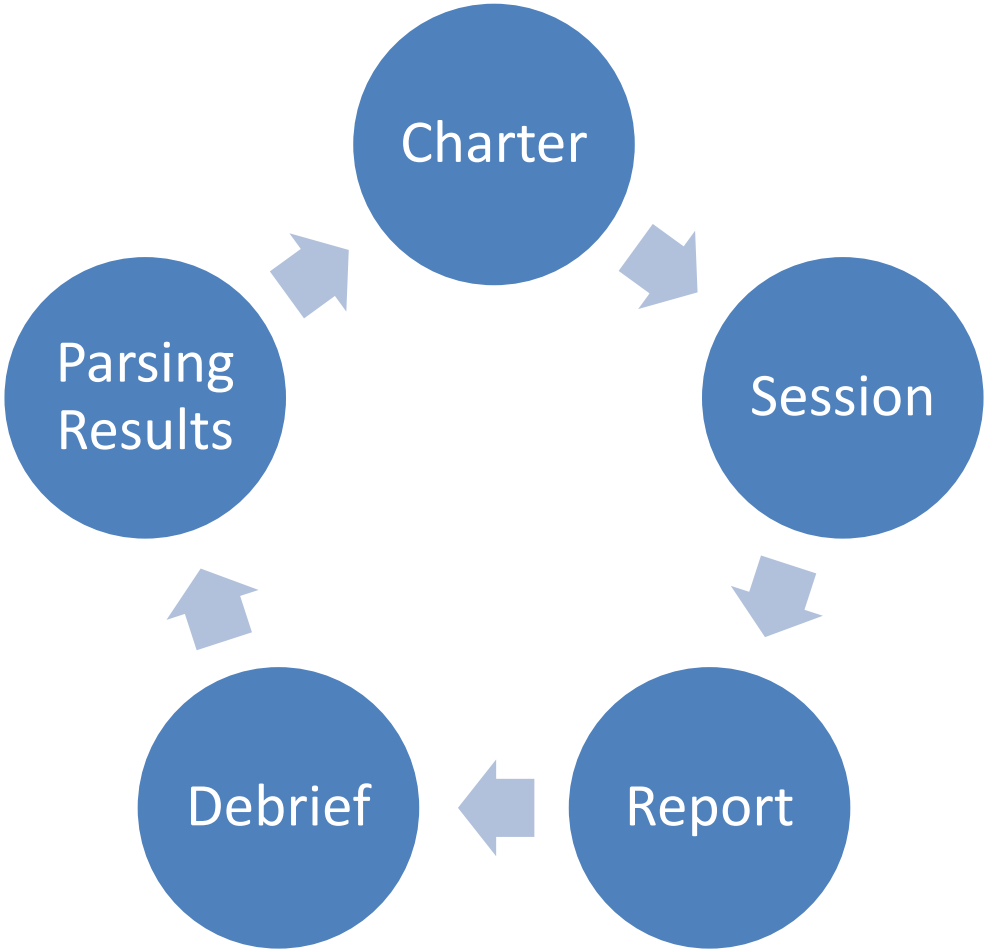
- http://www.exampler.com (Brian Marick)
- http://www.developsense.com (Michael Bolton)
- http://www.satisfice.com (James Bach)
- http://www.context-driven-testing.com
- http://www.kohl.ca/blog (Jonathon Kohl)
- http://tech.groups.yahoo.com/group/agile-testing
- http://www.agiletester.ca/ (Lisa Crispin, Janet Gregory)
- www.agilemanifesto.com
- www.agilealliance.org

Session Based Testing Extras

The following additional information on session based testing was primarily obtained or derived from a set of slides called “Rapid Software Testing” by James Bach.

Source: http://www.satisfice.com/rst.pdf

Session Based Structure



Goals

- Find bugs fast
- Fix bugs fast – through rapid testing feedback
- Visibility
- Transparency
- Auditability
- Maximize testing value

Charter

A goal or agenda for a test session in 1 to 3 sentences.

Created by the test team prior to the start of testing, but may be added or changed at any time.

Often created from a specification, test plan, or by examining results from previous test sessions.

Source: http://en.wikipedia.org/wiki/Session-based_testing#Charter

Session

An uninterrupted period of time spent testing, ideally lasting one to two hours.

Each session is focused on a charter, but testers can also explore new opportunities or issues during this time.

The tester creates and executes test cases on-the-fly and records their progress.

Source: http://en.wikipedia.org/wiki/Session-based_testing#Session

Report

- Charter
- Areas Tested
- Detailed notes on how testing was conducted
- A list of any bugs found
- A list of issues:
  - open questions
  - product concerns
  - process concerns
- Any files used
- % of time spent on charter vs. other:
- Session start time and duration
- Tester name

Source: http://en.wikipedia.org/wiki/Session-based_testing#Session_Report

Debrief

A debrief is a short discussion between the manager and tester (or testers) about the session and the session report.

Source: http://en.wikipedia.org/wiki/Session-based_testing#Session_Report

Parsing Results

With a standardized Session Report, software tools can be used to parse and store the results as aggregate data for reporting and metrics.

