Fariz Saracevic, Rational Solution for CLM Leader
IBM Software, Rational
June 22nd, 2011

IBM Rational solution for Collaborative Lifecycle Management

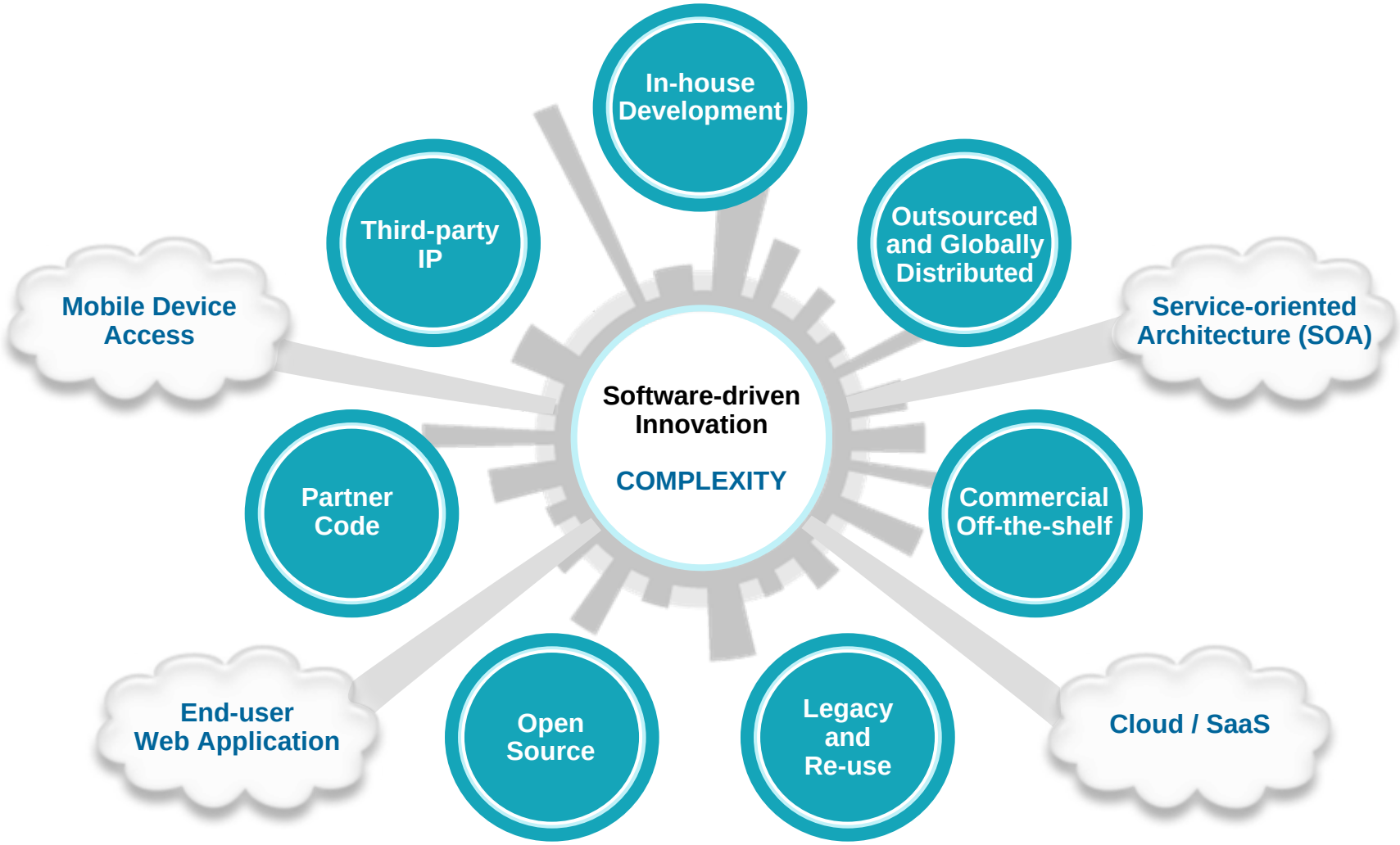
1. Overview
2. IBM Rational Collaboration Lifecycle Management
3. Delivering the 5 ALM imperatives
4. Proven results

Software drives today's innovation for a smarter planet

Transforming the way we live, work, and play

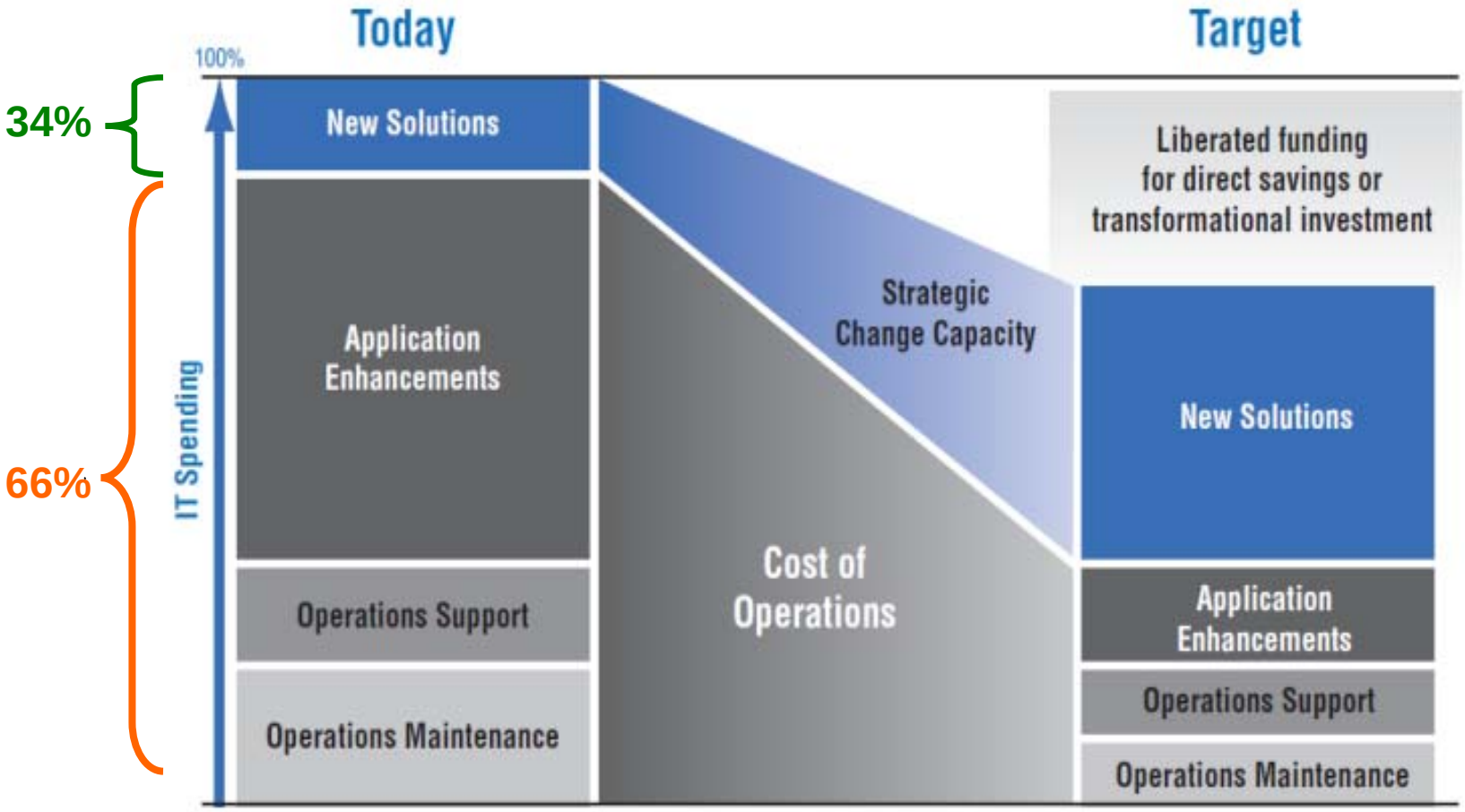
The defining challenge: Managing “systems of systems”

From back-end software to customer facing portals, systems of systems drive your relationships with customers, suppliers and business partners



