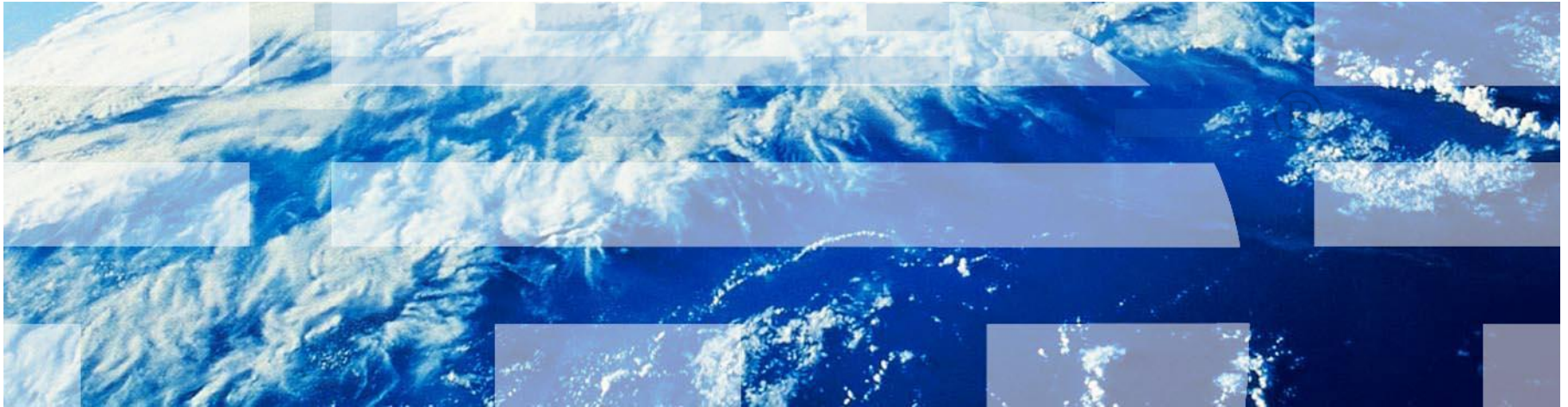


# IBM Rational Solution for Collaborative Lifecycle Management



# Introductions

- Please introduce yourself
- Name and organization
- Current integration technologies/tools in use



*What do you want out of this Exploration session?*

# Agenda

## ▪ Introduction to Collaborative Lifecycle Management

- Lab Overview
- Module 1: Aligning plans
- Module 2: Responding to a change request
- Module 3: Planning the sprint
- Module 4: Completing a story
- Module 5: Testing and fixing a story
- Module 6: Stabilizing the sprint
- Session Summary



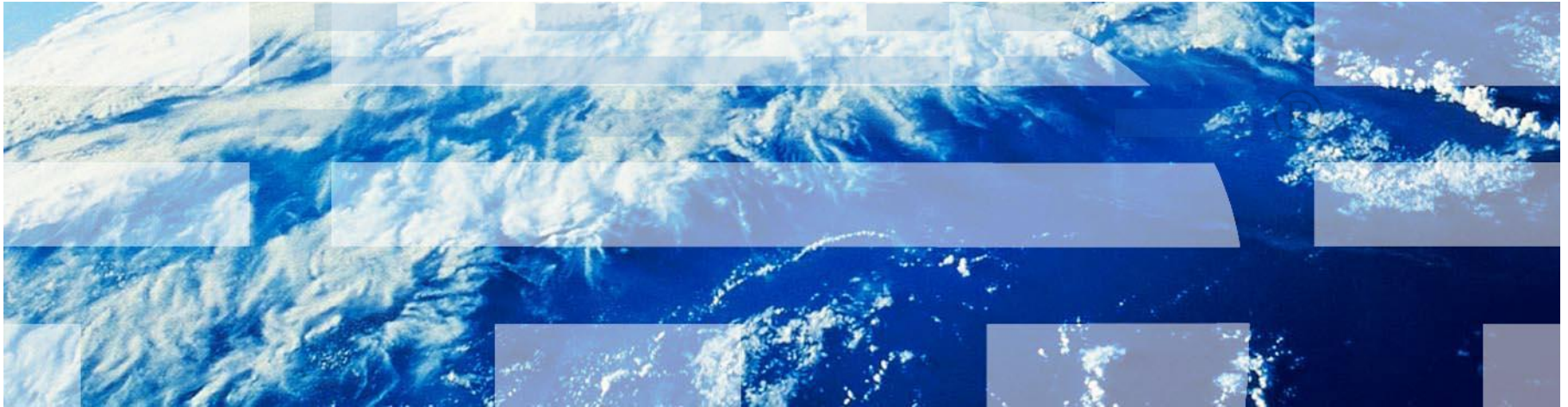
## Objectives

- Explore how the IBM Rational Solution from Collaborative Lifecycle Management:
  - Enables teams to **collaborate in real time in the context** of the work they are doing, especially in globally diverse environments.
  - Enables projects to be managed more effectively by providing visibility into **accurate project health information** drawn directly from actual work.
  - Automates traceability and auditability by **managing artifacts and their inter-relationships across the lifecycle**, empowering teams to deliver more value.
- Provide a hands on experience using IBM Rational Requirements Composer, IBM Rational Team Concert™, IBM Rational Quality Manager, and IBM Rational Software Architect Design Manager to automate the software delivery process.

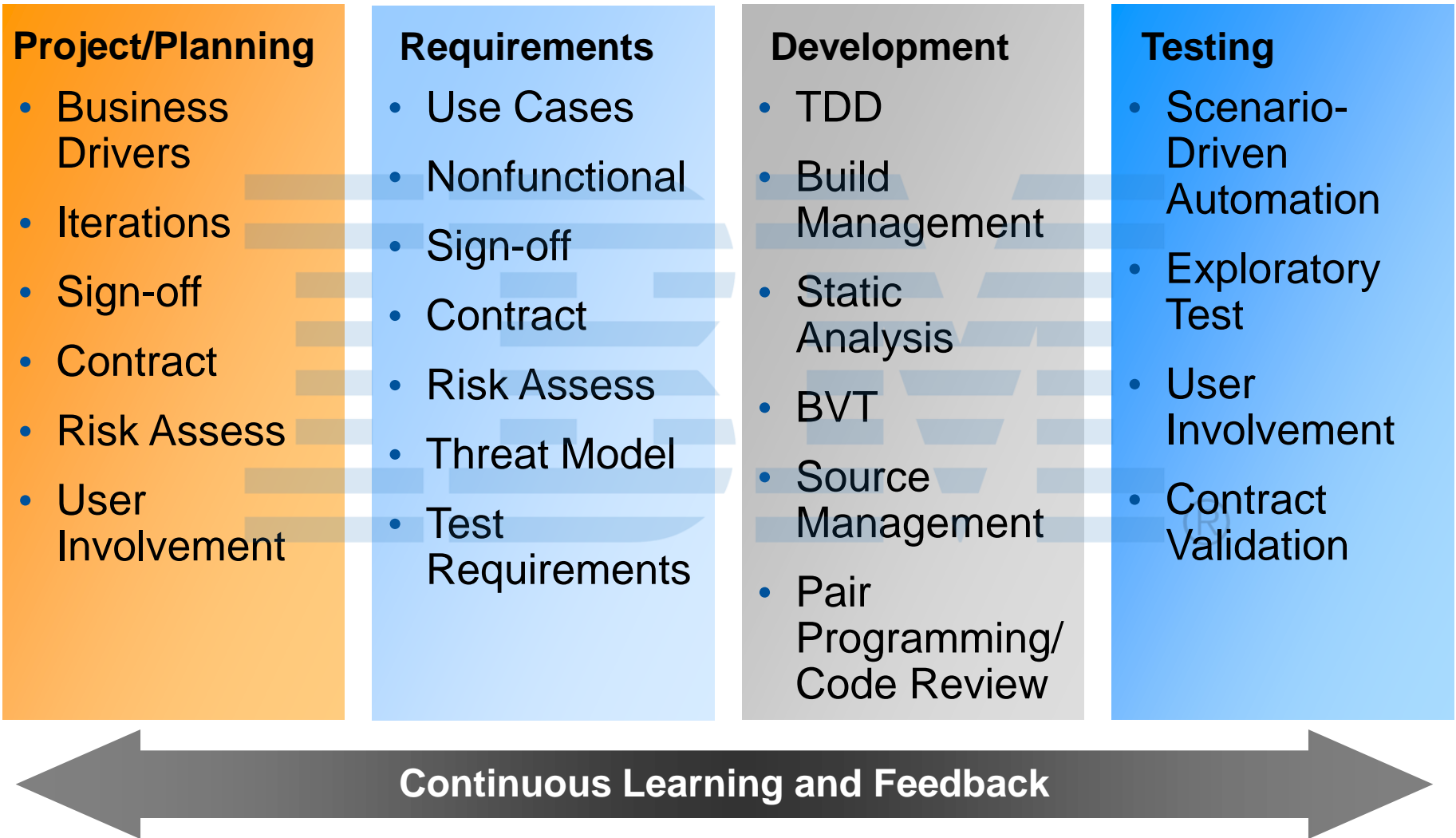




# Introduction to Collaborative Lifecycle Management (CLM)








## ALM is about connecting the disciplines



Source: Gartner Application Architecture, Development & Integration Summit Presentation, The Future and Present of AD, Thomas E. Murphy, December 2008

## Five Imperatives for Effective Application Lifecycle Management to improve organizational productivity



-  Maximize product value with **In-Context Collaboration**
-  Accelerate time to delivery with **Real-Time Planning**
-  Improve quality with **Lifecycle Traceability**
-  Achieve predictability with **Development Intelligence** <sup>®</sup>
-  Reduce costs with **Continuous Improvement**

Learn more at [ALM Everywhere](#)



# In-Context Collaboration always shows the latest

Comments (4)

- Bob to Deb, Marco 5 minutes ago (2 replies)  
**Percentage format - numer of decimal places**  
 How many decimal places should we support for percent...  
 Example: none - 16%, 1 decimal place 16.5%
- Marco to Bob, Deb 5 minutes ago  
**RE: Percentage format - numer of decimal places #1]**  
 I checked the DB table. We did not take into account decimals for this percentage. If we are going to support decimals I need to know ASAP.
- Deb to Bob, Marco Now  
**RE: Percentage format - numer of decimal places #2]**  
 Bob, is no decimal support acceptable to the business...
- Tanuj to Bob 1 minute ago  
**What is the minimum percentage?**

Threaded discussions on requirements

View All Artifacts

Page size: 20

ID	Name	Artifact Type	Last Modified By	Last Modified Date
44	Accounts Overview	Part	JTSAdmin	Mar 26, 2011 5:43:06 PM
50	Transaction History	Part	JTSAdmin	Mar 26, 2011 5:43:10 PM
53	Accounts Overview (Home Page)	Sketch	JTSAdmin	Mar 26, 2011 5:43:11 PM
55	Dividend Contribution - screen flow	Screen Flow	JTSAdmin	Mar 26, 2011 5:43:11 PM
60	Account Details	Part	JTSAdmin	Mar 26, 2011 5:43:13 PM
67	Dividend Contribution	Storyboard	JTSAdmin	Mar 26, 2011 5:43:15 PM
75	Donor must be registered user to access account details	Business Rule	JTSAdmin	Mar 26, 2011 5:43:19 PM
79	Donors will receive confirmation and receipt	Feature	JTSAdmin	Mar 26, 2011 5:43:21 PM
89	Dividend contribution - confirmation	Sketch	JTSAdmin	Mar 26, 2011 5:43:23 PM

Recent discussions highlighted on requirements

Work Items | Tag Cloud | Problems

Found 9 work items - Stories (current sprint)

Id	Status	P	Story Points	Summary	Owned By
55	New	13 pts	Frequency of dividend transfer	Deb	
59	Implemented	8 pts	Requests sent in form of email	Marco	
60	New	3 pts	<b>Organization must identify how much money is desired</b>	Marco	
62	New	1 pt	Organizations may apply with an initial request	Marco	
64	In Progress	2 pts	Customers can Nominate an Organization	Deb	
66	New	1 pt	Organization must provide justification for why funds are needed	Marco	
69	New	2 pts	Organizations can Apply	Marco	
70	In Progress	5 pts	<b>Donor Dividend Allocation Criteria</b>	Deb	
71	New	1 pt	JKE Charity Coordinator will respond to request in the website trigger...	Marco	

Unread work Items bolded for developers

# Real-Time Planning: One plan with multiple views

*Everyone sees meaningful data in the context of their work*



1 **BRM Sprint 2 (1.0) Plan** 26 items: 23 open, 3 closed | Ends in: 3 days

Plan Details Edit

Planned Items Links Snapshots Dashboard Notes

View As: Roadmap (3 items filtered) Add Work Item

Actions	Summary	Effective Estimate	Owned By
	Frequency of dividend transfer	--	Unassig
	Requests sent in form of email	--	Unassig
	Organization must identify how much money is desired	--	Marco
	Implement - Organization must identify how much money is desired	1.5 days	Marco
	Organizations may apply with an initial request	--	Marco
	Implement - Organizations may apply with an initial request	4 hours	Marco
	Customer must create an Organization	--	Deb
	Implement - Customer must create an Organization	--	Deb
	Organization must identify how much money is desired	--	Marco

2 **BRM Sprint 2 (1.0) Plan** 30 items: 25 open, 5 closed | Ends in: 6 days

Plan Details

Planned Items Links Snapshots Dashboard Notes

View As: Taskboard

Story	Open	In Pro
Donor Dividend Allocation Criteria	Implement - Donor Dividend Allocation Criteria	
Frequency of dividend transfer		Implement - Frequency of dividend transfer

3 **BRM Sprint 2 (1.0) Plan** 30 items: 25 open, 5 closed | Ends in: 6 days

Plan Details (R)

Planned Items Links Snapshots Dashboard Notes

View As: Planned Time

Bob	Closed Items: 0   Open Items: 1	Load: 0/8   +8 h
CP	Closed Items: 0   Open Items: 1	Load: 0/8   +8 h
Deb	Closed Items: 3   Open Items: 5	Load: 168/8   -160 h
Marco	Closed Items: 2   Open Items: 14	Load: 92/8   -84 h
Tanuj	Closed Items: 0   Open Items: 0	Load: 0/8   +8 h



# Lifecycle Traceability

## Team leads can improve quality and predictability

- ✓ Creating a shared vision delivers what the stakeholders want
- ✓ Whole team buy-in improves team trust, efficiency and focus
- ✓ Everyone knows what work is needed by whom and when.
- ✓ Traceability provides insight to knowing when you are DONE!