This allows reporting on the number of sessions per area or a breakdown of time spent on testing, bug investigation, and setup / other activities.

Source: http://en.wikipedia.org/wiki/Session-based_testing#Parsing_Results

Low Tech Testing Dashboard

Source: http://www.satisfice.com/rst.pdf

The Test Status Problem

"What's the status of testing?"

"What are you doing today?"

"When will you be finished?"

"Why is it taking so long?"

"Have you tested _____, yet?"

Source: http://www.satisfice.com/rst.pdf

Testing Reporting Challenges

- Management has little patience for detailed test status reports.

- Management doesn't understand testing:
  - *Testing is confused with improving.*
  - *Testing is considered a linear, independent task.*
  - *Testing is assumed to be exhaustive.*
  - *Testing is assumed to be continuous.*
  - *Test results are assumed to stay valid over time.*
  - *Impact of regression testing is not appreciated.*
  - *Test metrics are hard to interpret.*

Source: http://www.satisfice.com/rst.pdf

A Low Tech Dashboard Solution

Report test cycle progress in a simple, structured way...

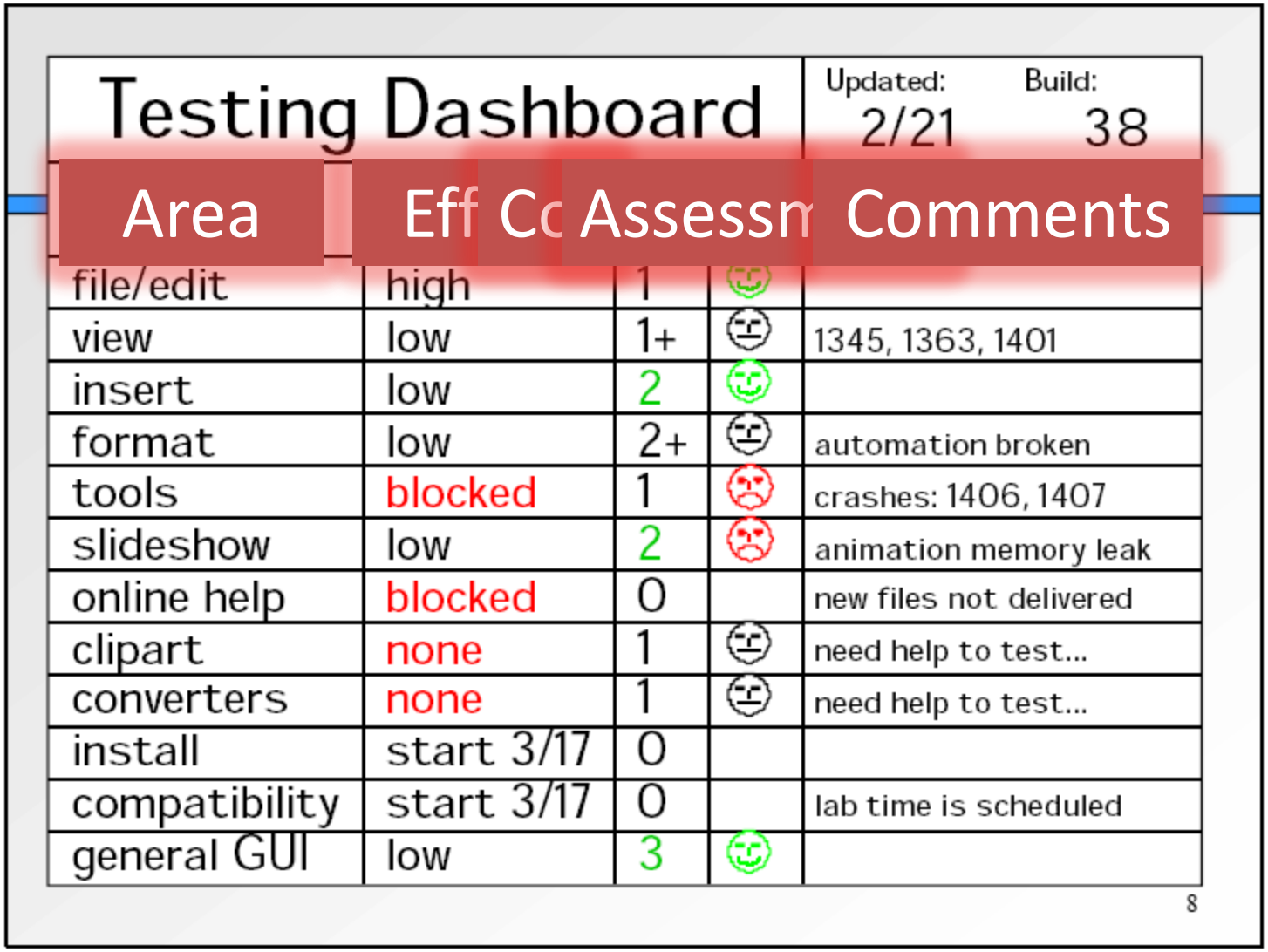
...that shows progress toward a goal...

... manages expectations...

...and inspires support...

