

# Quality is a Team Sport

***Davyd Norris***  
***Senior IT Architect – IBM Rational***  
***dnorris@au1.ibm.com***



# Agile Method Requires Discipline

With Agile implementation comes a new way of working that requires disciplined collaboration.

In order for teams to be effective and efficient in collaborating and communicating using an Agile development methodology, a variety of approaches are used.



# Agile Collaboration Essentials for Quality Management

## • Establish good relationships

Teams are built from people from many IBM products and projects:

- Learn who your product's team members are
- Establish accessible repositories for all team members to use
- Learn what other products your product will integrate with and who to contact on each team

# Learn who your product's team members are

- Create email /distribution lists
- Know where team members are located and
- be sensitive to time-zone differences
- Create accessible repositories for change management and test tracking for all team members to use- our team uses Rational Team Concert (RTC) and Rational Quality Manager (RQM).



# Agile Collaboration Essentials for Quality Management

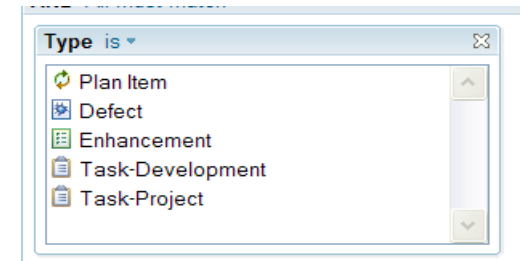
- Participation is key to staying informed
  - Project level meetings (managers and team leads)
  - Iteration meetings
  - Scrum meetings
  - Work product reviews (Test Plans, Test Cases, Product Assessment Reports)
  - Customer meetings and Beta programs
  - Cross-functional team meetings
  - Lessons learned session



## How do Rational Team Concert (RTC) and Rational Quality Manager (RQM) assist Agile?

Looking for iteration deliverables information?

- Queries and reports can be created to list iteration deliverables for both products.
  - RQM provides test asset organization, control and reporting features with many product integrations.
  - RTC provides traceability through work item assignments, ownership of work items, identifying related work items as well as parent and child work items.
  - RTC tracks several types of work items:



# Collaboration 101

## ■ Establish relationships

- New definition of “the team”: Developers, QE, UA (Doc), UX, Graphics, Install, Build
- Iteration deliverables meeting
- Cross functional team meetings (CFT)

## ■ Drive focus on customer business solutions

- Use cases from requirements drive development feature deliverables
- Customer based scenario test cases
- Role based scenario implementation of test cases (C/ALM)
- Test gap analysis of customer APARs to improve test coverage in scenarios

## ■ Address Quality

- Issue tracking and closure
- Work item subscription, email notification
- Requirements traceability to test cases (ReqPro)
- The sooner we can test it, the sooner we can provide feedback to developers

# The Team's Agile

- Change in how everyone does things
  - A customer team defines the requirements for each iteration
  - Developers estimate in a massive iteration planning meeting and volunteer for tasks they wish to do
  - 2 week iterations for testing delivered features
    - picking up new builds when major defects are fixed to validate
  - Iteration end date does NOT move
  - All features seen in the user interface (UI) work all the way through to the back end datastore



## Moments of fear and doubt

- 2 week iterations!! No one can do that
- What kind of tests can we run and when?
- Fine, I will do what I am told
- This will never work! We have no structure
- This does not look disciplined – looks like a free for all
- But we need 6 more weeks to explore the infrastructure
- How can we do system verification testing (SVT) test for each iteration?

# Rational Quality Manager project begins

## Work Item Query\*

Run Save Save Copy

Name: Plan items for QM 1.01

Conditions Details Result Layout

**Result Columns**

- Type
- Id
- Summary
- Status
- Priority
- Severity
- Modified Date
- + Add Column...

**Sort Order**

Modified Date Descending ▾

+ Add Sort Column...

## Plan items for QM 1.01



Show All ▾ Items Per Page Previous | 1 - 7 of 7 | Next Type Filter Text

Type	Id	Summary	Status	Priority	Severity	Modified Date ▾
	12330	IN-LI-15 - Support for BVT	⇒ Committed			Last Week
	12070	IN-LI-19: Common C/ALM UI Elements	⇒ Committed			Last Week
	12620	Tivoli integration: TPM, TADDM, SRM	✓ Closed			Wednesday, Apr 8
	12437	Build Forge 7.1.1 Adoption	✓ Closed			Wednesday, Apr 8
	12551	Closer parity with CQTM, RMT and RTM	⇒ Committed			Tuesday, Apr 7
	12621	Support staff/stax integration	✓ Closed			Friday, Apr 3
	11817	Changes for migrating UI to Jazz 2.0 M2	✓ Closed			Monday, Mar 23

Previous | 1 - 7 of 7 | Next

# Test Plans and Test Cases in RQM

Rational Quality Manager
ANNE M. BURKE | Log Out  
Quality Manager

- Requirements
- Planning
- Construction
- Lab Management
- Execution
- Reports
- Defects

Home
View Test Plans
RQM v2.0
RQM v1.0.1

**Manage Sections**

**Table Of Contents**

- Summary
- Business Objectives
- Test Objectives
- Formal Review
- Requirements
- Test Schedules
- Test Estimation
- Test Environments
- Test Team
- Quality Objectives
- Entry Criteria
- Exit Criteria
- Test Cases**
- Resources
- Attachments

Show All Sections

**RQM v1.0.1**

Test Plan Overview | [View Snapshots](#)

Originator: Justin J Kutticherry Action: Select Action State: Draft

Test plan for the 1.0.1 release.