**Developers understand the requirements, test criteria and test results**



**Teams collaborate and clarify the details of requirements**



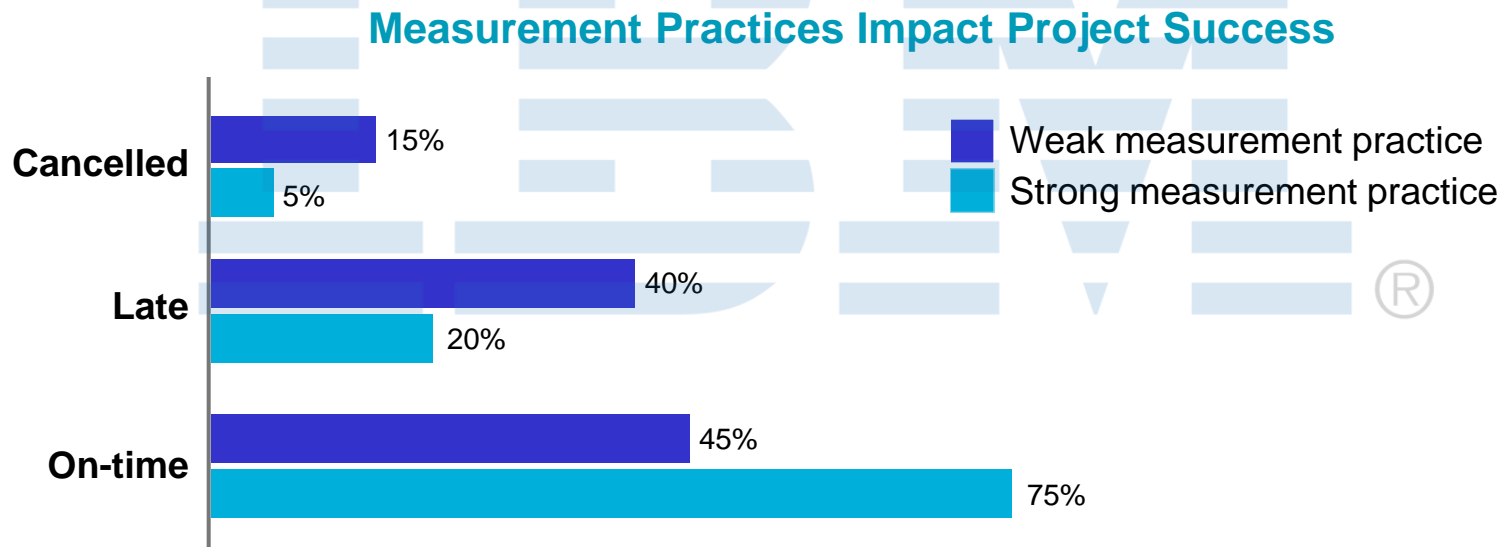
**Testers define and execute tests cases with a clear understanding of requirements**



# Development Intelligence improves predictability



- Enable **fact-based decision making** (to communicate status, monitor progress, diagnose problems, identify corrective actions)
- **Steer projects and programs** to deliver on-time
- Apply **Business Intelligence techniques** to software and systems development



Sources: Capers Jones, *Measurement, Metrics and Industry Leadership*, Copyright © 2002-2007 by Capers Jones LLC  
Capers Jones, *Software Engineering Best Practices*, Copyright © 2010 McGraw-Hill



# Use dashboards to provide that one view of project health



## JKE Banking (Change Management)\* ?

Overview | Time to Value (Schedule) | Product Value | Product Cost | Product Quality | Reports | Stabilization +

### Story Status Current Iteration

Shows the status of all stories planned for the current iteration

Status	Count
Testing	4
Implementing	18
Invalid	2
Done	18
Ready to Test	6
New	14
Ready to Implement	1

### Burndown Current Iteration

Shows the remaining amount of estimated work in hours of work items planned for the current iteration.

1,935

### Stories Open/Closed/In Progress

Shows the number of stories which are open, in progress, done during the iteration.

63

### Current Stories (80)

- 150501: Improve Web UI error reporting
- 154382: Completion (M14) of foundation support for migrating RRC 2.x to 3.0, and JTS consolidation
- 155918: Provide TVT testcases and testcase updates to TVT team
- 154263: [Continuation] Create server diagnostics tests and reports that make various checks
- 155024: Track the fixing of hangs occurring on shutdown of the CLM server
- 155398: [Repository - M14] Performance, quality and robustness enhancements
- 156051: Improvements and fixes for CLM migration automation
- 154829: [Repository - M14] Address high severity defects, Reduce backlog
- 155033: Explore servlet dispatching enablement as a backup option
- 155486: [Repository M14] Serviceability and Consumability Improvements

Page 1 of 6

### Burndown

2,776

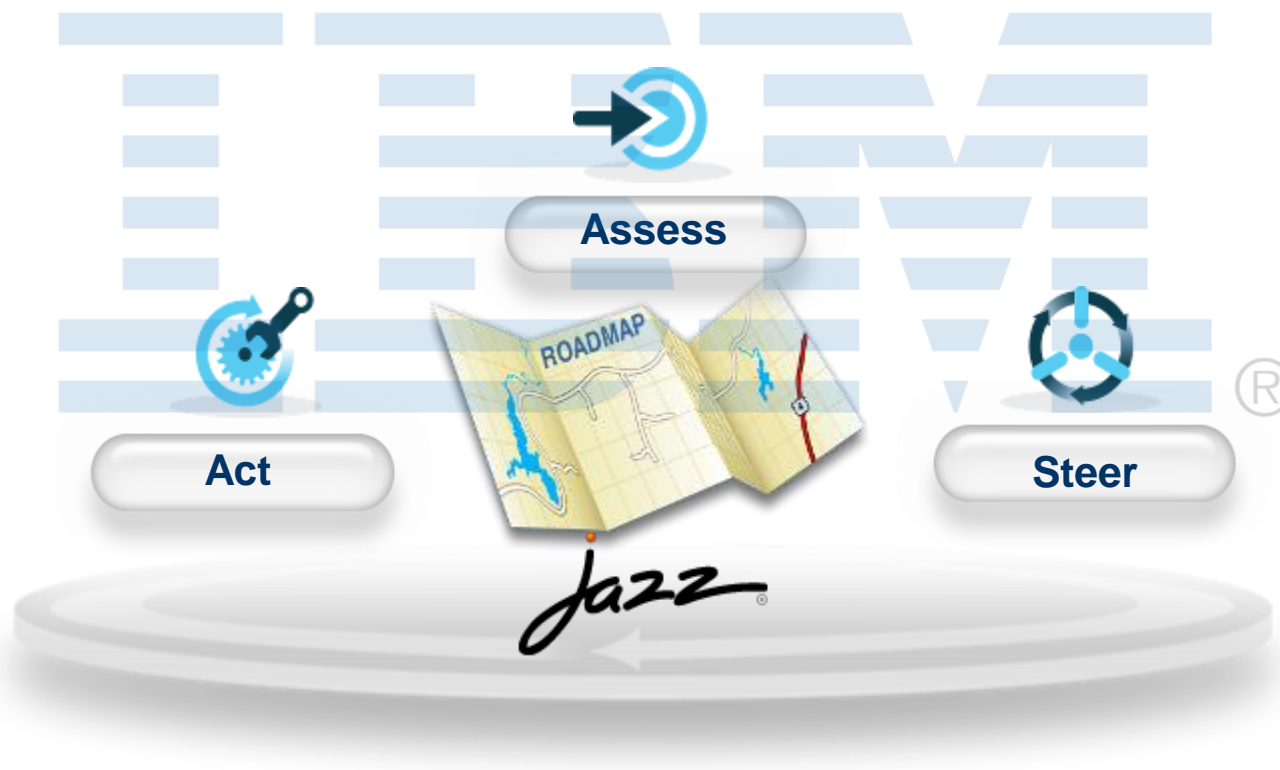
### Stories Open, Closed, In Progress

67



## Continuous Improvement reduces cost

- Improve software delivery through the ongoing **adoption of best practices and automation** to reduce manual, non-creative and error prone tasks
- Promote **incremental improvement of a project** when needed
- Enable breakthrough improvement by **capturing best practices and reusing** across teams
- Allow **everyone to participate** with easy to adopt best practices at your fingertips.



# Team retrospectives help identify areas of improvement



- Tracking Retrospectives help team members to collaborate on improvements
  - Ranking of process improvements help to focus on where the team feels the biggest pain

The screenshot shows the IBM Jazz Change and Configuration Management (ccm) interface. The main window displays a list of retrospectives under the heading "Retrospectives". The list has columns for Id, Status, Summary, Filed Against, Planned For, and Actions. Three items are visible:

Id	Status	Summary	Filed Against	Planned For	Actions
77	New	Builds are not completing as frequent as before	Build	Unassigned	
32	New	Retrospective for Sprint 2	JKE	Sprint 2	
31	Done	Retrospective for Sprint 1	JKE	Sprint 1	

A detailed view of item 31, "Retrospective for Sprint 1", is shown in a pop-up window. It includes the following information:

- Status: Done
- Filed Against: JKE
- Planned For: Sprint 1
- Project Area: JKE Banking (Change Management)
- Resolution Date: March 26, 2011 5:44 PM
- Owned By: Marco
- Resolved By: Marco

Below the list is a "Discussion" section with 4 comments. The comments are:

- Tanuj** Sep 1, 2011 1:22 PM: I have some concerns about late-in-the-cycle deliveries causing code churn. We had some challenges with the retesting required late in the sprint.
- Tanuj** Sep 1, 2011 1:24 PM: Should've also mentioned that it's been hard to find information on the right build for the fix and to understand where to find the latest good build.
- Deb** Sep 1, 2011 1:42 PM: @tanuj - I don't think we've had a lot of disruptive changes at the end of the sprint. Let's talk about this more in depth at the meeting tomorrow. I do agree @tanuj that tracking down build status has been too hard.
- Bob** Sep 1, 2011 1:45 PM: In the meeting, I'd like us to discuss some issues where requirements were not well understood and that required some things to be reworked during this sprint(which may contribute to @tanuj's impressions in comment 1). Hopefully, as a team we can come up with some ideas on reducing this confusion.

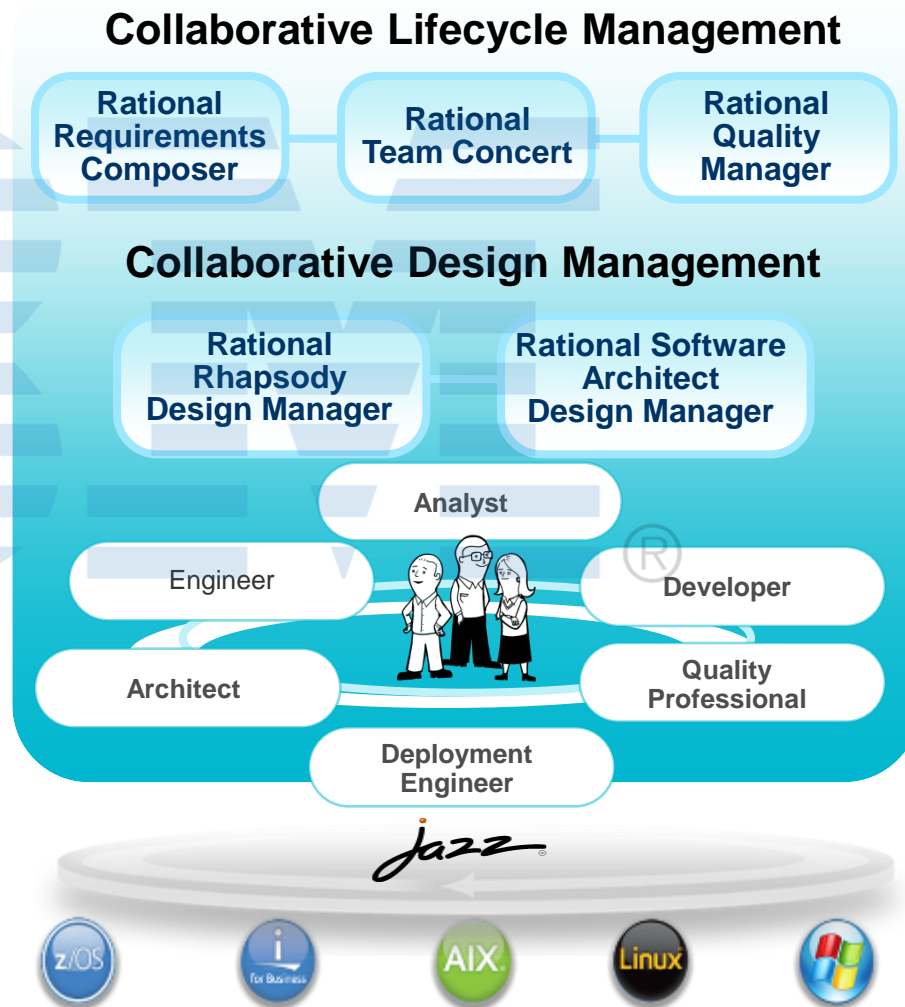
# Core Application Lifecycle Management offerings

## IBM Rational ALM Solutions

*Get Up and Running Quickly*

Reduce the costs of inefficient, multiplatform software development with integrated ALM solutions!