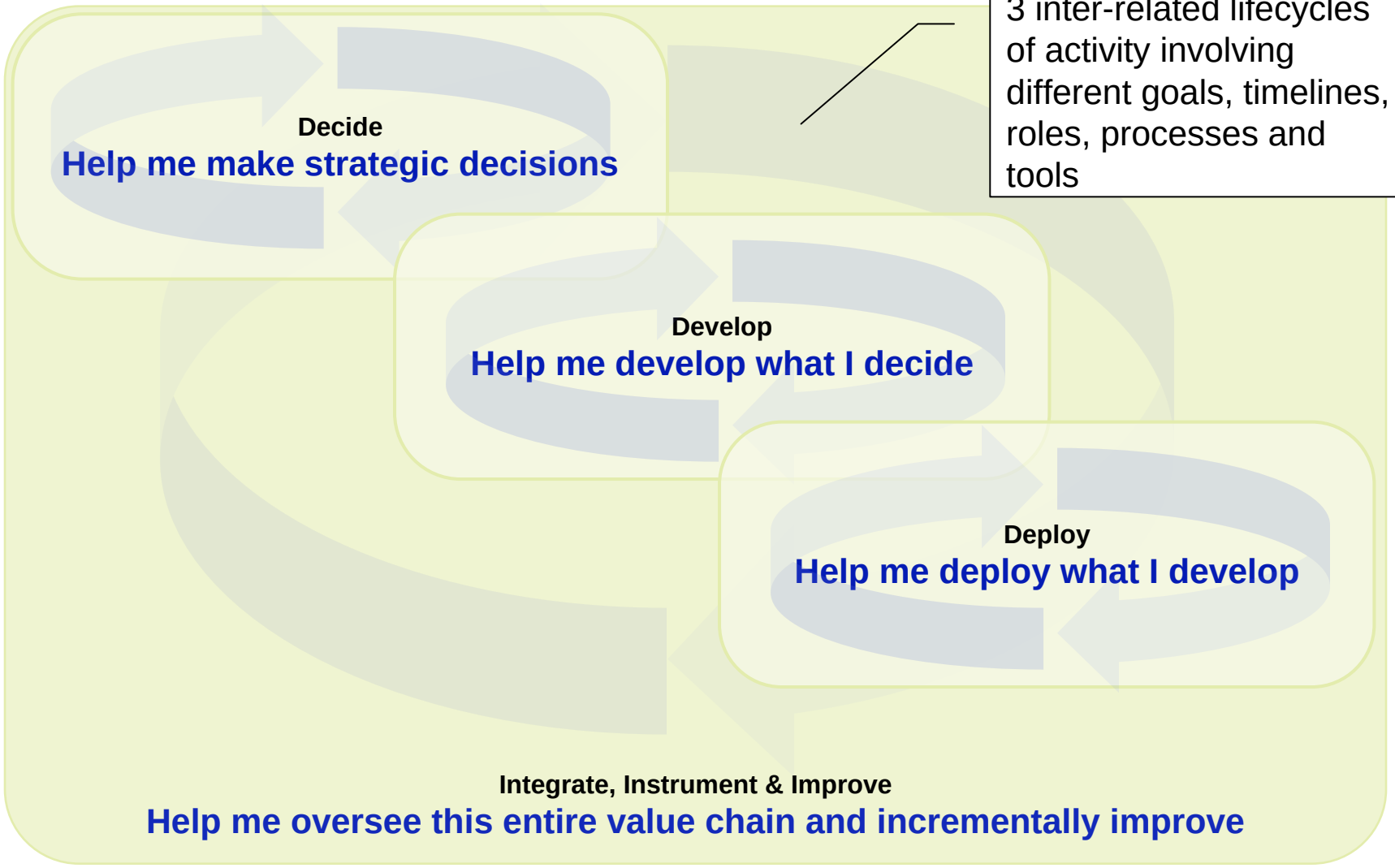
Business and IT Agility: Balancing Resources to Support Business Innovation

Balance IT investments to focus on new solutions.

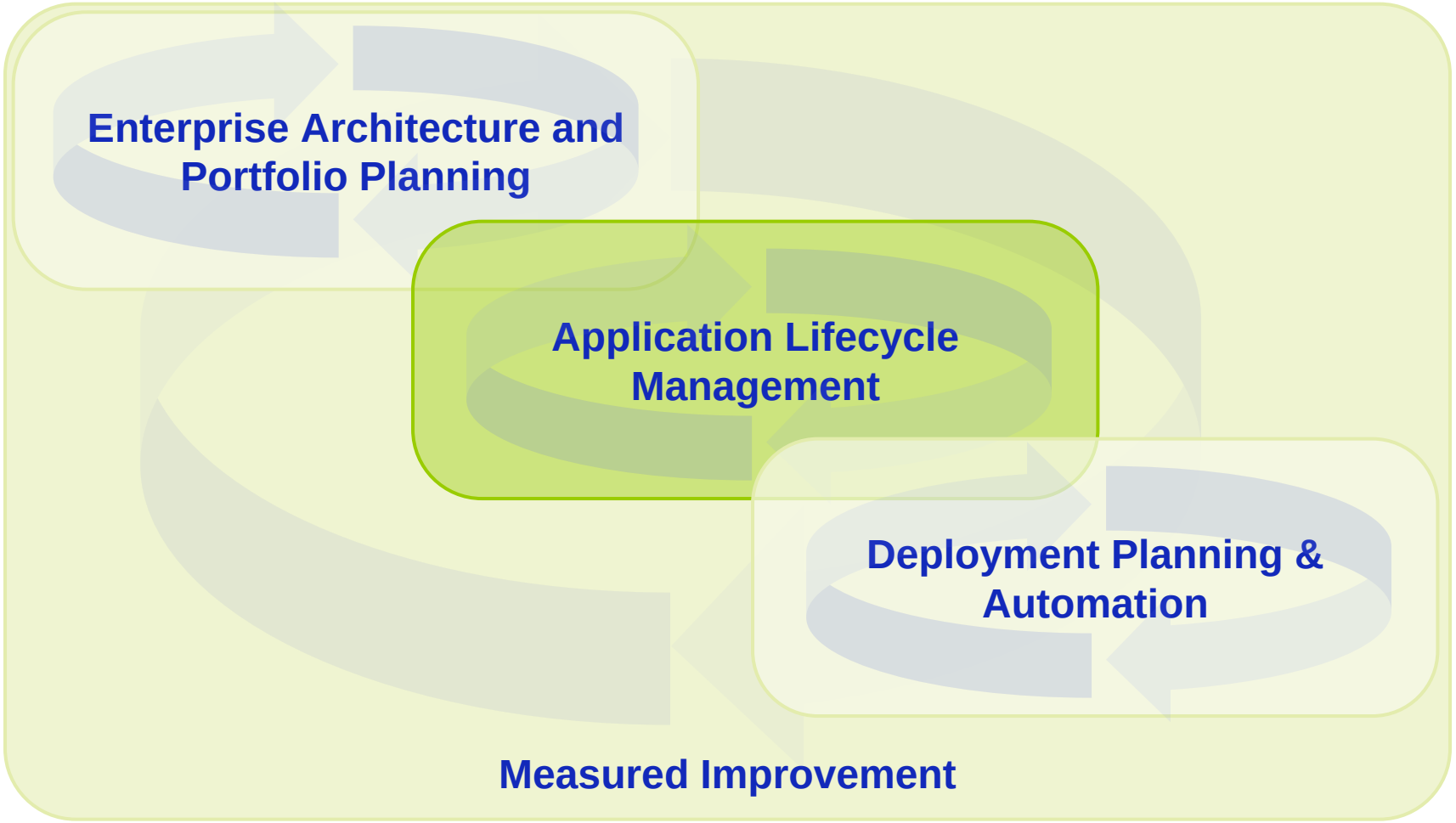


- Forrester estimates that ongoing operations and maintenance consume ***66%*** of IT budgets
- While new projects and software initiatives represent only ***34%***

Key Software development lifecycles



Key Software development lifecycles



ALM is about connecting the disciplines

Project/Planning

- Business Drivers
- Iterations
- Sign-off
- Contract
- Risk Assess
- User Involvement

Requirements

- Use Cases
- Nonfunctional
- Sign-off
- Contract
- Risk Assess
- Threat Model
- Test Requirements

Development

- TDD
- Build Management
- Static Analysis
- BVT
- Source Management
- Pair Programming/ Code Review

Testing

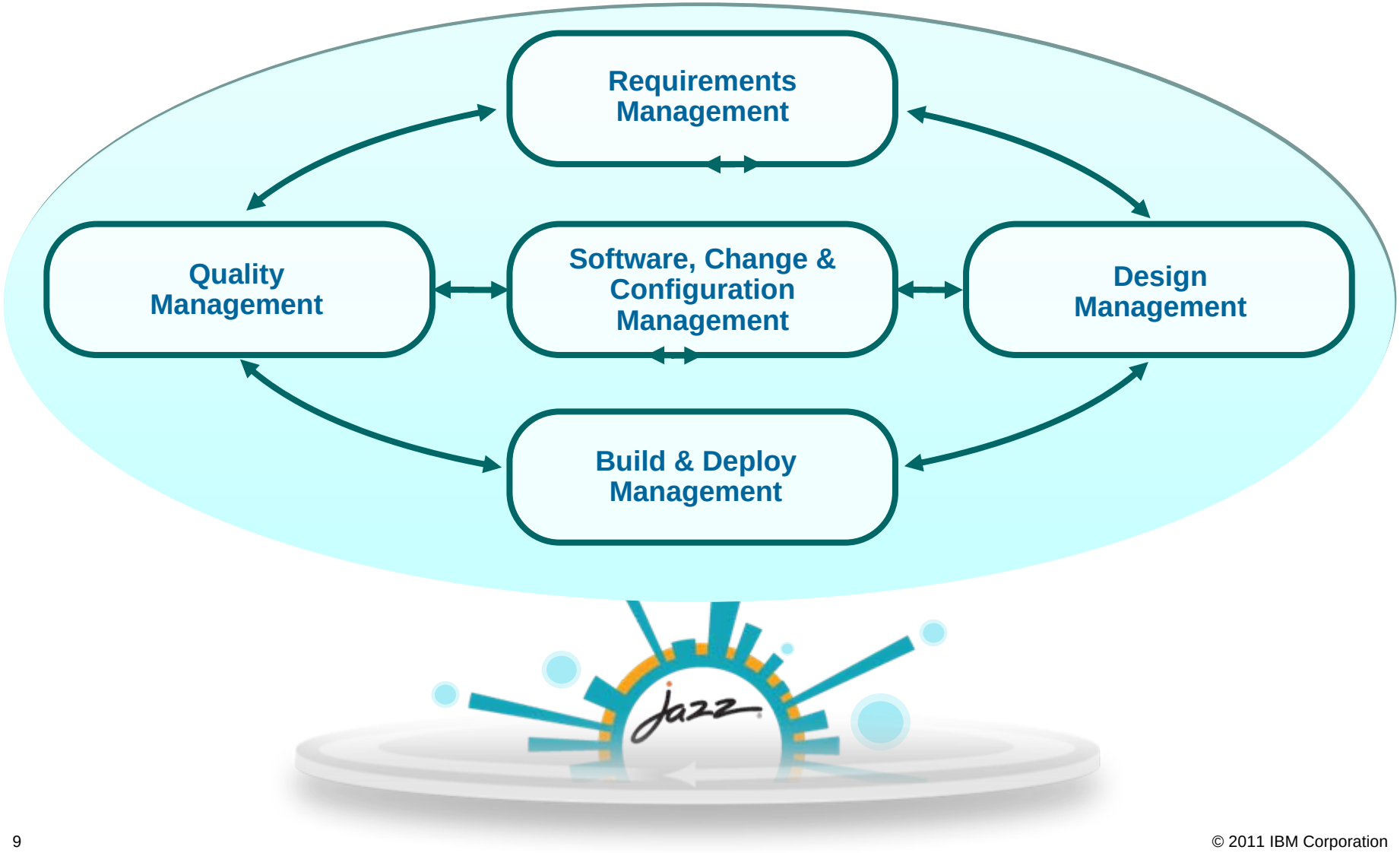
- Scenario-Driven Automation
- Exploratory Test
- User Involvement
- Contract Validation