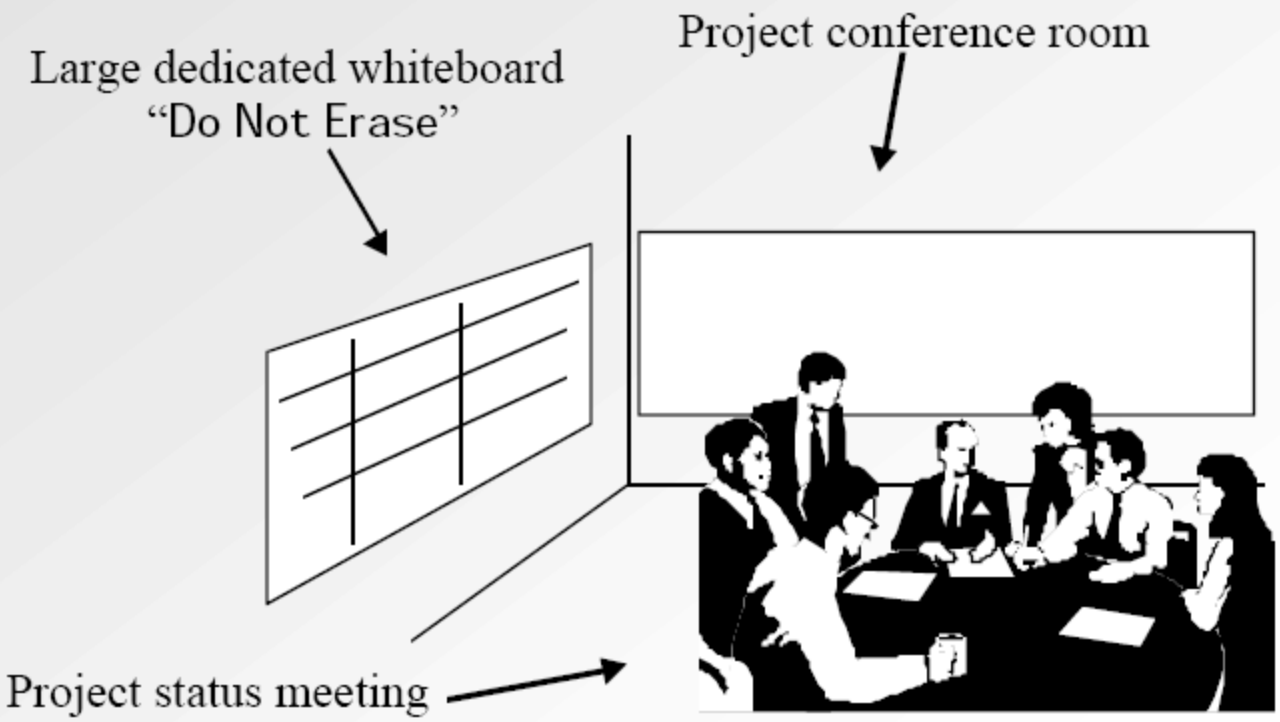
...for an effective test process.

Source: http://www.satisfice.com/rst.pdf



Source: http://www.satisfice.com/rst.pdf

The Dashboard Concept



5

Source: http://www.satisfice.com/rst.pdf

Product Area

Area
file/edit
view
insert
format
tools
slideshow
online help
clipart
converters
install
compatibility
general GUI

- 15-30 areas (keep it simple)
- Avoid sub-areas: they're confusing.
- Areas should have roughly equal value.
- Areas together should be inclusive of everything reasonably testable.
- "Product areas" can include tasks or risks- but put them at the end.
- Minimize overlap between areas.
- Areas must "make sense" to your clients, or they won't use the board.

9

Source: http://www.satisfice.com/rst.pdf

Test Effort

- Use red to denote significant problems or stoppages, as in blocked, none, or pause.
- Color ship green once the final tests are complete and everything else on that row is green.
- Use neutral color (such as black or blue, but pick only one) for others, as in start, low, or high.

Source: http://www.satisfice.com/rst.pdf

Test Coverage

- Color green if coverage level is acceptable for ship, otherwise color black.
- Level 1 and 2 focus on functional requirements and capabilities: *can* this product work at all?
- Level 2 may span 50%-90% code coverage.
- Level 2+ and 3 focus on information to judge performance, reliability, compatibility, and other "ilities": *will* this product work under realistic usage?
- Level 3 or 3+ implies "if there were a bad bug in this area, we would probably know about it."

Source: http://www.satisfice.com/rst.pdf

Quality Assessment

"We know of no problems in this area that threaten to stop ship or interrupt testing, nor do we have any definite suspicions about any."

"We know of problems that are possible showstoppers, or we suspect that there are important problems not yet discovered."

"We know of problems in this area that definitely stop ship or interrupt testing."

14

Source: http://www.satisfice.com/rst.pdf

Comments

Use the comment field to explain anything colored red, or any non-green quality indicator.

- Problem ID numbers.
- Reasons for pausing, or delayed start.
- Nature of blocking problems.
- Why area is unstaffed.

15

Source: http://www.satisfice.com/rst.pdf