- Optimize your team's productivity through the 5 ALM Imperatives
- Collaborate across teams and create deep integrations across the lifecycle
- Extend as your needs evolve with role-based licensing
- Unify your infrastructure and protect your current investments with a single, open, extensible platform



# Open Services for Lifecycle Collaboration (OSLC)

*Working to standardize the way software lifecycle tools share data*

## What is OSLC?



**Open Services for Lifecycle Collaboration**  
 Lifecycle integration inspired by the web

- ▶ Community Driven – @ **open-services.net**
- ▶ Specifications for numerous disciplines
  - ▶ Such as, ALM, PLM and DevOps
  - ▶ Defined by scenarios – solution oriented
- ▶ Inspired by Internet architecture
- ▶ A different approach to industry-wide proliferation



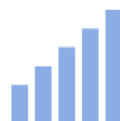
## How does it work?



Inspired by the web



Free to use and share



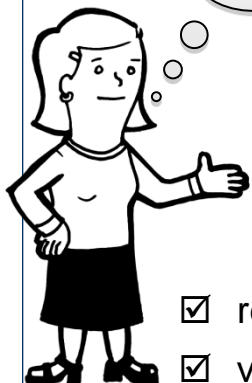
Changing the industry

**GET INVOLVED AND CONTRIBUTE!**

# Get Started! Collaborative Lifecycle Management

**Get involved at jazz.net!**

**Free Downloads!**



- read articles & blogs
- view presentations
- watch videos
- listen to podcasts
- see a project dashboard
- review the release plan

**Try it out at jazz.net!**

**1 Download** RTC, RRC or RQM for CLM

<https://jazz.net/downloads/rational-team-concert/>

Sample scenarios for CLM

[https://jazz.net/wiki/bin/view/Main/MTM\\_Lifecycle\\_Welcome](https://jazz.net/wiki/bin/view/Main/MTM_Lifecycle_Welcome)

**Track our progress at jazz.net!**

We openly develop our software with our own tools!

<https://jazz.net/projects/clm/>

**Interactive White Board**



Short, simple way to share the

*5 ALM imperatives*

[ibm.co/alm-everywhere](http://ibm.co/alm-everywhere)

**Role-based Demo**

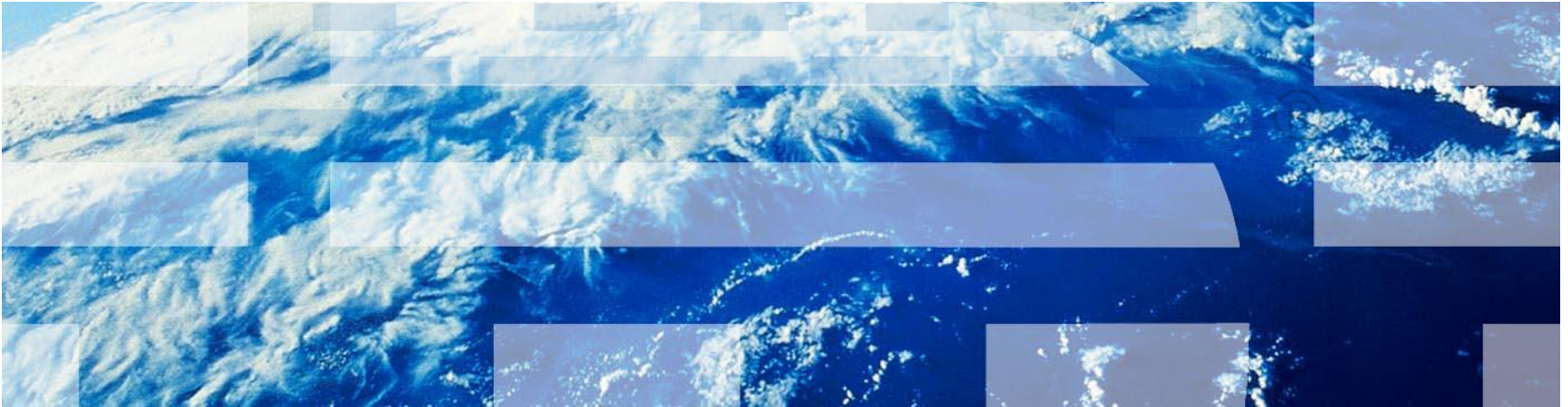


VP of Development, a Business Analyst or Project Owner, Agile Team Lead, Developer and Test Lead.

[bit.ly/jazzCLMdemo](http://bit.ly/jazzCLMdemo)



# Lab Overview





# Agenda

- Introduction to Collaborative Lifecycle Management

- **Lab Overview**

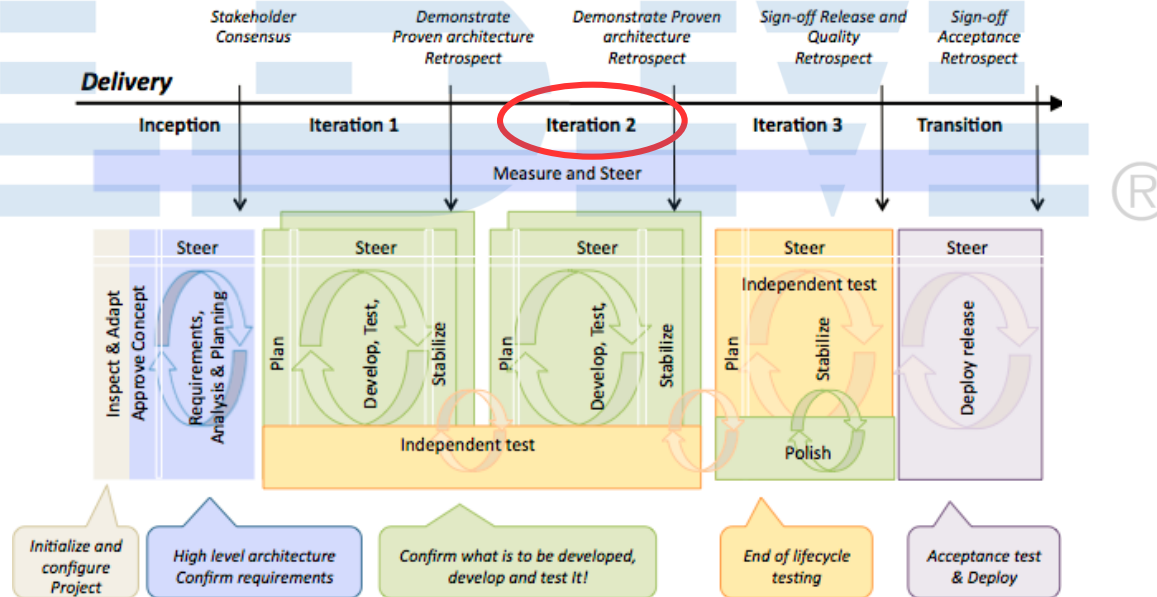
- Module 1: Aligning plans
- Module 2: Responding to a change request
- Module 3: Planning the sprint
- Module 4: Completing a story
- Module 5: Testing and fixing a story
- Module 6: Stabilizing the sprint
- Session Summary



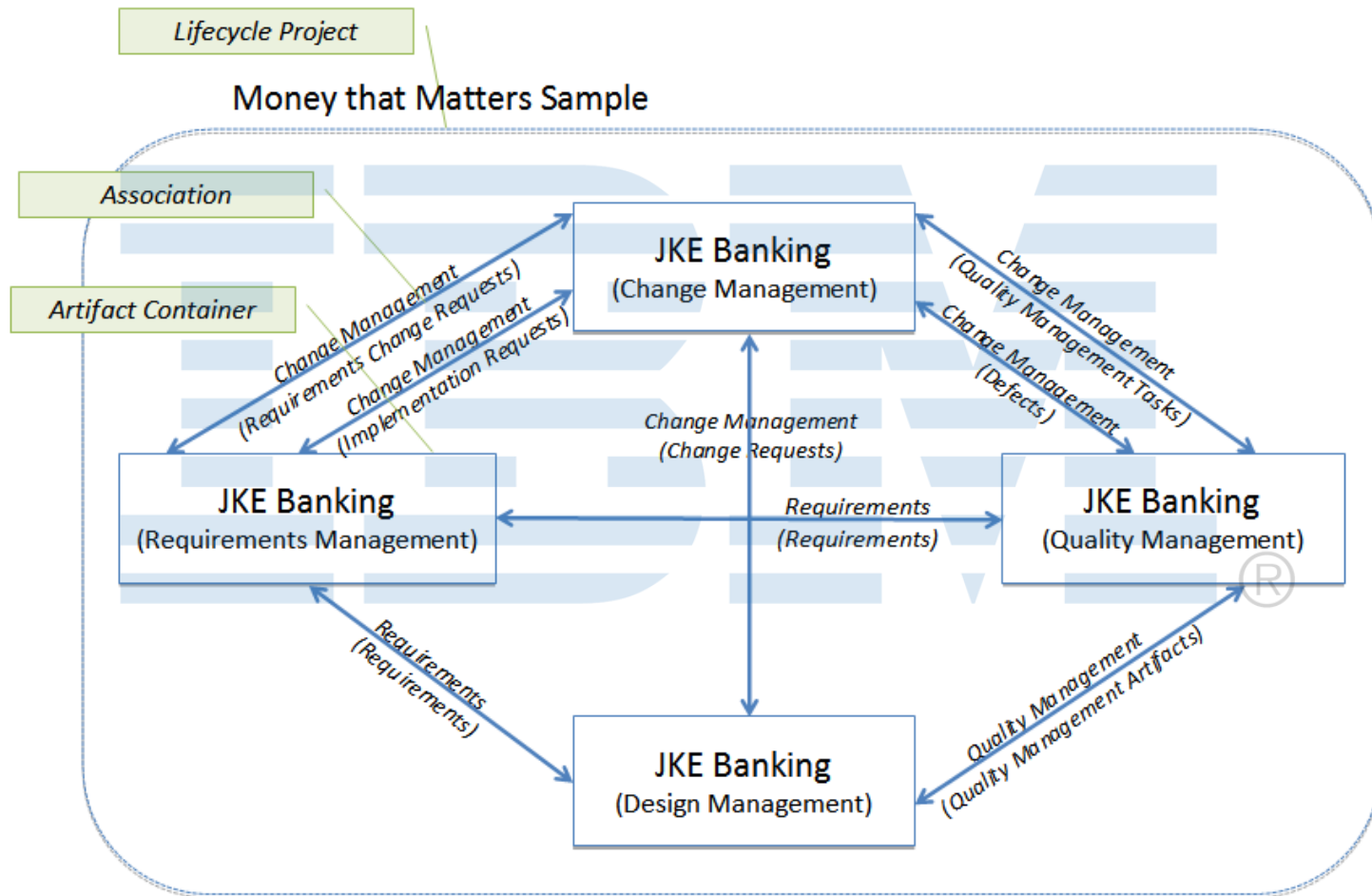
# Scenario for Proof of Technology (PoT) Labs

- This workshop is based on the **Money that Matters lifecycle scenario** which uses a fictitious banking company called JKE and a realistic software development project whose goal is to deliver **Release 1 of the JKE Banking application**.
- The scenario centers on a single iteration (sprint) by the team responsible for delivering an initiative called "Business Recovery Matters".