Continuous Learning and Feedback

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

Rational Application Lifecycle Management (ALM)

Modular, open and extensible



1 Overview

2 IBM Rational Collaboration Lifecycle Management

3 Delivering the 5 ALM imperatives

4 Proven results

An ALM solution powered by Jazz

Rational solution for Collaborative Lifecycle Management

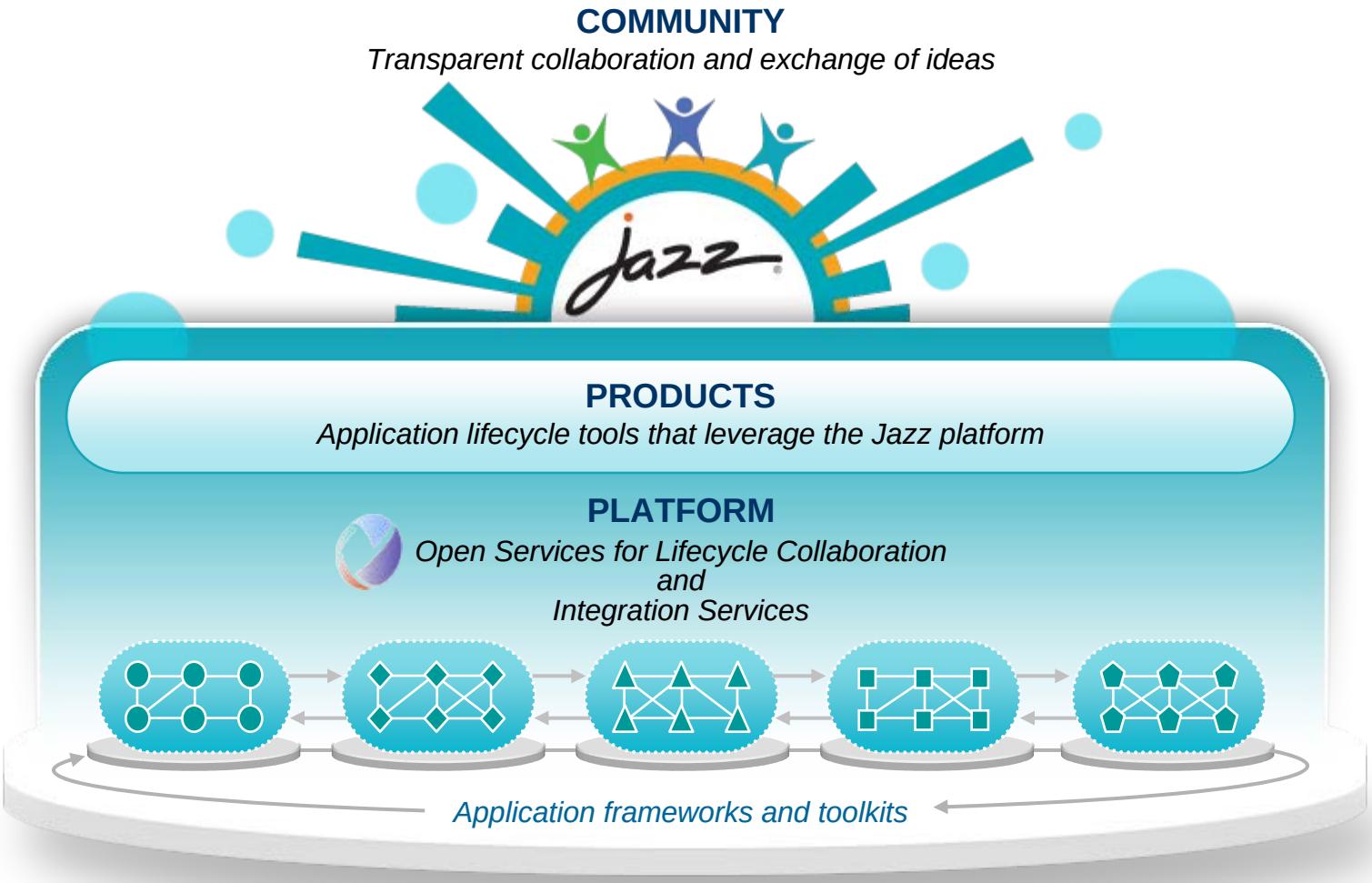
CREATE SOFTWARE

Real-time Planning, Lifecycle Traceability, Team Collaboration, Development Intelligence, Continuous Improvement

Rational Requirements Composer	Rational Team Concert	Rational Quality Manager
Requirements Management	Planning, Change, Configuration & Build Management	Quality Management

*extensions

Jazz provides open collaboration across the software and systems lifecycle



Learn more at: https://jazz.net/about/

Open Services for Lifecycle Collaboration (OSLC)

An initiative aimed at simplifying data linking and tool integration across the lifecycle

Barriers to sharing resources and assets among tools

- Multiple vendors, open source projects, and in-house tools
- Private vocabularies, formats and stores
- Entanglement of tools with their data

Open Services for Lifecycle Collaboration

- Community Driven – specified at **open-services.net**
- Specifications for ALM, PLM and DevOps Interoperability
- Inspired by Internet architecture
  - Loosely coupled integration with "just enough" standardization
  - Common resource formats and services
- A different approach to industry-wide proliferation

Learn more at: http://open-services.net/

1 Overview

2 IBM Rational Collaboration Lifecycle Management

3 Delivering the 5 ALM imperatives

4 Proven results

What is required to deliver end-to-end visibility across teams, tools and projects?

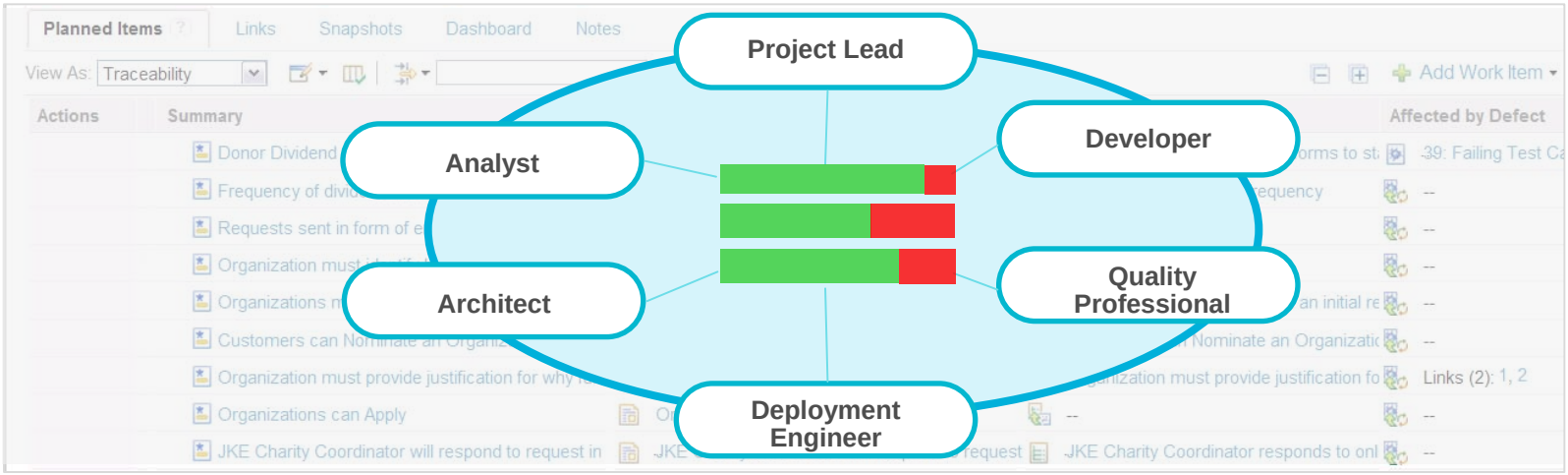
Criteria for effective lifecycle management: ALM imperatives

1. Real-time planning
2. Lifecycle traceability
3. In-Context collaboration
4. Development intelligence
5. Continuous improvement