Discard Changes Save

**Test Cases**

Work Item: [Create](#)

Lists the test cases associated with a given plan. You can add and remove associations to test documents and create and associate a new test case. Removing a test case will remove the association to this test plan but not delete the test case.

Group by: Ungrouped

10 Items per page Previous | 1 - 10 of 23 | Next

ID	Suspect	Name	State	Category	Function	Theme	Weight	Modified
118	◇	mplus_proj1_101	Draft	Unassigned	Unassigned	Unassigned	665	Apr 29, 2009
119	◇	mplus_proj2_101	Draft	Unassigned	Unassigned	Unassigned	745	19 minutes ago
120	◇	mplus_proj3_101	Draft	Unassigned	Unassigned	Unassigned	635	Apr 25, 2009
122	◇	SmallTeam_Story_101	Draft	Unassigned	Unassigned	Unassigned	400	Apr 24, 2009
30	◇	Migrate_CQTM	Draft	Unassigned	Unassigned	Unassigned	100	Mar 20, 2009
130	◇	MultipleProjects_Setup1	Draft	Unassigned	Unassigned	Unassigned	10	Mar 20, 2009
131	◇	MultipleProjects_TC1	Draft	Unassigned	Unassigned	Unassigned	50	Mar 23, 2009
12	◇	LM_Self_Reservations	Draft	Unassigned	Unassigned	Unassigned	5	Mar 20, 2009
13	◇	LM_Request_w_Reservation	Draft	Unassigned	Unassigned	Unassigned	30	Mar 20, 2009
11	◇	LM_Resource_CRUD	Draft	Unassigned	Unassigned	Unassigned	10	Mar 23, 2009

Previous | 1 - 10 of 23 | Next

**Test Plan Workitems**

---

**Related Sites**

- [IBM Rational](#)
- [IBM Rational Quality Mgmt](#)

# How the first two weeks went

From no code at all, to...

- ✓ Import xml file to populate database with machines
- ✓ Search for machines based off of any machine attribute
- ✓ Show a list of machines in a search results table
- ✓ Provide Context Sensitive Help
- ✗ Select a machine to see its details- missed del date by 6 hrs