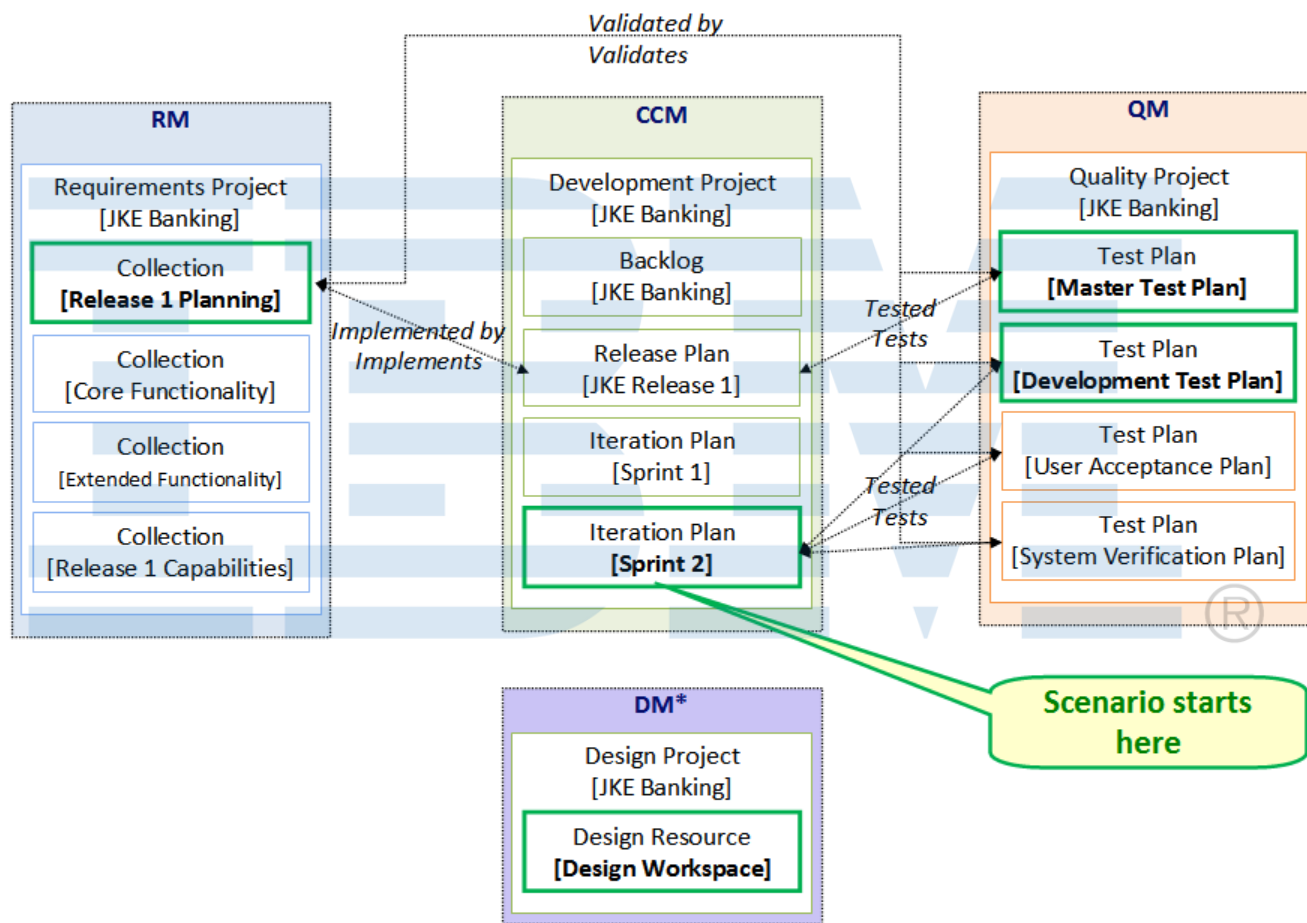
**Release Lifecycle Overview**  
Hybrid agile practices applied at scale



# JKE Lifecycle Project



# All Plans are linked



\* DM is included in the CCM development plans. DM elements are connected to other elements at a lower level (requirements, test case, etc.).

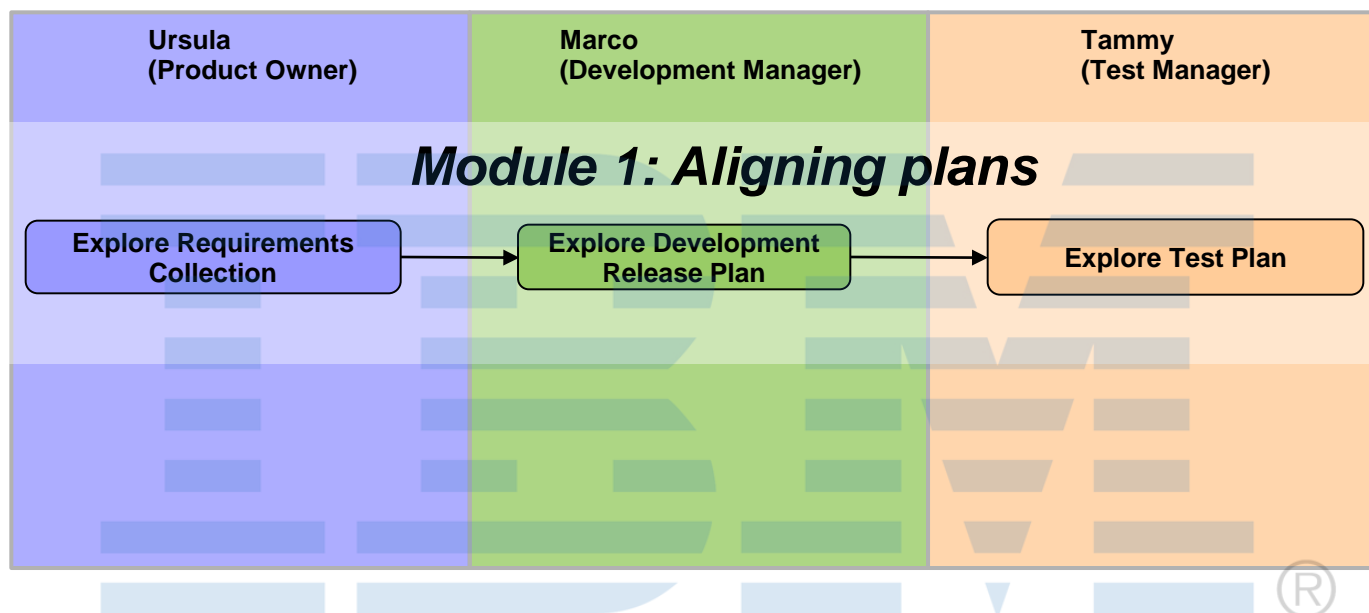
## Scenario for Workshop

- You will play the role of various members on the JKE Banking project as they work through the definition, prioritization, implementation, testing and fixing of a new requirement:
  - **Ursula**: The product owner for the whole JKE Banking solution
  - **Bob**: The product owner for the Business Recovery Matters solution
  - **Al**: The architect
  - **Marco**: The development manager and Scrum Master
  - **Deb**: A developer
  - **Rebecca**: The release engineer
  - **Tammy**: The test manager
  - **Tanuj**: A tester
  
- The project follows the Scrum methodology.
  - The CLM Solution is process agnostic. Scrum/agile is just one way to implement the CLM solution. Traditional waterfall, iterative or hybrid approaches may be used as well.
  
- You will be using IBM Rational Requirements Composer, IBM Rational Team Concert, IBM Rational Quality Manager, and IBM Rational Software Architect Design Manager as the project's collaborative lifecycle management and software development environment.

## Quick Scrum Overview

- “Scrum” is an agile process for software development. *(Often spelled “SCRUM”, although not an acronym.)*
  
- Scrum roles:
  - **Scrum Master:** Individual who maintains the processes (essentially a project manager). Responsible for removing impediments to team progress.
  - **Product Owner:** Represents the stakeholders
  - **Team:** Cross-functional group of people who do the work
  
- Scrum concepts:
  - **Story:** A brief description of a user need. Each story has a relative priority and complexity (story points).
  - **Product backlog:** A prioritized set of high-level requirements of work, usually described in stories.
  - **Sprint:** A two-to-four week period in which the team creates a potentially shipping product. A Scrum project consists of several sprints.
  - **Sprint planning meeting:** A meeting to determine which backlog items go into a sprint.
  - **Scrum:** A short daily meeting where each team member shares what they accomplished yesterday, what they will work on today and what, if anything is blocking their progress.

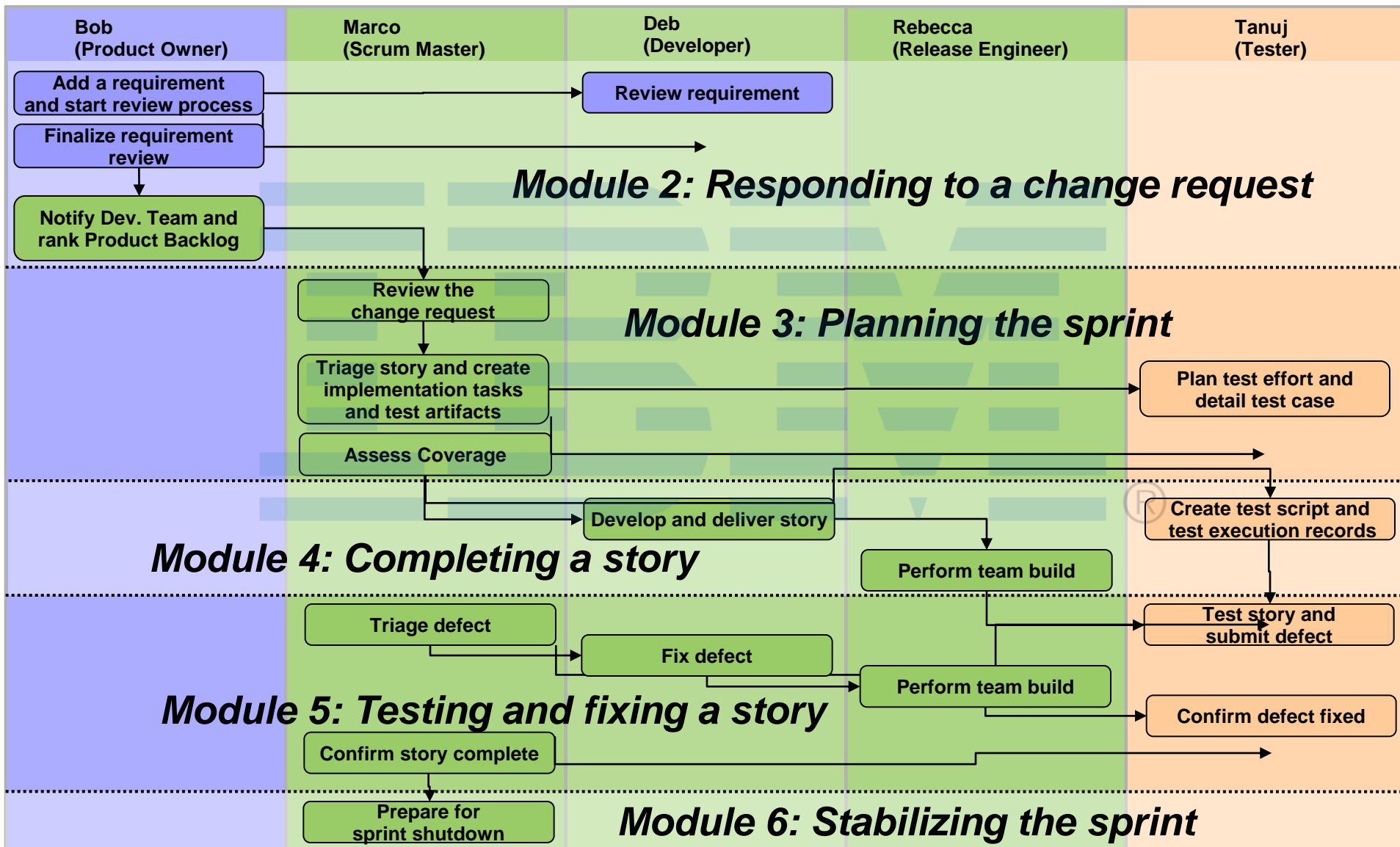
## Lab Sequence of Events



- The sample application comes with a release plan already in place.
- In the first module, you will explore the existing collection, development release plan and test plan as well as the links between them.