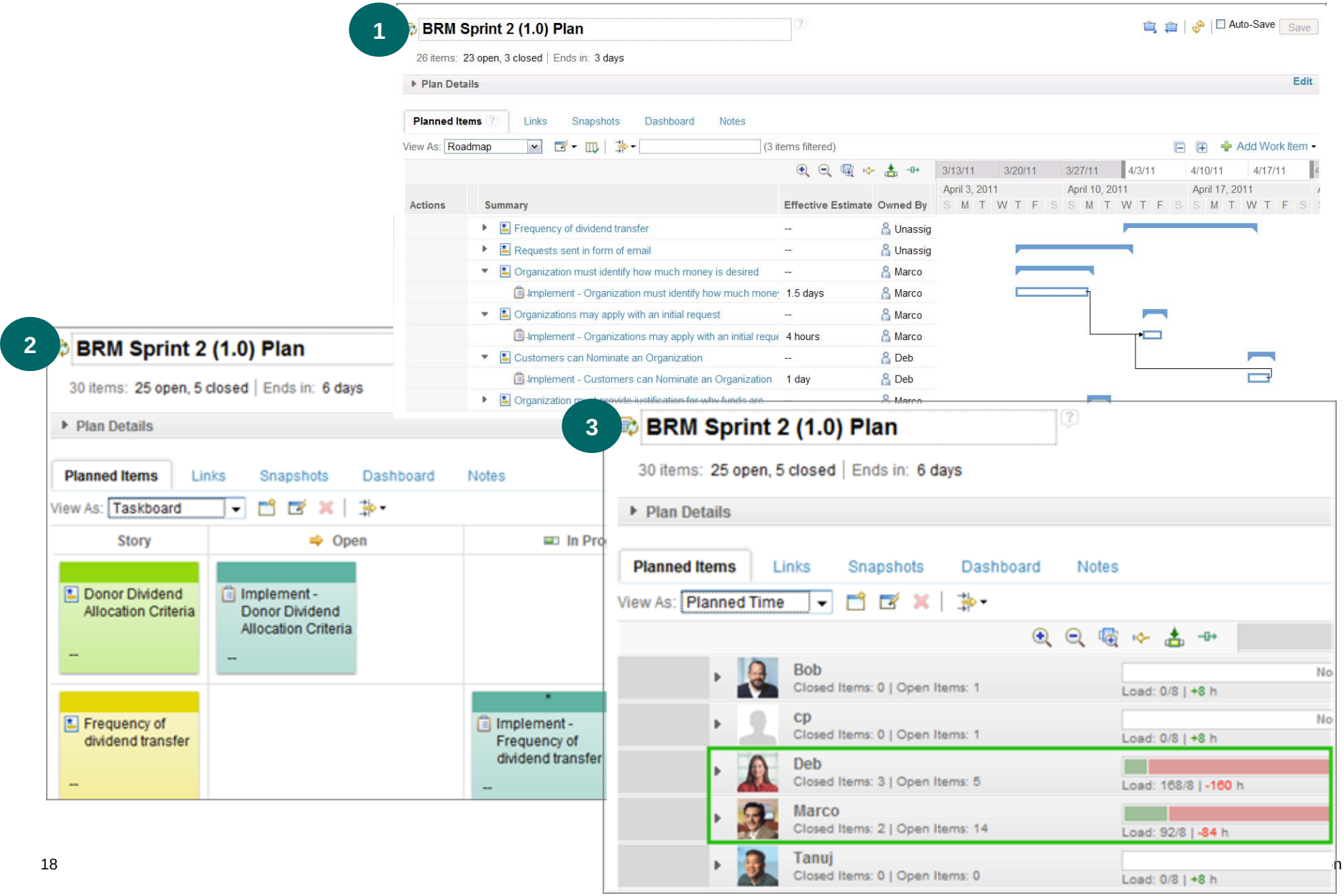
What is Real-time Planning?

Real-time Planning improves time to delivery by:

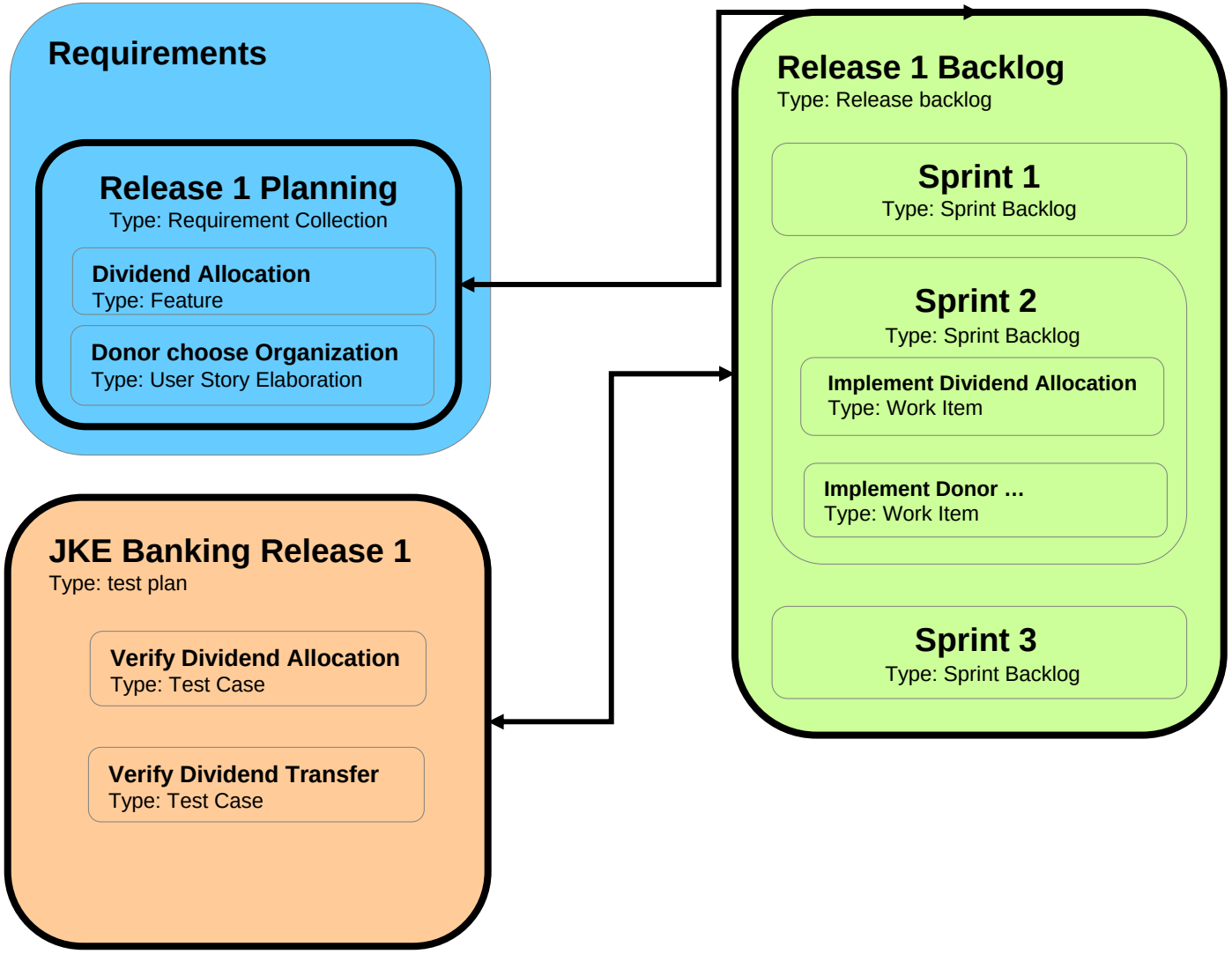
- Providing a **single plan** that spans requirements, development, and test, ensuring a team understanding of the overall scope of a project
- Allowing **everyone to participate** in keeping the plan current and accurate
- **Integrating planning with execution,** ensuring the entire team understands the true project status
- Helping teams **respond to the unexpected** in a timely manner ensuring the team stays on schedule



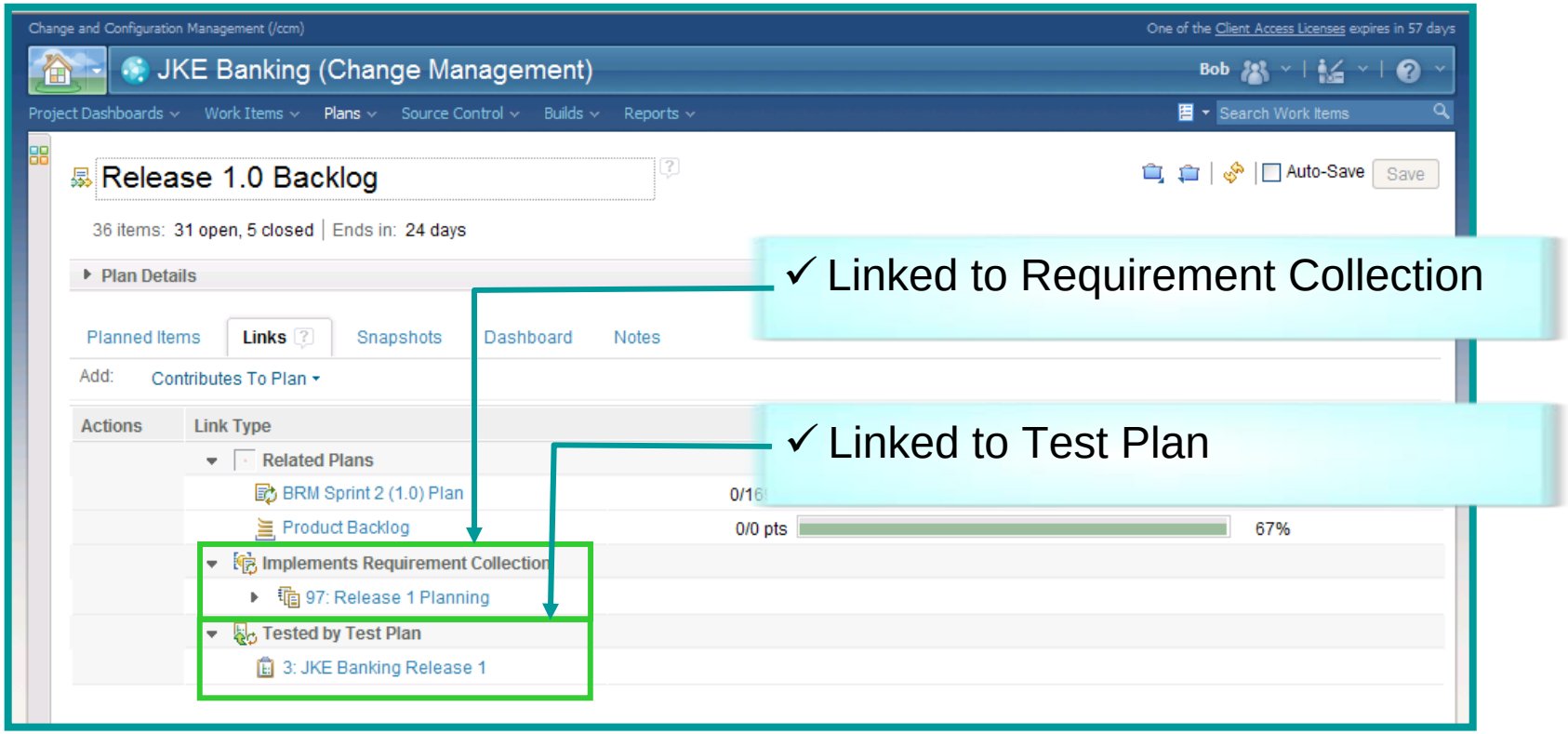
One plan - Multiple views facilitate detailed analysis