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- labMachines xmlns:lm="http://www.ibm.com/testLabManagement/schema/labMachines.xsd" -->
<machine>
  <hostname>styxMachine01</hostname>
  <ipAddress>9.34.80.001</ipAddress>
  <make>IBM</make>
  <memory>1024</memory>
  <operatingSystem>Windows XP</operatingSystem>
  <processorType>Intel</processorType>
</machine>
<machine>
  <hostname>styxMachine02</hostname>
  <ipAddress>9.34.80.002</ipAddress>
  <make>Dell</make>
  <memory>2048</memory>
  <operatingSystem>Linux</operatingSystem>
  <processorType>AMD</processorType>
</machine>
</labMachines>
  
```

The screenshot shows the Lab Management application interface. The search filters are set to:

- Make: IBM
- Processor Type: Intel
- Operating System: Windows XP
- Memory (MB): 1024
- Machine Status: Available
- IP Address: (empty)

The search results table displays the following data:

Host Name	Make	Processor Type	Operating System
styx001	IBM	Intel	Windows XP
styx002	IBM	Intel	Windows XP

## What worked?

- Code containment within iterations
- Strong collaboration
- Cohesive problem solving: focus on the “now”
- Documentation for features was delivered
- Individual contributions were recognized and valued
- Install is a critical part of our product efforts
- SVT participation early in development cycle to provide feedback sooner

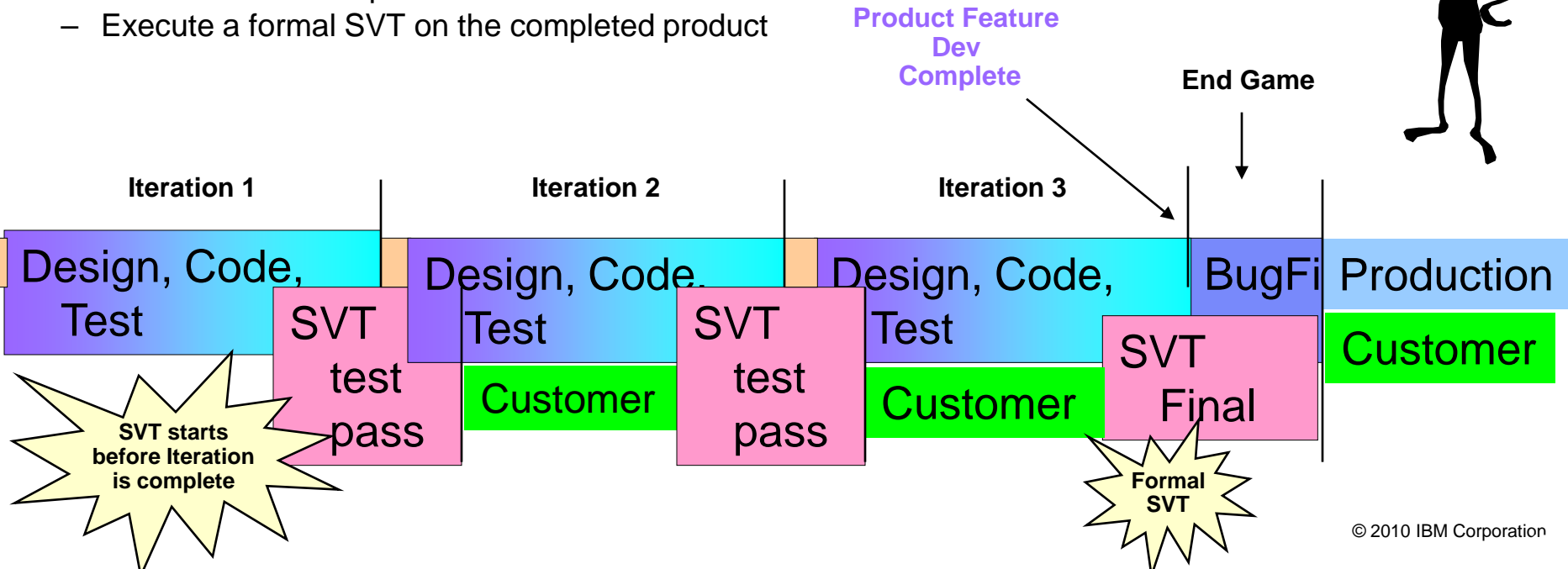
## How System Verification Test Fits into Agile:

- Run on Milestone builds
  
- N Iterations = 1 Milestone
  - Aka  $I1 + I2 + I3 = M1$ ,  $I4 + I5 + I6 = M2$
  
- This allows for planning and test case development during a time when
  - delivered code is available to see running
  - some code is in progress and can be seen on a developer's machine
  - the features intended to be delivered can be seen on storyboards or in plan items.

# Agile SVT approach

- Basic concepts

- Execute system-level testing on all iterations with end to end stories
- Testing may overlap development activities (either on current or next iteration) especially for performance and reliability runs
- Focus on functional capability, usability, consumability, basic performance, and regression testing. While writing testcases using iterations, open defects. Be an early tester in a pair environment – even during prep activities.
- Track successes, failures, defect status, and issues during the iteration test pass
- Provide SVT feedback to development prior to all code deliveries to actual Customers
  - \* Defects
  - \* Assessment reports
- Execute a formal SVT on the completed product



## What does the SVT Schedule look like?

Key Project Milestones	Plan	Outlook / Actual	Completion / Comments
Plan DCP approval	03/10/09	03/10/09 A	Complete
SVT Test Plan Approval	03/20/09	04/09/09 A	Complete
Milestone Test Entry - M2/M3D1	03/23/09	03/23/09 A	Complete
Milestone Test Exit - M2/M3D1	04/17/09	04/17/09 A	Complete. 95% attempted, 88% completed.
Milestone Prep Test Entry – M3	04/20/09	04/20/09 A	Complete
Milestone Prep Test Exit – M3	04/24/09	04/24/09 A	Complete
Milestone Test Entry – M3	04/27/09		Green
Milestone Test Exit – M3	05/08/09		Green
SVT Prep Start	04/20/09	04/20/09 A	Complete
SVT Prep End	05/11/09		Green
Final DCUT	05/04/09		Green



# The Team

## ■ Issues

- People find it hard to let go of original team members and to let new people in
- Lack of bonding
- Hard to see the BIG picture of how all of the team fits together
- How do we know the customer will want this

## ■ Solutions

- No more “teams” – 1 team, areas have contributors
- Test deliverables are reviewed by many contributors
- Experience/Assessment reports are generated by test team for all to review their evaluation of the product’s health
- People volunteer for comfort zones and people test out new areas and team mates mentor
- Celebrations during a lessons learned meeting to discuss what went well and what to improve
- Formation and participation in customer meetings and beta programs with real customers providing guidance



# Accomplishments

- People are happy choosing what to work on and learning new things
- SVT has figured out the rhythm in the schedule for test development and execution
- Communication between Customer Team and the rest of the team is now on track - storyboards make a difference!!
- We are grateful for our awesome install team!

## Take this home

- Short Iterations based off of estimates done by the team result in code delivered and tested that is consumable
- Complete and detailed storyboards completed before the iteration drives success
- Customer focus on requirements avoids rework and redirection later on

## And this....

- The Team is comprised of Development, QE, Documentation, User Experience, Graphics, Install...
- QE loves being involved early and the product is delivered to SVT with higher quality when QE can influence the design of features
- SVT testing early eases test burden
  - Creates building blocks for suites and formal SVT entry...reuse test artifacts for regression testing (ROI)
- Celebrate your successes and acknowledge everyone's contributions

# Questions



© Copyright IBM Corporation 2009. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.