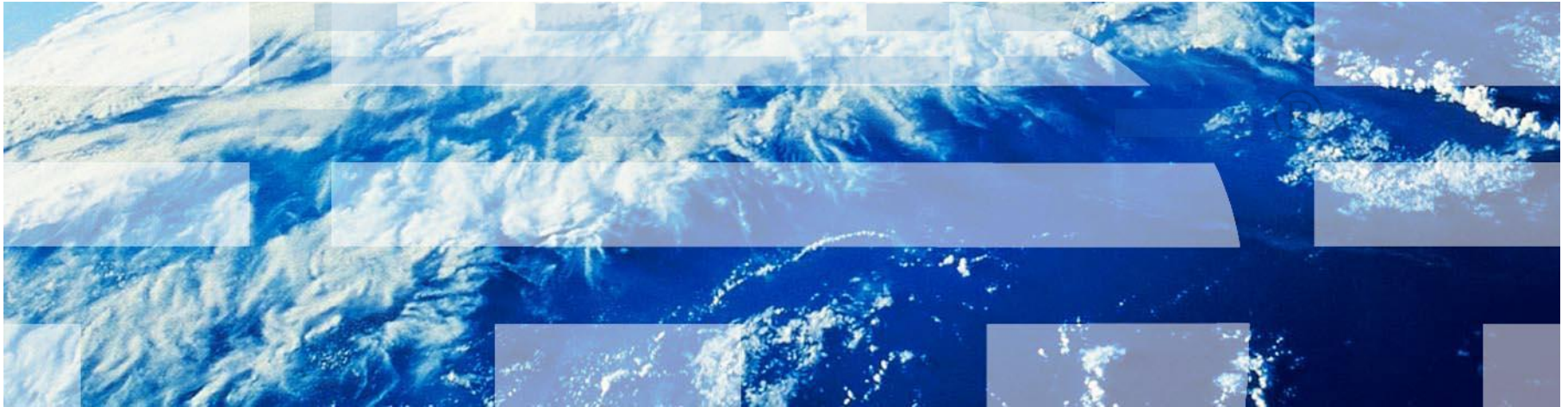
# Lab Sequence of Events



# Questions

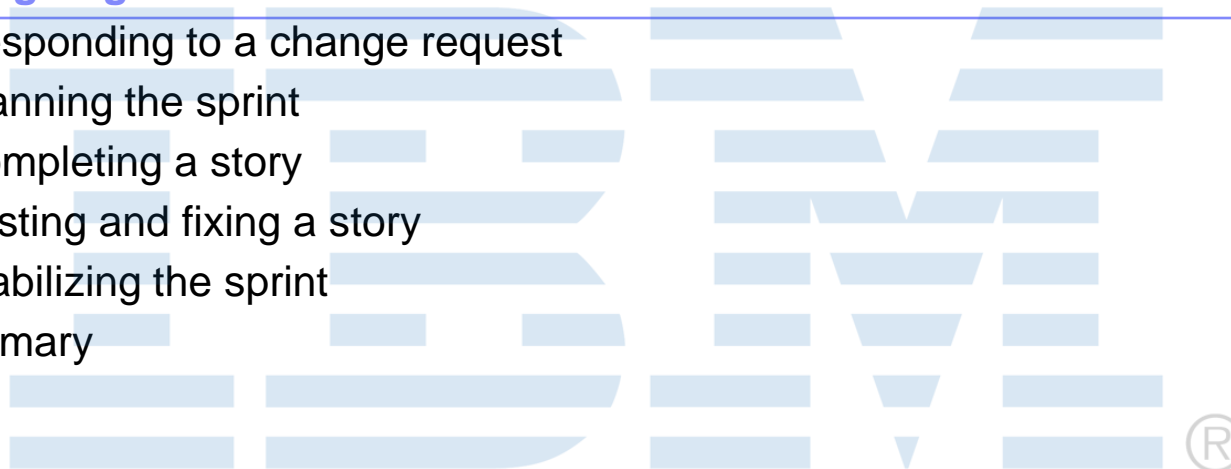


# Module 1 – Aligning Plans



# Agenda

- Introduction to Collaborative Lifecycle Management
- Lab Overview
- **Module 1: Aligning Plans**
- Module 2: Responding to a change request
- Module 3: Planning the sprint
- Module 4: Completing a story
- Module 5: Testing and fixing a story
- Module 6: Stabilizing the sprint
- Session Summary

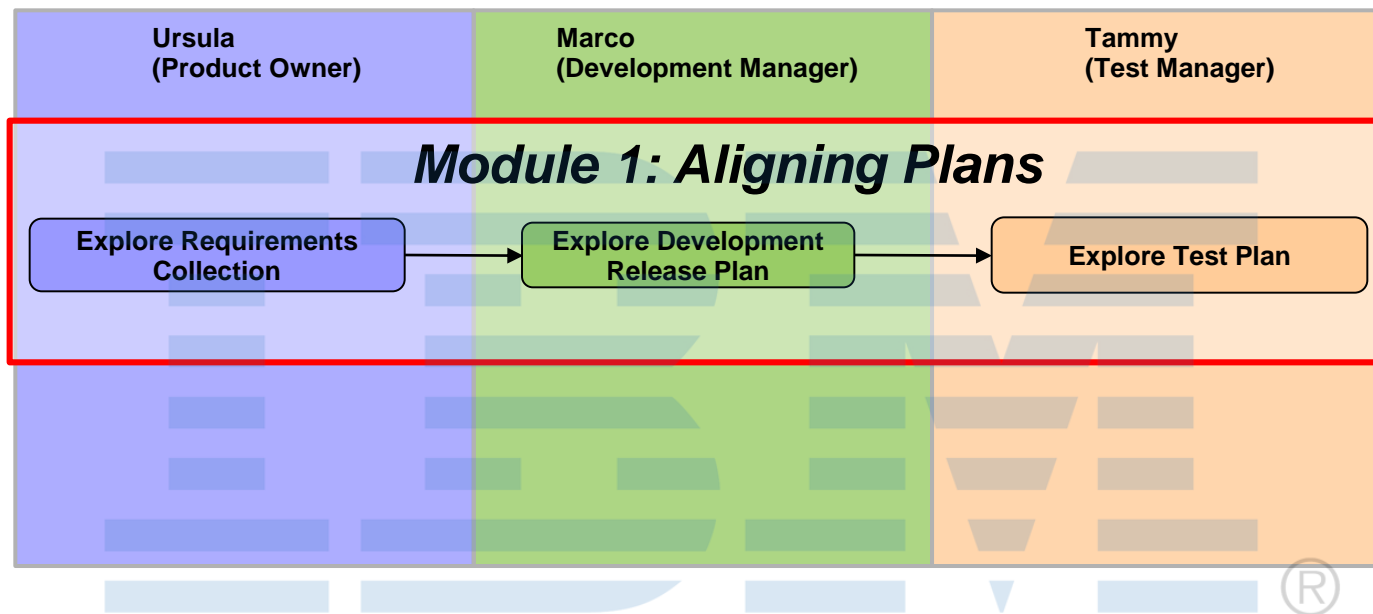


## Objective

- Explore the IBM Rational Requirements Composer (RRC), IBM Rational Team Concert (RTC), and IBM Rational Quality Manager (RQM) Web interfaces.
- Explore how IBM Rational Requirements Composer can be used to capture and manage requirement artifacts and collections.
- Explore the IBM Rational Team Concert can manage release development plans.
- Explore how IBM Rational Quality Manager can manage test plans.
- Explore how all those three products seamlessly integrate to align the plans together.



## Lab Sequence of Events



- The sample application comes with a release plan already in place.
- In the first module, you will explore the existing collection, development release plan and test plan as well as the links between them.

## Lab 1 Scenario

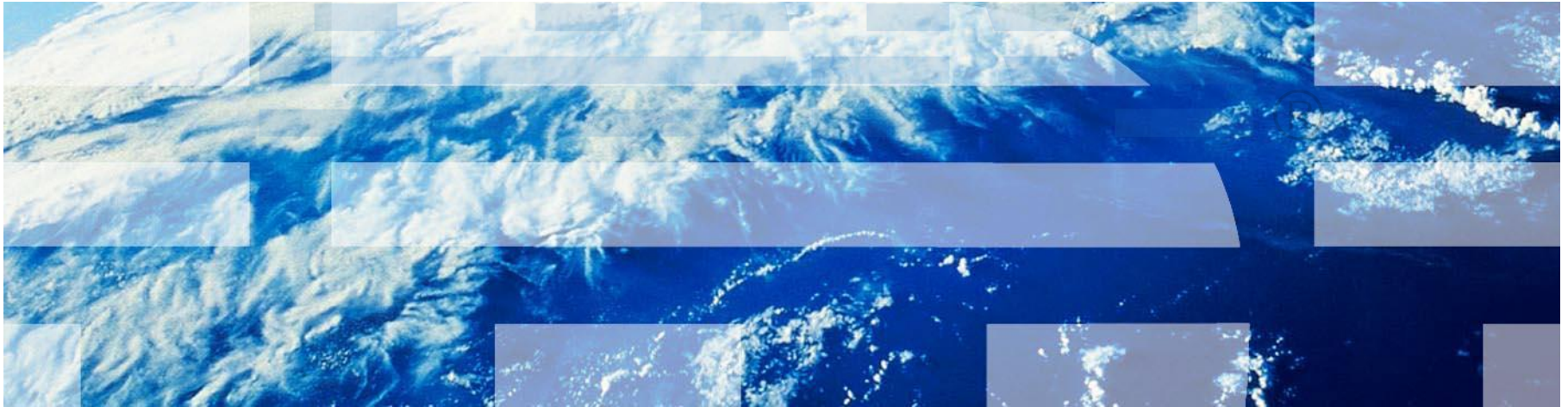
- You will play the role of Ursula, the product owner, in this lab.
- You will explore how release plan have been created and are aligned from 3 different perspectives.
- As Ursula, the product owner, you will use the IBM Rational Team Concert Web interface to explore the stories and the release plan and see how they are linked to requirement collections and test plans.
- As Ursula, you will also use the IBM Rational Quality Manager Web interface to explore the test plans and see how they are linked to requirement collections and release plans.
- As Ursula, you will finally use the IBM Rational Requirement Composer Web interface to explore the requirement artifacts and collections and see how they are linked to release and test plans.



## Lab 1 Concepts Learned

- IBM Rational Requirements Composer (RRC) can be used to define and describe high level requirements and to organize them in collections.
- The IBM Rational Team Concert (RTC) can be used to plan a release, and organize the development in multiple iterations or sprints.
- IBM Rational Quality Manager (RQM) can be used to capture test plans and test cases.
- IBM Rational Requirements Composer, IBM Rational Team Concert, and IBM Rational Quality Manager:
  - Work together to help all team members stay in sync with the development plan.
  - Give users of each tool visibility into data from the other tools, and link items from each of the tools together.
  - In all three products, dashboards contain interactive viewlets that provide each team member with the information they require. Projects, teams and individuals can have their own dashboards customized to their needs.

# Module 2 – Responding to a change request



# Agenda

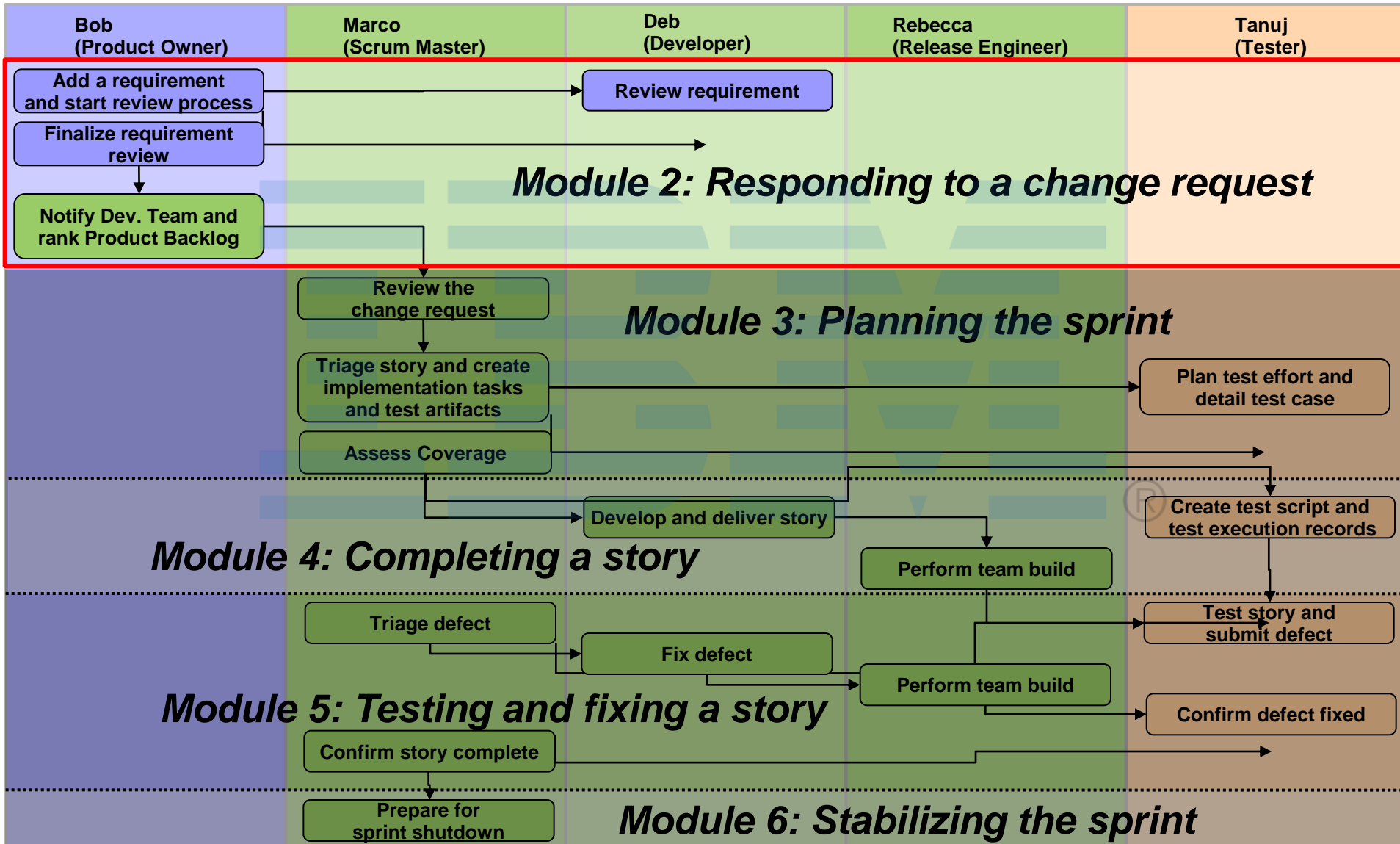
- Introduction to Collaborative Lifecycle Management
- Lab Overview
- Module 1: Aligning plans
- **Module 2: Responding to a change request**
- Module 3: Planning the sprint
- Module 4: Completing a story
- Module 5: Testing and fixing a story
- Module 6: Stabilizing the sprint
- Session Summary



## Objectives

- Further explore the IBM Rational Requirements Composer (RRC) Web interface.
- Explore the integration between IBM Rational Requirements Composer and IBM Rational Team Concert (RTC) that enables creating and linking to new or existing RTC work items from RRC requirements that help ensure and track story to development work item coverage.
- Explore how IBM Rational Requirements Composer directly supports requirements review and approval process.
- Explore the Rational Team Concert Web interface from an product owner's point of view.
- Explore how Rational Team Concert can manage SCRUM story and product work item backlogs.
- Explore how to use Rational Software Architect Design Manager to collaborate around the design of the product you are developing.