Key planning relationships

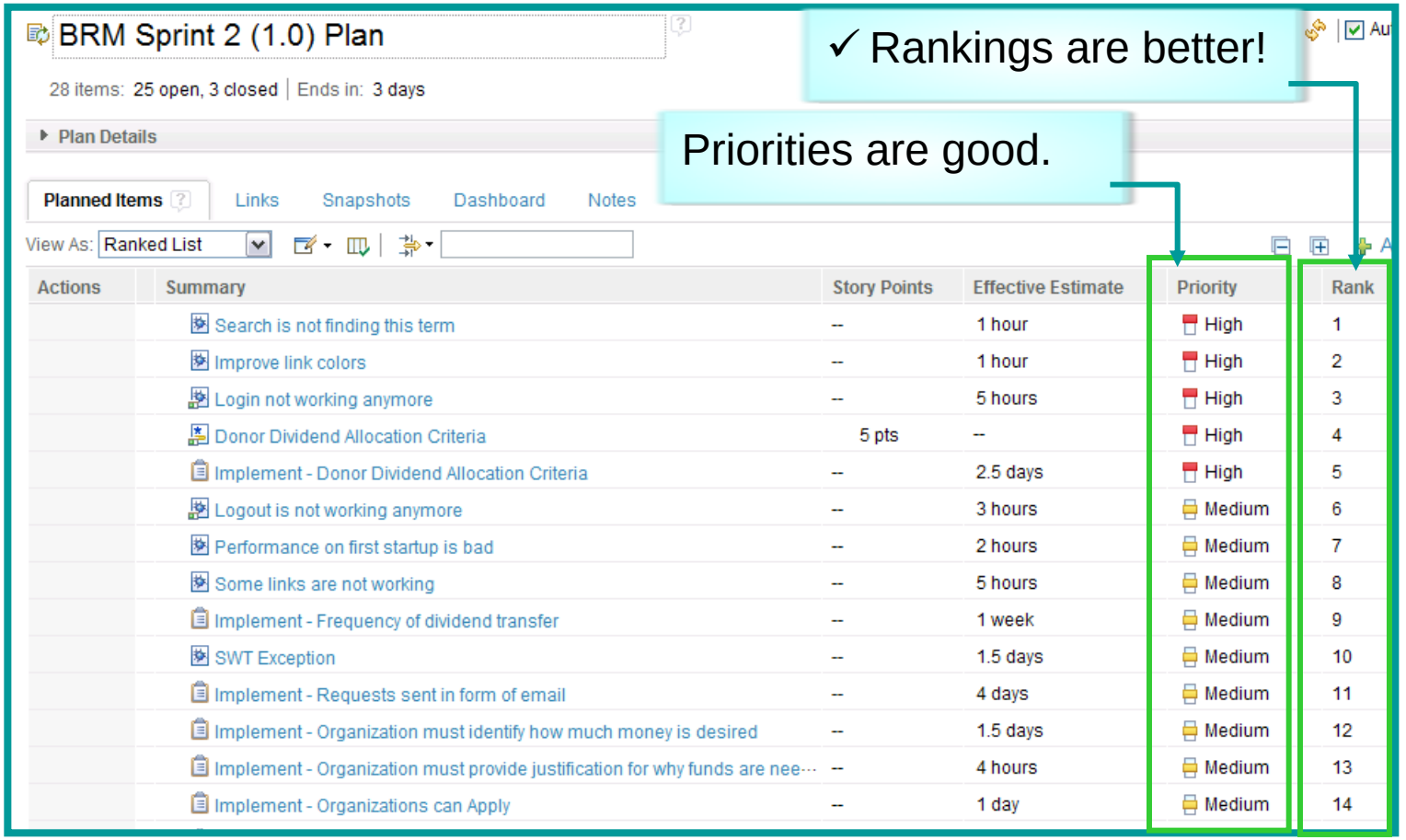


Insure alignment across the business, development and test teams



Rankings improve clarity

Team clearly understand the goals



BRM Sprint 2 (1.0) Plan

28 items: 25 open, 3 closed | Ends in: 3 days

Plan Details

Planned Items | Links | Snapshots | Dashboard | Notes

View As: Ranked List

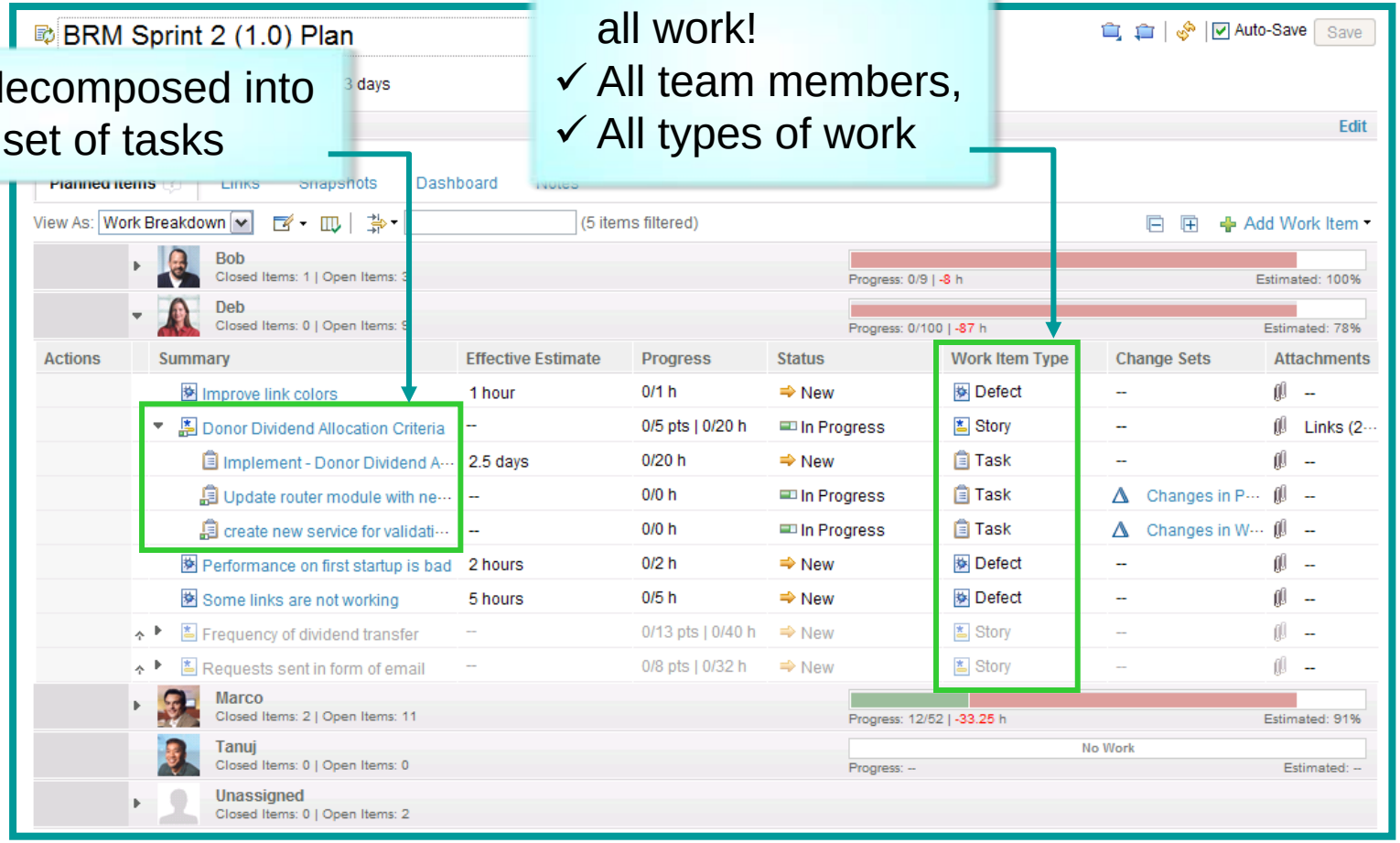
Actions	Summary	Story Points	Effective Estimate	Priority	Rank
	Search is not finding this term	--	1 hour	High	1
	Improve link colors	--	1 hour	High	2
	Login not working anymore	--	5 hours	High	3
	Donor Dividend Allocation Criteria	5 pts	--	High	4
	Implement - Donor Dividend Allocation Criteria	--	2.5 days	High	5
	Logout is not working anymore	--	3 hours	Medium	6
	Performance on first startup is bad	--	2 hours	Medium	7
	Some links are not working	--	5 hours	Medium	8
	Implement - Frequency of dividend transfer	--	1 week	Medium	9
	SWT Exception	--	1.5 days	Medium	10
	Implement - Requests sent in form of email	--	4 days	Medium	11
	Implement - Organization must identify how much money is desired	--	1.5 days	Medium	12
	Implement - Organization must provide justification for why funds are nee···	--	4 hours	Medium	13
	Implement - Organizations can Apply	--	1 day	Medium	14

Plan at the right level of detail

- Everyone participates in planning

✓ Stories decomposed into the right set of tasks

✓ Plans include all work!
✓ All team members,
✓ All types of work



Test your real-time planning capabilities

Can your team…

- Plan across the entire team?
- Plan for waterfall, iterative and agile environments?
- Integrate planning with execution?
- Instantly see the impact of a change in project scope or resources?