# Lab Sequence of Events



## Lab 2 Scenario

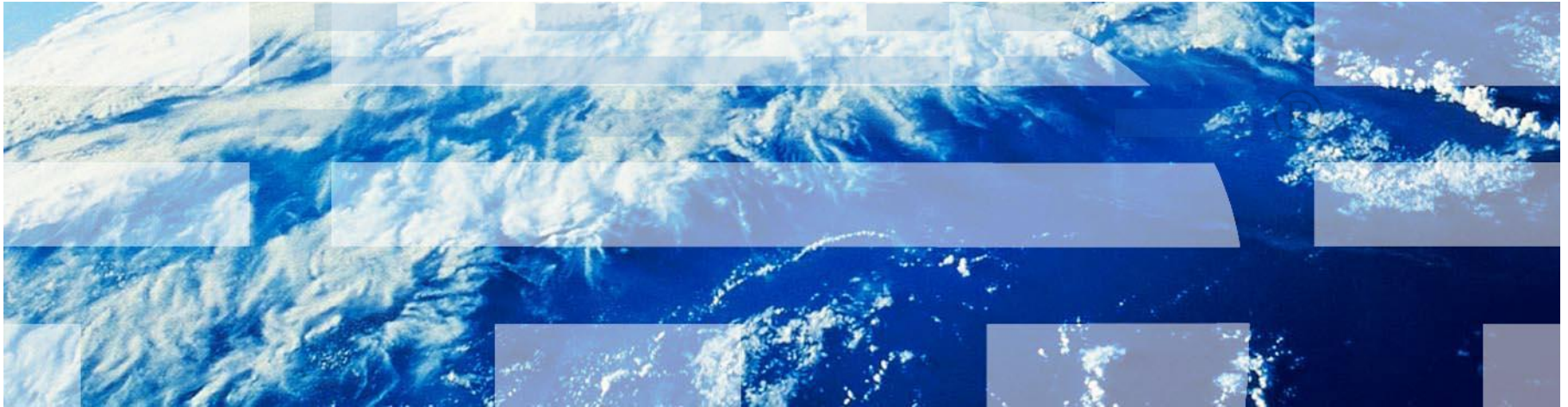
- You will play two different roles in this lab!
- As Bob, the product owner, you will take into account a new stakeholder change request and make sure appropriate requirement artifacts in IBM Rational Requirements Composer for this new story and then start the review process.
- As Curtis, a customer, you will review the new requirement, and approve it using IBM Rational Requirements Composer Web interface.
- As Bob, the product owner again, you will prioritize the current sprint's story work item backlog in IBM Rational Team Concert to ensure that the new story is considered for the coming sprint, and notify the development team about that change request so they can analyze the impact. You will also review the design of the application models against the corporate policies.

## Lab 2 Concepts Learned

- IBM Rational Requirements Composer (RRC) can be used to define and describe various requirement artifacts utilizing a variety of techniques and resource types including rich text (note cards), user interface sketches, storyboards, and flows.
- All RRC resource types can have attributes, be tagged, and linked to each other to create a broader “web of linked context” to facilitate better definition and understanding of what is required from development and test .
- IBM Rational Requirements Composer (RRC) directly supports requirements review and approval process.
- IBM Rational Team Concert provides direct support for Scrum. (Note: While this session uses Scrum, IBM Rational Team Concert provides excellent support for a variety of development methodologies.)
- Requirements artifacts defined in IBM Rational Requirements Composer can be linked to new or existing development stories and work items in IBM Rational Team Concert (RTC).
- IBM Rational Software Architect Design Manager (RSA/DM) supports collaboration between team members around the design of the product.



# Module 3 – Planning the sprint





# Agenda

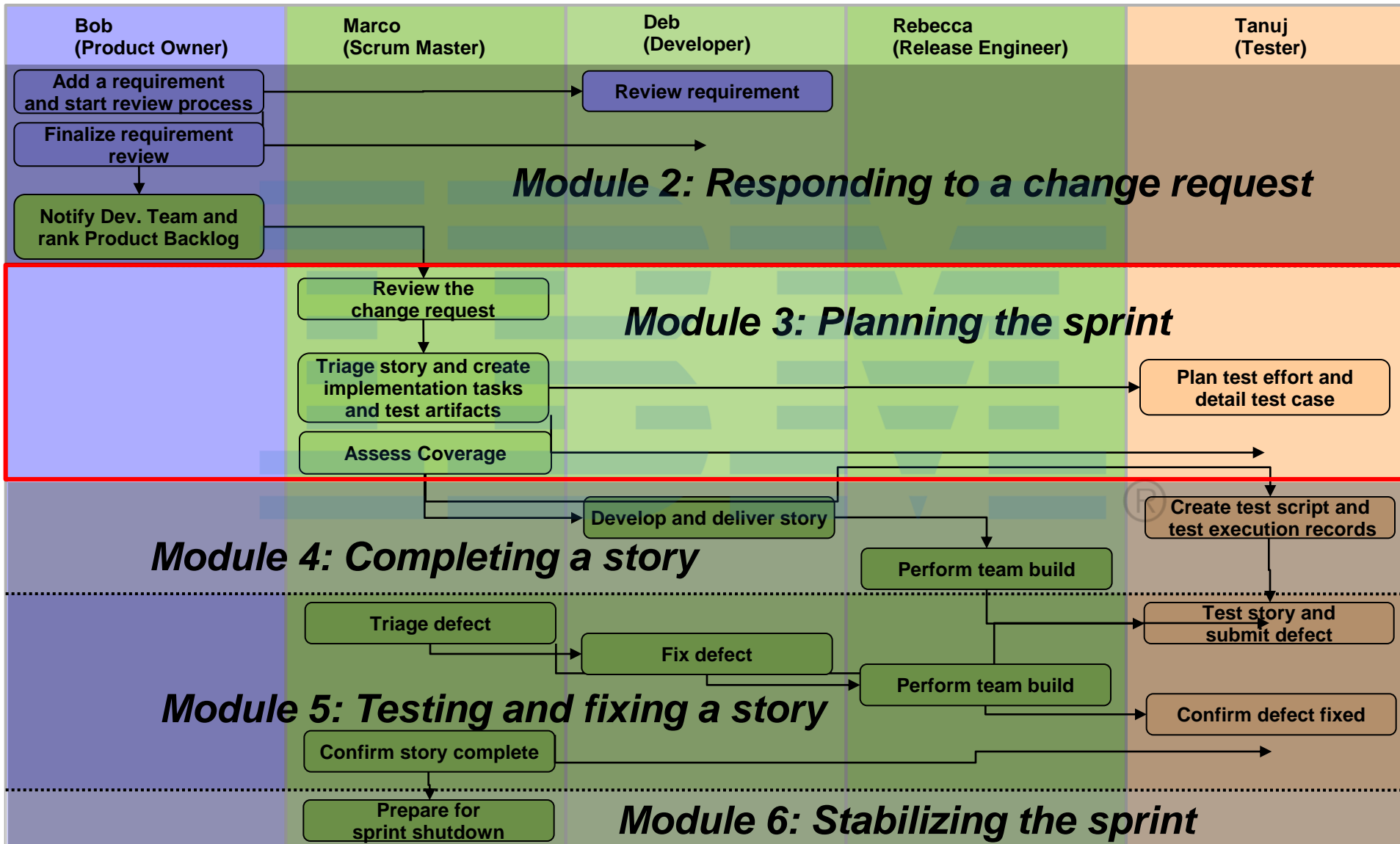
- Introduction to Collaborative Lifecycle Management
- Lab Overview
- Module 1: Aligning plans
- Module 2: Responding to a change request
- **Module 3: Planning the sprint**
- Module 4: Completing a story
- Module 5: Testing and fixing a story
- Module 6: Stabilizing the sprint
- Session Summary



## Objectives

- Explore how IBM Rational Software Architect Design Manager helps you design just enough to prepare for sprint meetings.
- Further explore the IBM Rational Team Concert (RTC) Web interface as well as dashboards and viewlets.
- Explore how IBM Rational Team Concert enables agile teams to manage sprint/iteration plans and manage tasks.
- Explore how IBM Rational Team Concert can be used to facilitate a sprint planning meeting.
- Create a new test case directly from IBM Rational Team Concert associated with a story.
- Explore how IBM Rational Quality Manager can be used to detail and organize test cases and plan test effort.
- Explore how to assess plan coverage from with IBM Rational Team Concert.

# Lab Sequence of Events



## Lab 3 Scenario

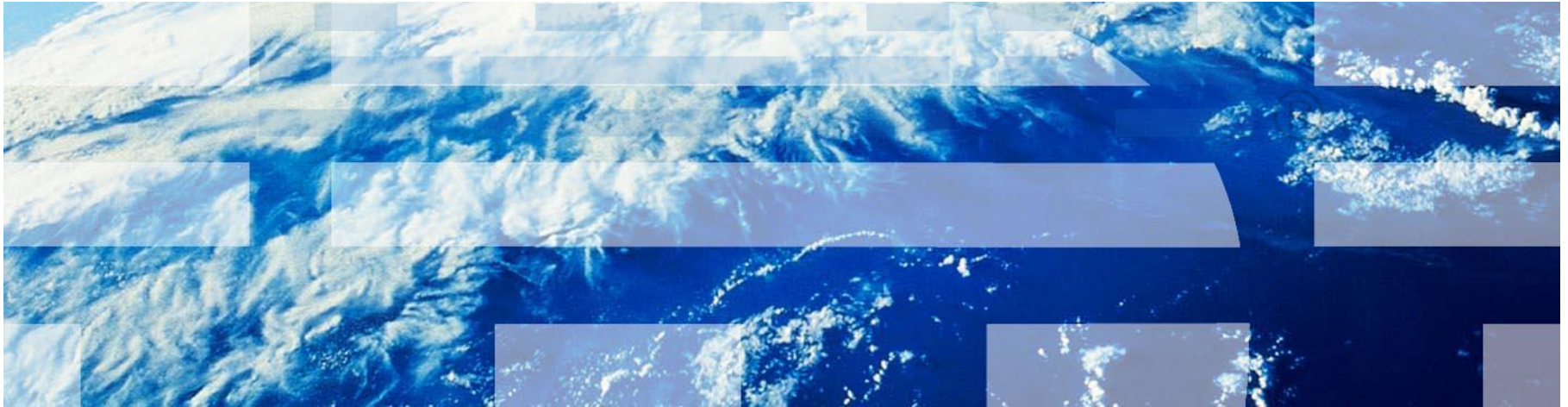
- You will play three different roles in this lab!
- As AI, the architect, you will use the IBM Rational Team Concert Web interface and the IBM Rational Software Architect Design Manager to plan the development effort.
- As Marco, the Scrum Master and development lead, you will use the IBM Rational Team Concert Web interface to review the Product Backlog, triage the new story to the current sprint/iteration, and create the implementation tasks and associated test case.
- As Tanuj, a tester, you will use IBM Rational Quality Manager to review and detail the newly created test case and plan the test effort.
- As Marco again, you will assess plan coverage using IBM Rational Team Concert.

## Lab 3 Concepts Learned

- IBM Rational Team Concert and IBM Rational Software Architect Design Manager makes it easy to manage a project's development plan.
- Stories and related implementation tasks defined in IBM Rational Team Concert (RTC) can be used to create and link to new or existing test case in IBM Rational Quality Manager (RQM).
- IBM Rational Quality Manager helps create and manage test plan and test cases linked with development stories.
- IBM Rational Requirements Composer, IBM Rational Team Concert and IBM Rational Quality Manager provides full visibility of CLM relationships across requirements, development and test artifacts.



# Module 4 – Completing a story



# Agenda

- Introduction to Collaborative Lifecycle Management
- Lab Overview
- Module 1: Aligning plans
- Module 2: Responding to a change request
- Module 3: Planning the sprint
- **Module 4: Completing a story**
- Module 5: Testing and fixing a story
- Module 6: Stabilizing the sprint
- Session Summary



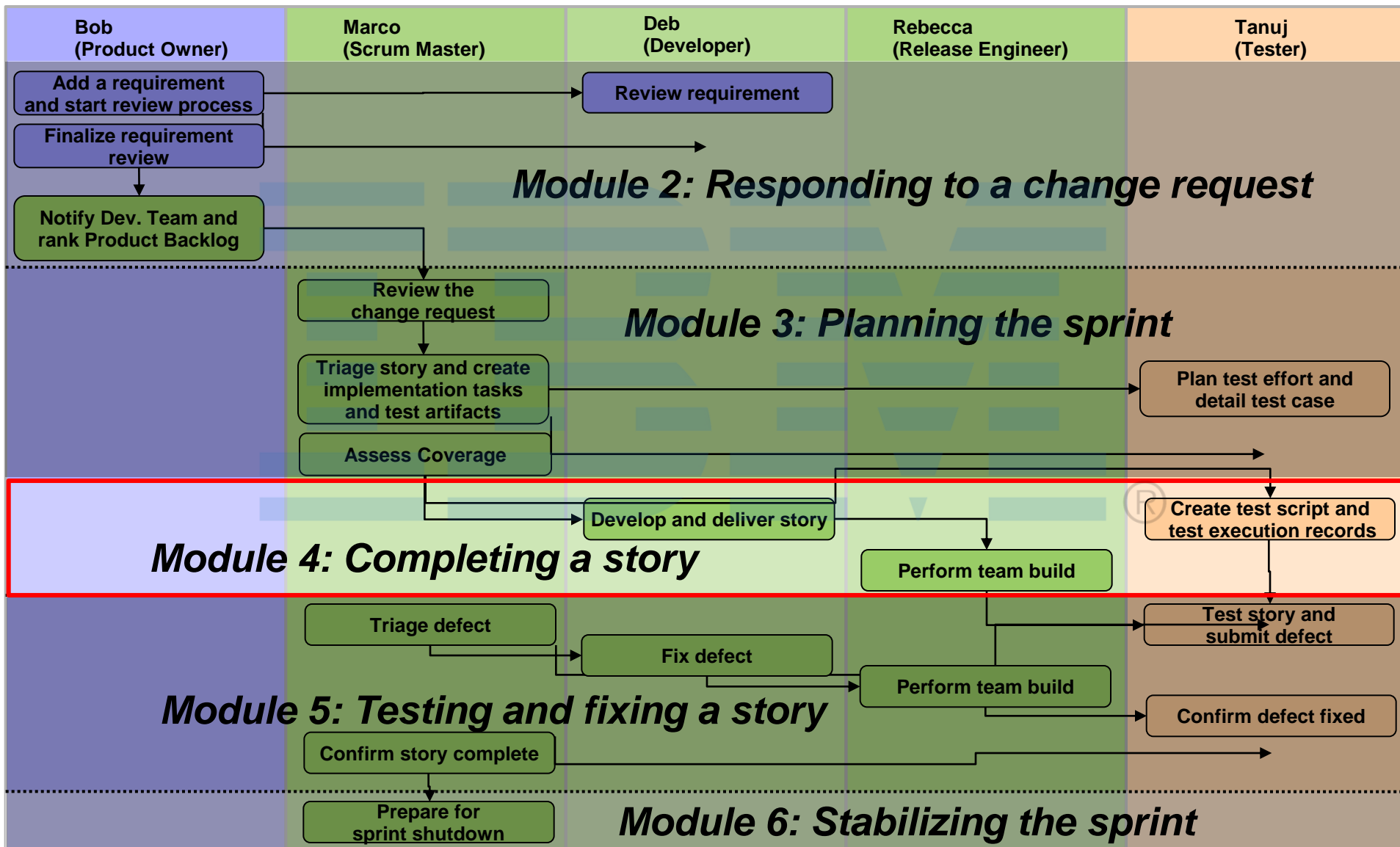


## Objectives

- Explore how IBM Rational Quality Manager can be used to create test implementations (scripts) and associate them with test cases.
- Explore the IBM Rational Team Concert Eclipse client from a developer's perspective.
- Understand how to use IBM Rational Team Concert to accept work, complete development tasks and deliver updated work to the team.
- Explore the team build features of IBM Rational Team Concert.
- Explore how all team members can monitor the status of team builds.



# Lab Sequence of Events



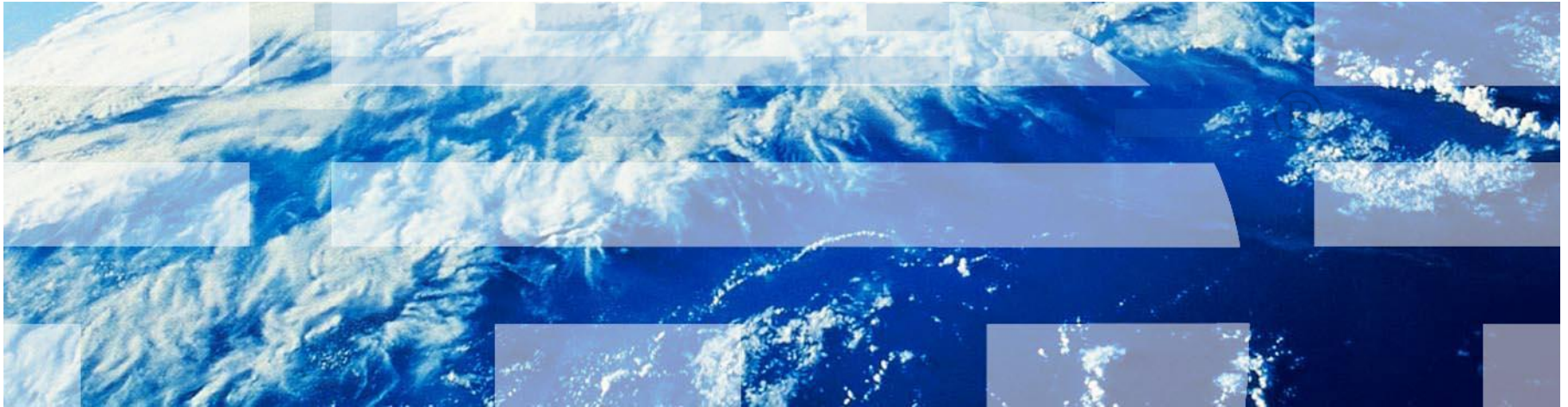
## Lab 4 Scenario

- You will play four different roles in this lab!
- As Tanuj, a tester, you will use IBM Rational Quality Manager to create and edit new test scripts and test execution records for the test case you added in the previous lab.
- As Al, the architect, you will review the design models and create a logical topology and a deployment model to help Deb understand the target application server environment and the assumptions he made while generating the initial implementation code from the service design model.
- As Deb, a developer, you will use the IBM Rational Team Concert Eclipse client to plan your work, complete your development tasks, execute an personal build, and deliver your updates to the team.
- As Rebecca, the release engineer, you will use the IBM Rational Team Concert Eclipse client to execute an integration build on the team's build server.

## Lab 4 Concepts Learned

- IBM Rational Quality Manager provides a highly descriptive manual testing facility that provides the tester with the right level of detail required to execute the test correctly.
- IBM Rational Quality Manager test scripts implement test cases that can be linked to the drivers for those test cases (in this case, stories).
- Jazz Source Control provides private repository workspaces to track and back up your changes before you share them with the team using a stream for integration.
- A change set is the fundamental unit of change and collaboration in your team environment. Change sets are migrated between streams via two operations: accepting and delivering.
- Developers can run "personal builds" on the team's build server to ensure that the code they see in their workspace successfully builds using the team's build process. Ensuring compilation in the integrated development environment (IDE) isn't always enough.
- Team members can request a "team build" that will grab the latest code on the team's integration stream and build it on the team build server.
- Every team member has access to build data from team builds. This promotes communication and collaboration among the contributors – on local or remote sites.

# Module 5: Testing and fixing a story



# Agenda

- Introduction to Collaborative Lifecycle Management
- Lab Overview
- Module 1: Aligning plans
- Module 2: Responding to a change request
- Module 3: Planning the sprint
- Module 4: Completing a story
- **Module 5: Testing and fixing a story**
- Module 6: Stabilizing the sprint
- Session Summary



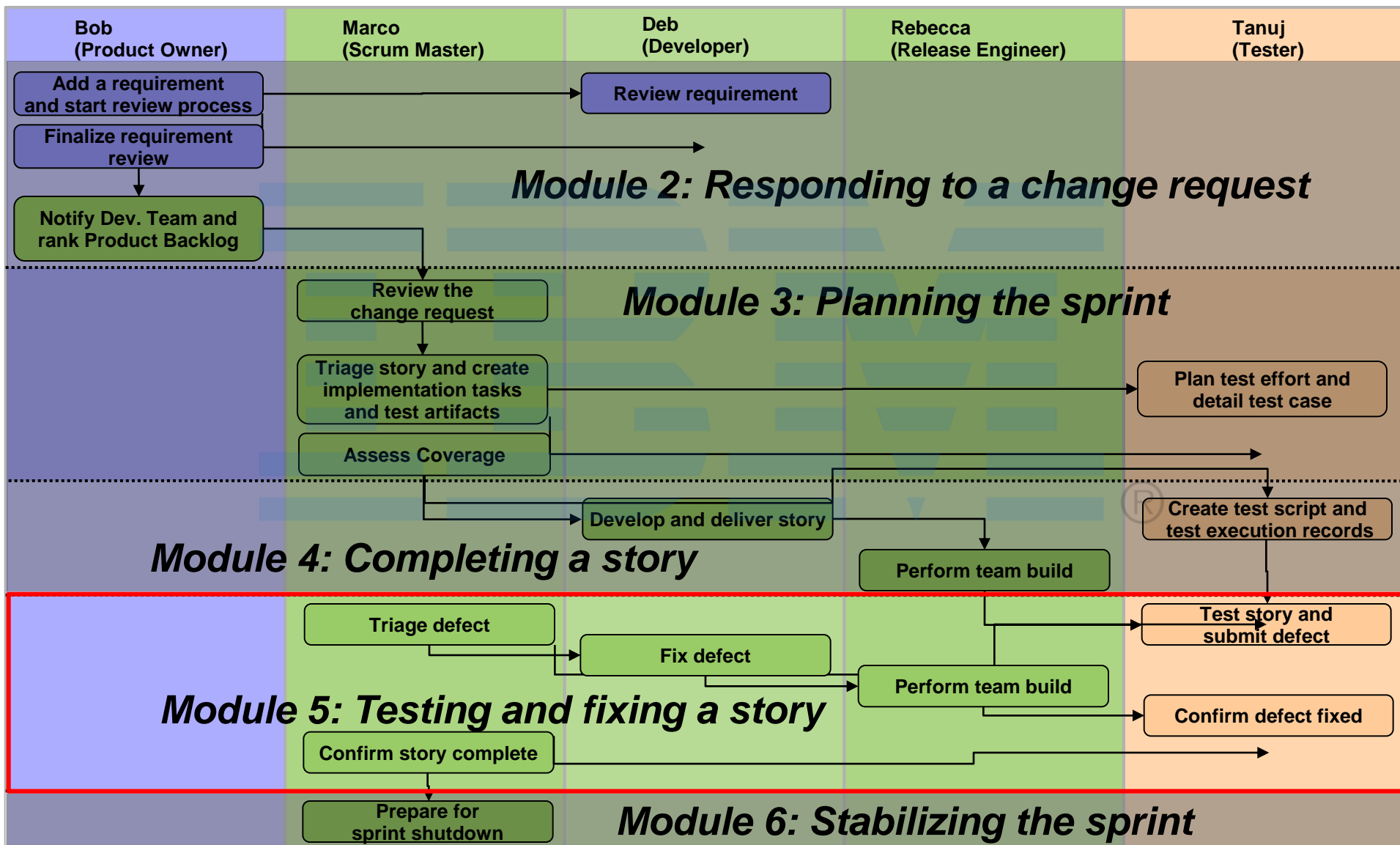
## Objectives

- Explore how IBM Rational Quality Manager can be used to execute manual tests, log test results and generate defects for the development team to fix.
- Explore how IBM Rational Team Concert can be used to triage defects and help determine who should be assigned new work.
- Further explore how a developer uses IBM Rational Team Concert to discover new work assigned to them, fix defects, deliver fixes to the team and work with team builds.
- Explore how a tester can use IBM Rational Quality Manager to monitor the team build status, determine what work is included in a new build and verify that defects have been fixed.