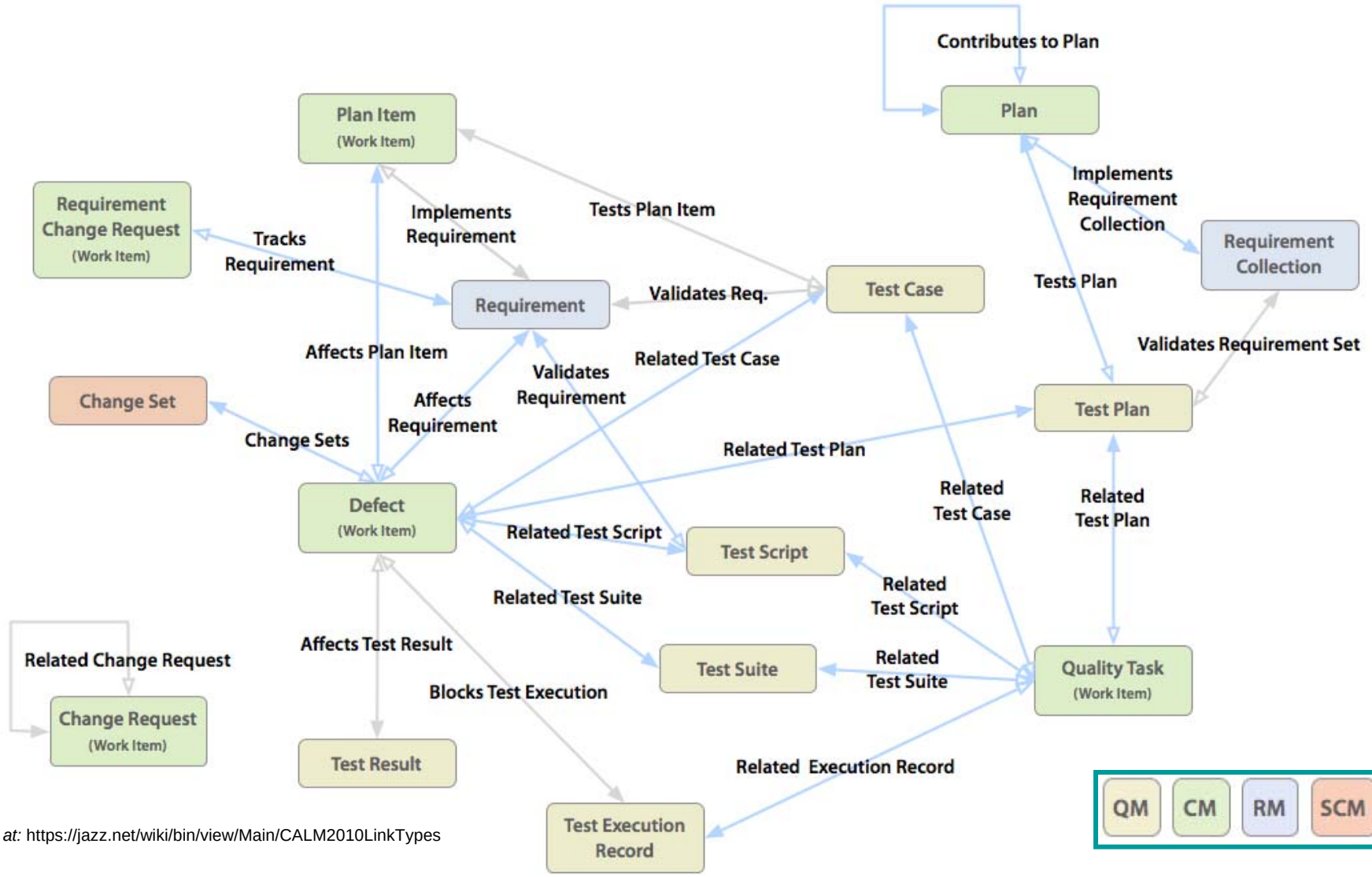
What is Lifecycle Traceability?

Lifecycle Traceability improves quality by:

- Establishing **relationships between software artifacts**
- Helping you **identify and close artifact gaps,** ensuring **coverage** across disciplines
- Provides visibility into the **completeness** of planned items by inspecting all related artifacts
- Provides easy **access to related artifacts** ensuring everyone shares the same view
- Delivers transparency which enables everyone to make **fully informed decisions** based business priorities

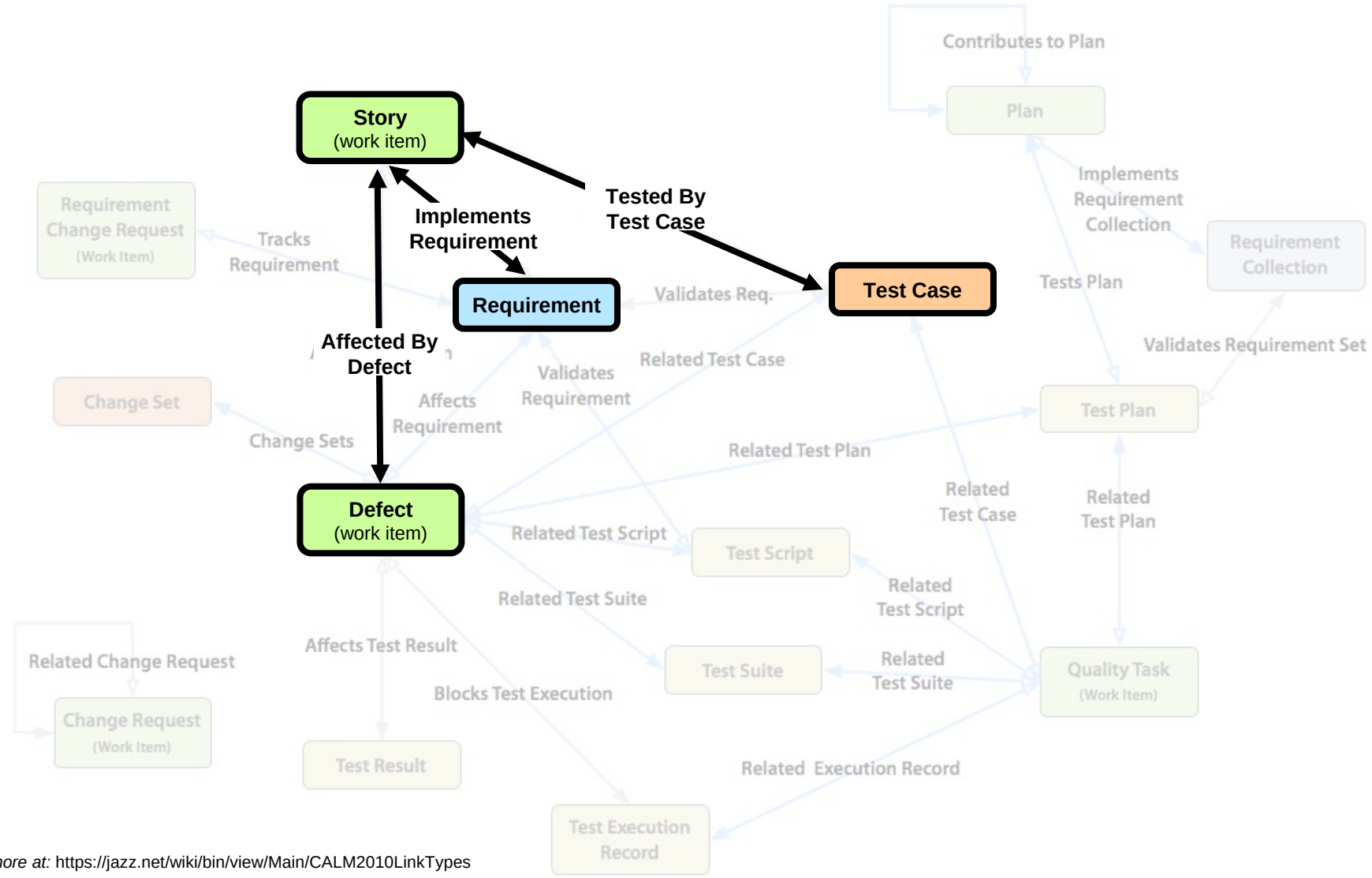
Instant access to details from any point in the development process

CLM Link Types



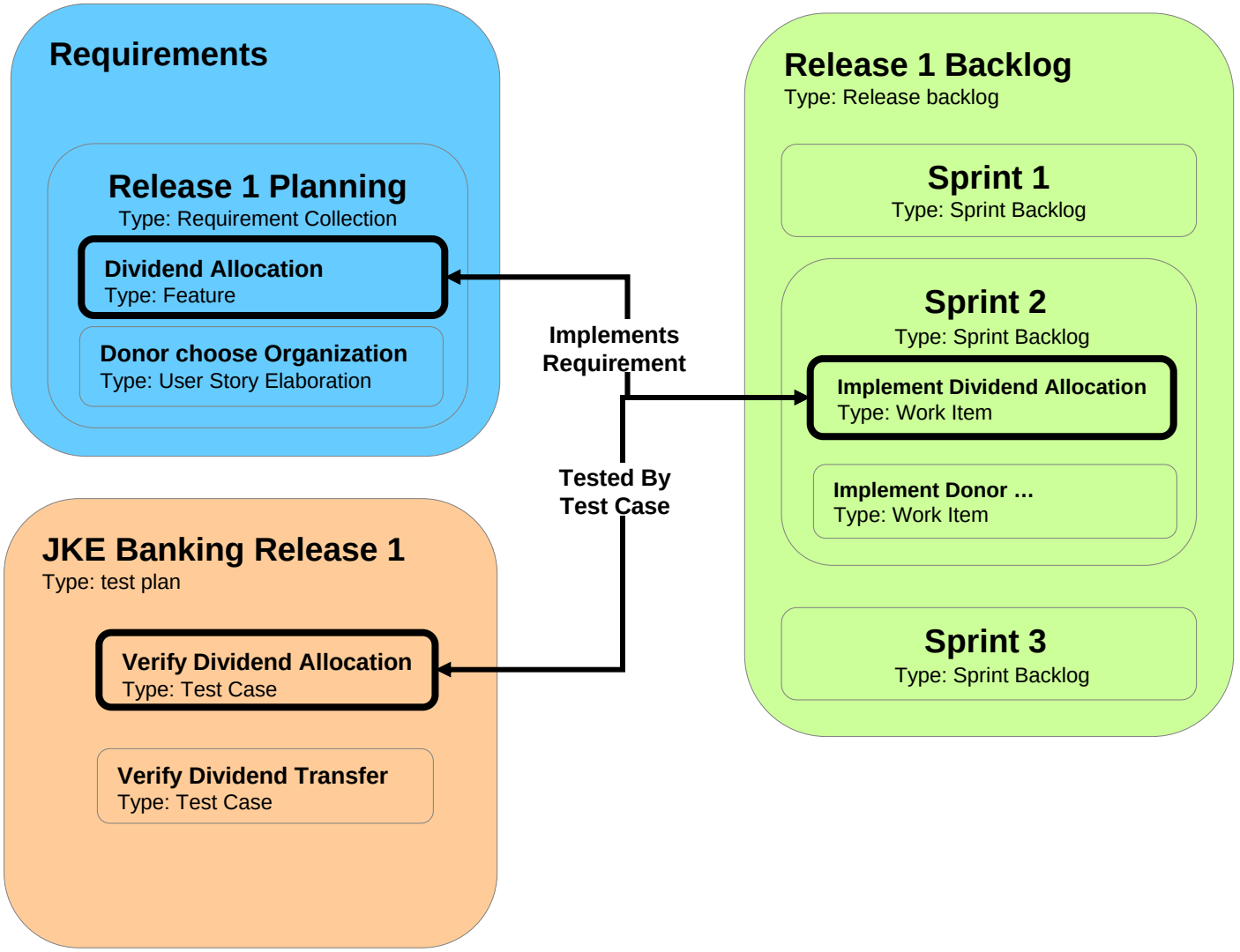
Learn more at: https://jazz.net/wiki/bin/view/Main/CALM2010LinkTypes