# Lab Sequence of Events



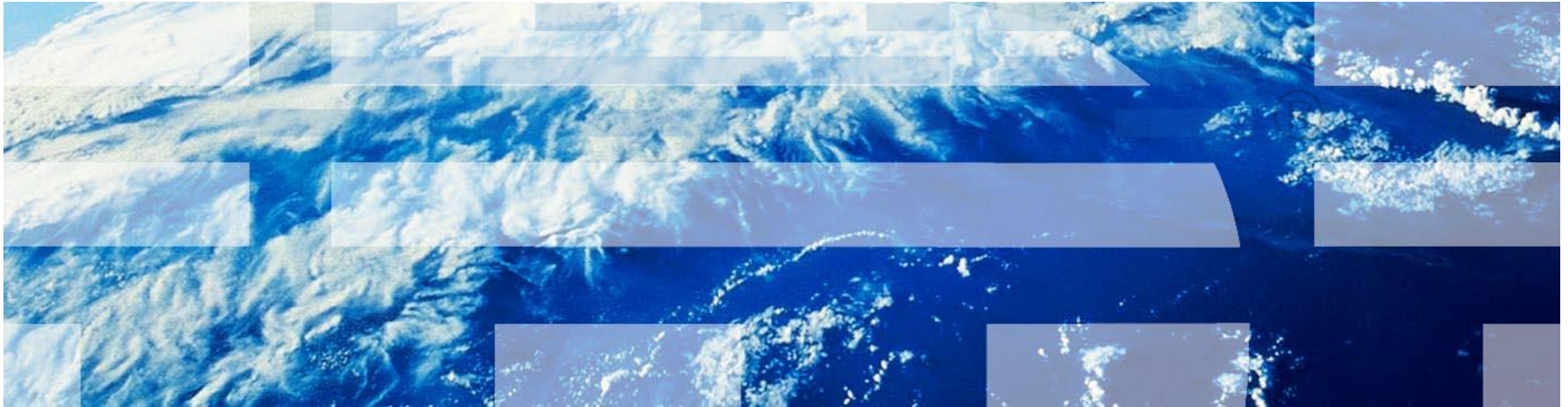
## Lab 5 Scenario

- You will play four different roles in this lab!
- As Tanuj, a tester, you will use IBM Rational Quality Manager to execute a manual test, denote that the test failed and generate a new defect related to the test result.
- As Marco, the development lead, you will use the IBM Rational Team Concert Web interface to analyze the new defect, determine who should fix the defect and assign the defect.
- As Deb, a developer, you will use the IBM Rational Team Concert Eclipse client to determine that there is a new defect assigned to you, analyze the test result related to that defect, resolve the defect and request a new team build.
- As Rebecca, the release engineer, you will use the IBM Rational Team Concert Eclipse client to execute an integration build on the team's build server.
- As Tanuj again, you will notice the new build result, determine what is fixed in that build, deploy the new build and verify that the defect you created is indeed fixed.
- As Marco again, you will use IBM Rational Team Concert to confirm that story is complete.

## Lab 5 Concepts Learned

- The IBM Rational Quality Manager manual test facility allows the test to document step-by-step test results as the test is being executed, including grabbing helpful screen snapshots that may be useful in debugging.
- Test results can document the build that was used for the test, to aid in debugging.
- Defects can be created from test results and routed to the development team. Defects created from test results are automatically linked back to the test result, to aid in debugging.
- Testers can denote which defects are preventing (blocking) test cases from succeeding ensuring they don't waste time executing the same test and generating duplicate defects. This also helps testers monitor the defects they've generated so they know when it's worthwhile to run the test again.
- IBM Rational Team Concert queries make it easy for team leads to find new work and determine the appropriate person for assignment.
- Change sets can be associated with work items to denote which artifacts were modified to implement a work item.

# Module 6: Stabilizing the sprint



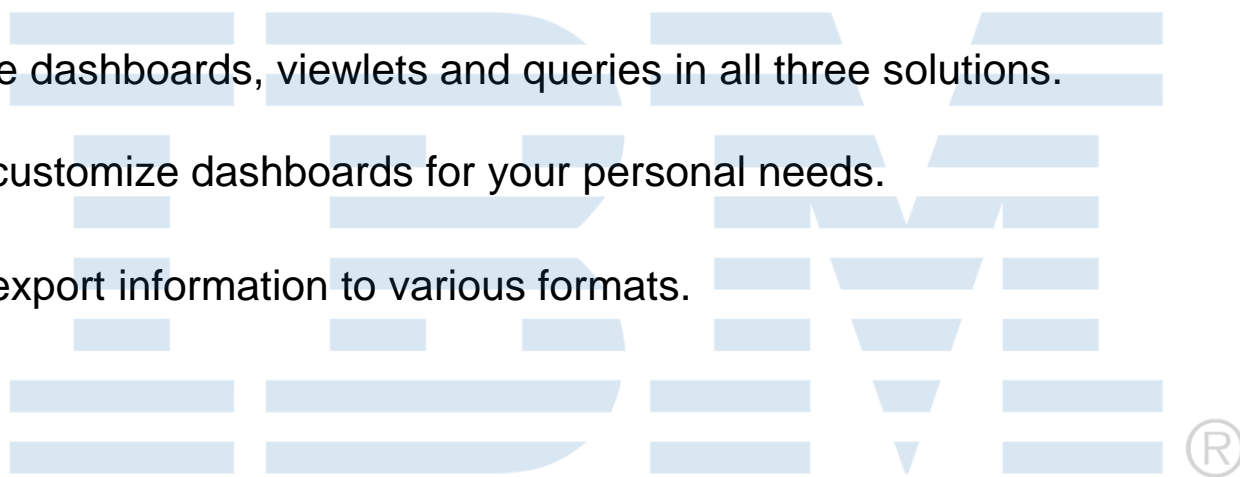
# Agenda

- Introduction to Collaborative Lifecycle Management
- Lab Overview
- Module 1: Planning a release
- Module 2: Responding to a change request
- Module 3: Planning the sprint
- Module 4: Completing a story
- Module 5: Testing and fixing a story
- **Module 6: Stabilizing the sprint**
- Session Summary

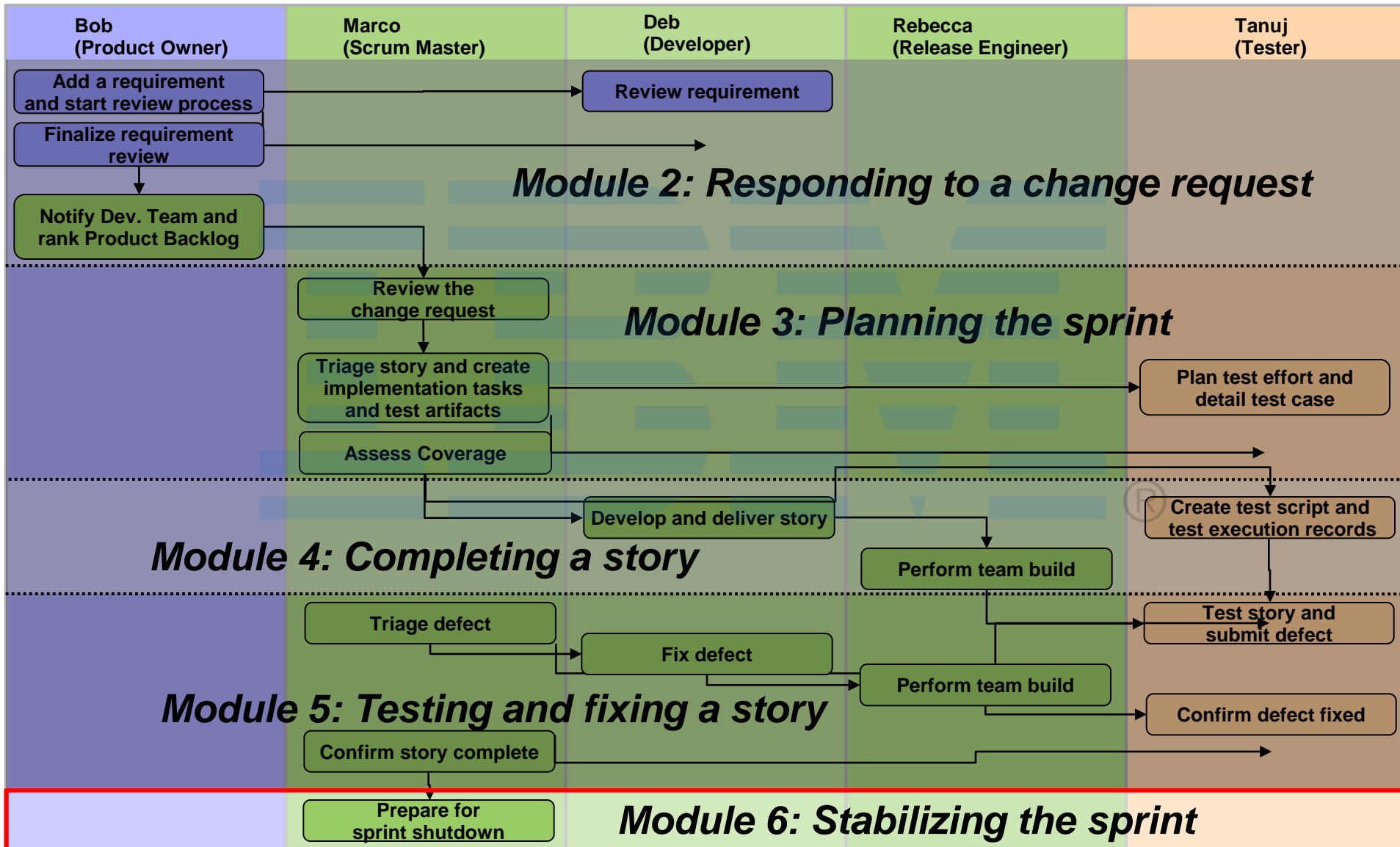


## Objectives

- Explore how IBM Rational Requirements Composer, IBM Rational Team Concert and IBM Rational Quality Manager can be used by various team roles to gain insight into the aspects of project status that are important to them.
- Further explore dashboards, viewlets and queries in all three solutions.
- Learn how to customize dashboards for your personal needs.
- Learn how to export information to various formats.



# Lab Sequence of Events





## Lab 6 Scenario

- You will play the role of various team members as they use IBM Rational Requirements Composer, IBM Rational Team Concert or IBM Rational Quality Manager to gain access to the data they need to monitor project status.

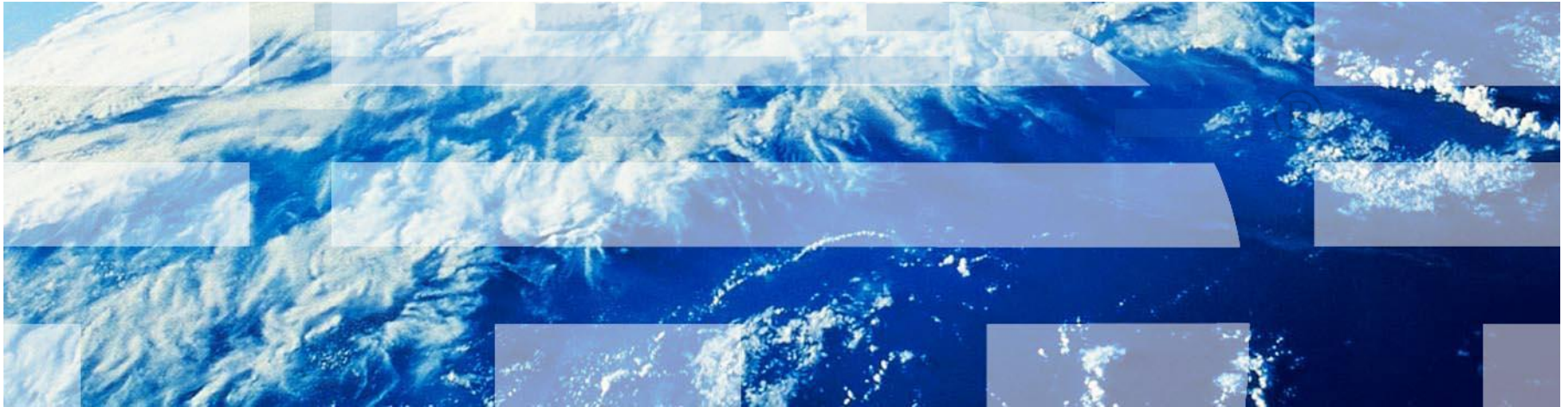


## Lab 6 Concepts Learned

- IBM Rational Requirements Composer, IBM Rational Team Concert and IBM Rational Quality Manager provide excellent reporting facilities within and across the solutions.



# Session Summary



# Agenda

- Introduction to Collaborative Lifecycle Management
- Lab Overview
- Module 1: Aligning plans
- Module 2: Responding to a change request
- Module 3: Planning the sprint
- Module 4: Completing a story
- Module 5: Testing and fixing a story
- Module 6: Stabilizing the sprint
- **Session Summary**



## Session summary

- We have described current collaboration challenges with distributed teams
- We have explored how Rational Requirements Composer, Rational Team Concert, Rational Quality Manager, and Rational Software Architect Design Manager can:
  - Enable teams to **collaborate in real time in the context** of the work they are doing, especially in globally diverse environments
  - Enable projects to be managed more effectively by providing visibility into **accurate project information** drawn directly from actual work
  - Automate traceability and auditability by **managing artifacts and their interrelationships** across the life cycle, empowering teams to deliver more value
- We have provided a hands-on experience using these products to automate the software delivery process

## Next Steps

- Engage your local Rational team
- Provide a customized demo for your team
- Conduct a targeted proof of concept
- Register on jazz.net and explore learning tutorials and videos
  - <http://jazz.net>
  - <http://www.ibm.com/developerworks/>



## Additional resources

- Learn more about and download free trials of Rational Requirements Composer, Rational Team Concert and Rational Quality Manager

<https://jazz.net/downloads>

- Explore tutorials, demos and other developer learning resources

<http://jazz.net>

- Participate in the open commercial development of Jazz by joining the community

<http://jazz.net>

- Learn more about the Jazz technology and the future IBM Rational product roadmap

<http://ibm.com/rational/jazz/roadmap>



# Questions







- Learn more at:

- [IBM Rational software](#)
- [IBM Rational Software Delivery Platform](#)
- [Process and portfolio management](#)
- [Change and release management](#)
- [Quality management](#)
- [Architecture management](#)

- [Rational trial downloads](#)
- [Leading Innovation Web site](#)
- [developerWorks Rational](#)
- [IBM Rational TV](#)
- [IBM Business Partners](#)
- [IBM Rational Case Studies](#)



© Copyright IBM Corporation 2008. 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, 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.