SCRUM process Template



Learn more at: https://jazz.net/wiki/bin/view/Main/CALM2010LinkTypes

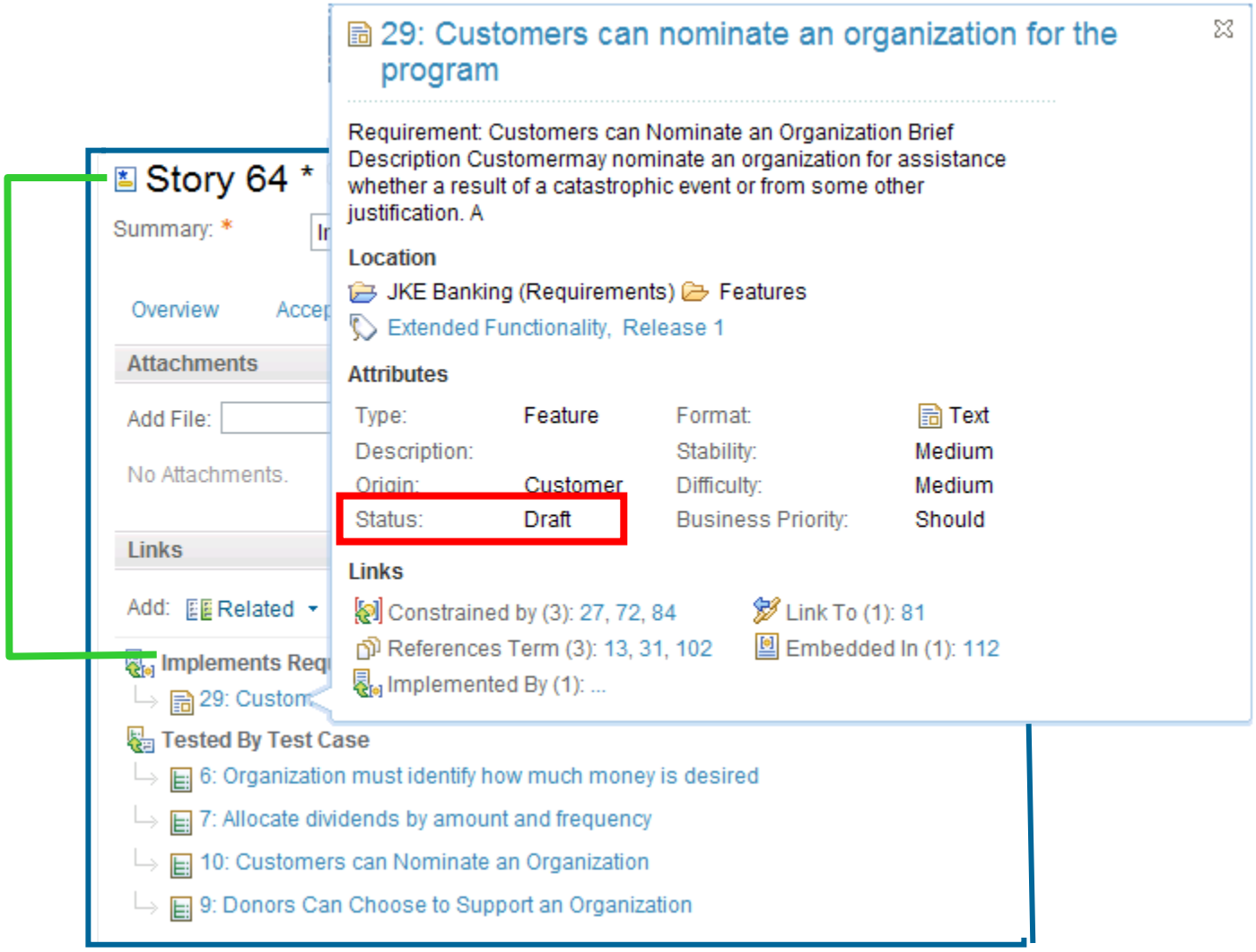
Key lifecycle traceability relationships



Developer have visibility into the Requirement they will be implementing

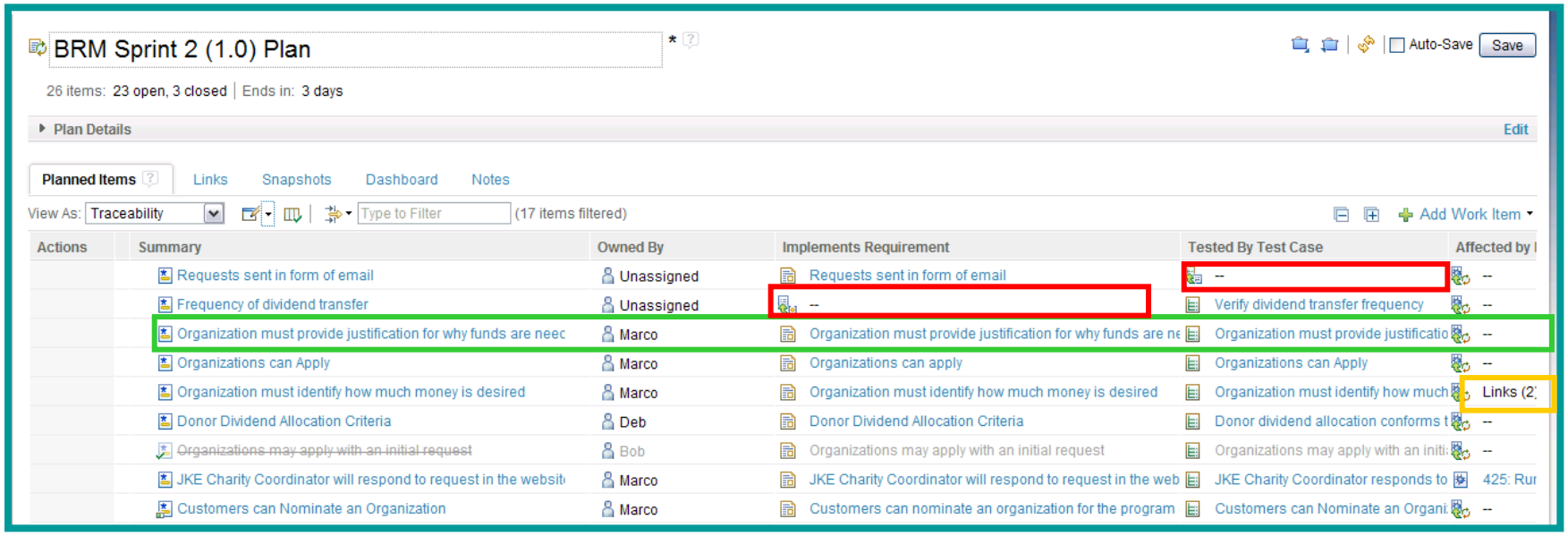
For the Developer

- ✓ **Rich Hovers** provides insight at your finger tips reducing time to value.
- ✓ **Clickable links** provides valuable information on requirement



Team Leads have visibility into coverage & completeness

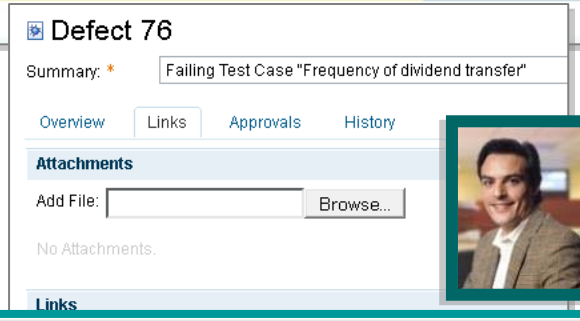
- Proactively respond to gaps as they surface through out the project
- Issues quickly highlighted and resolved



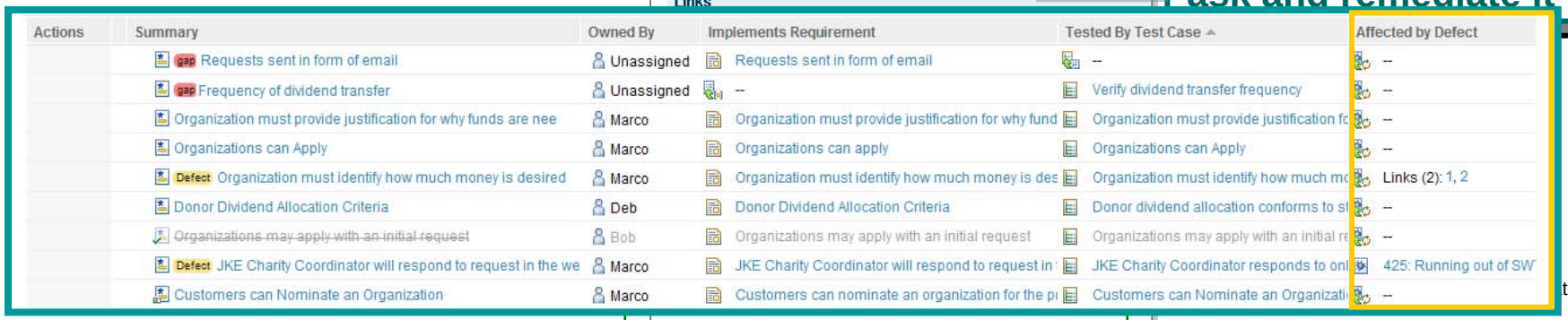
Automated defect traceability reduces costs and improves quality

- 4-clicks to submit a defect automatically linked to impacted artifacts
- Test results are recorded and linked to test cases, and associated requirements
- Test results can be linked to software builds
- Everyone has visibility into the defects, their impact, and the action taken to resolve them

Testers execute tests and submit defects



Developers can see the exact test failure without having to ask and remediate it



Test your lifecycle traceability capabilities

Can your team answer these questions…

Analyst

Which requirements are addressed in this iteration?

Are all of the requirements tested?

What's the quality of the high priority requirements?

What defects are reported against which requirements?

Project Manager

Can we pass an audit?

Are we ready to release?

What defects were resolved in this release?

What tradeoffs can we make to release on time?

Developer

What requirements am I implementing?

What test uncovered this defect, on which environment and what build?

How can I recreate the last version to do a patch?

What changes occurred overnight?

Release Engineer

How can I standardize when teams use different tools?

Are build times getting longer or shorter?

How can I speed up my builds?

Quality Professional

What is the quality of the build?

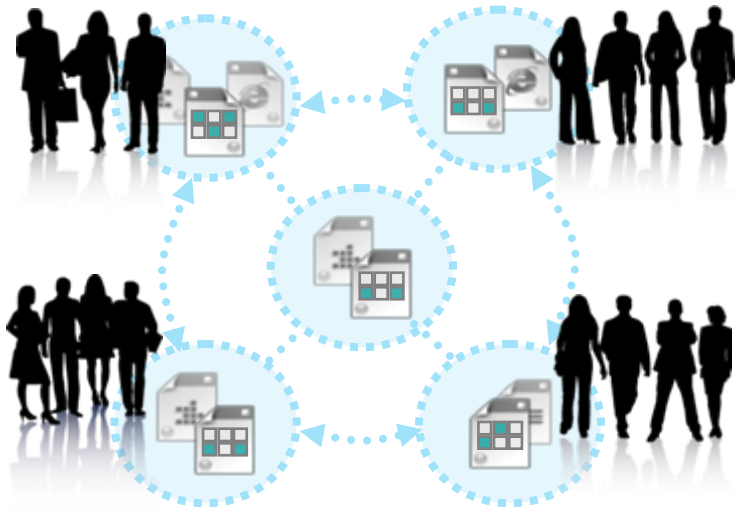
What has changed that I need to test?

What defects have been addressed since the last build?

What is In-Context Collaboration?

In-Context Collaboration improves product value by:

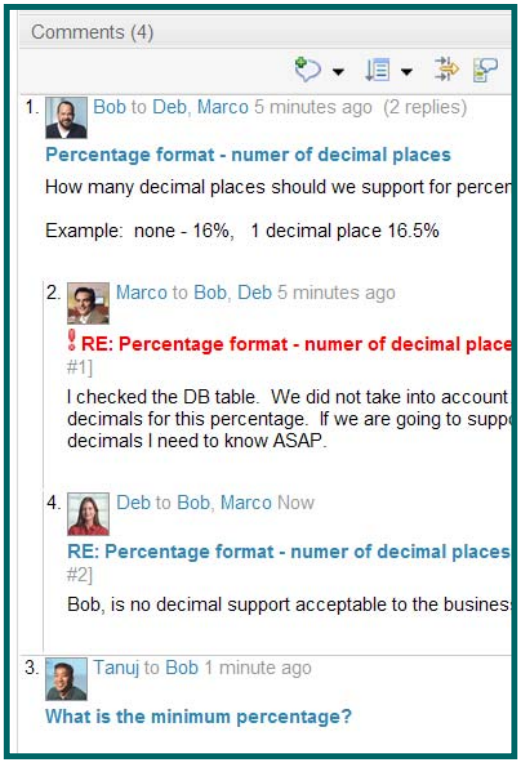
- Making **information immediately accessible** to all team members in the context of their work
- Empowering teams to collaborate on and review software development artifacts so they can **incorporate feedback early and often**
- Providing **single source of truth** hosted in a shared repository so that team members can collaborate effectively around the globe



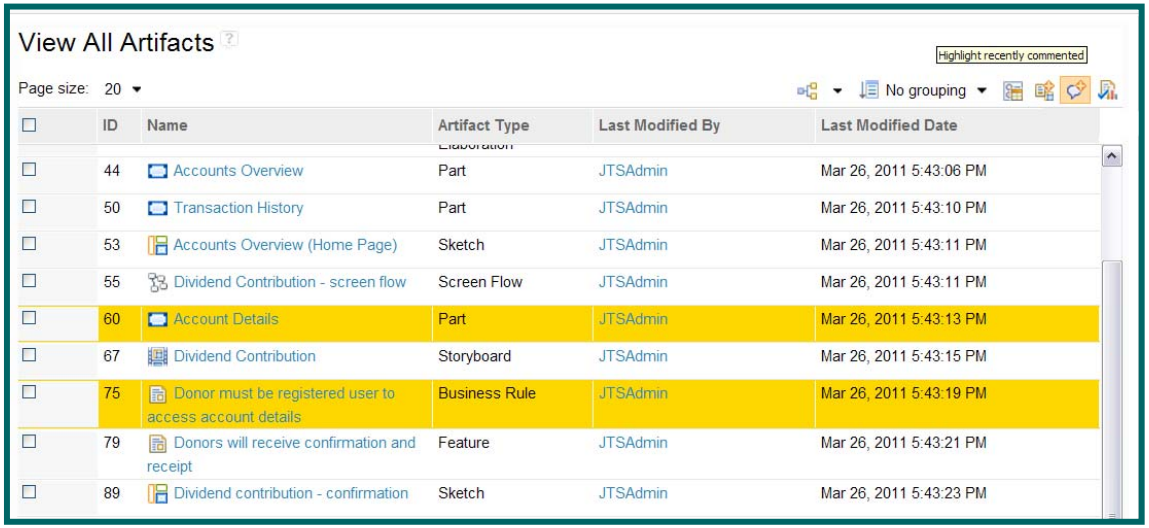
Rich hovers automate information access without having to click

- ✓ Rich hovers provide quick access to information to determine if additional details are required!

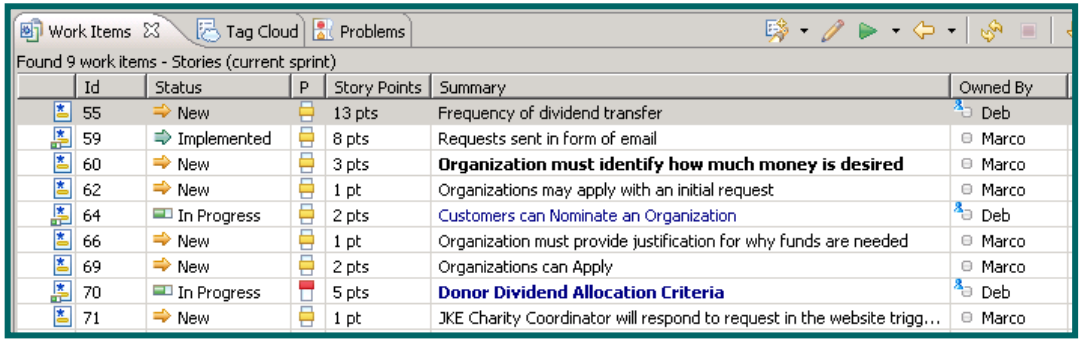
In-Context Collaboration always shows the latest



Threaded discussions on requirements

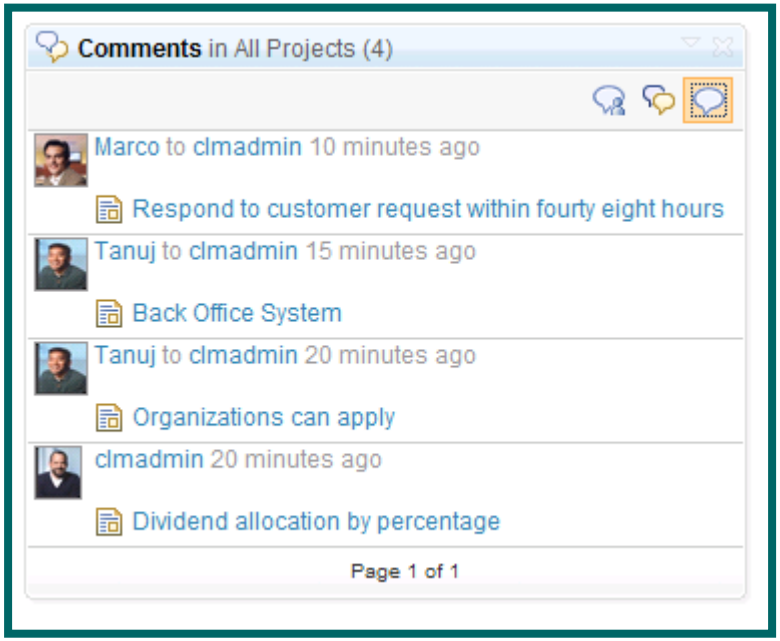


Recent discussions highlighted on requirements



Unread work Items bolded for developers

Viewlets focus the team on recent comments and changes



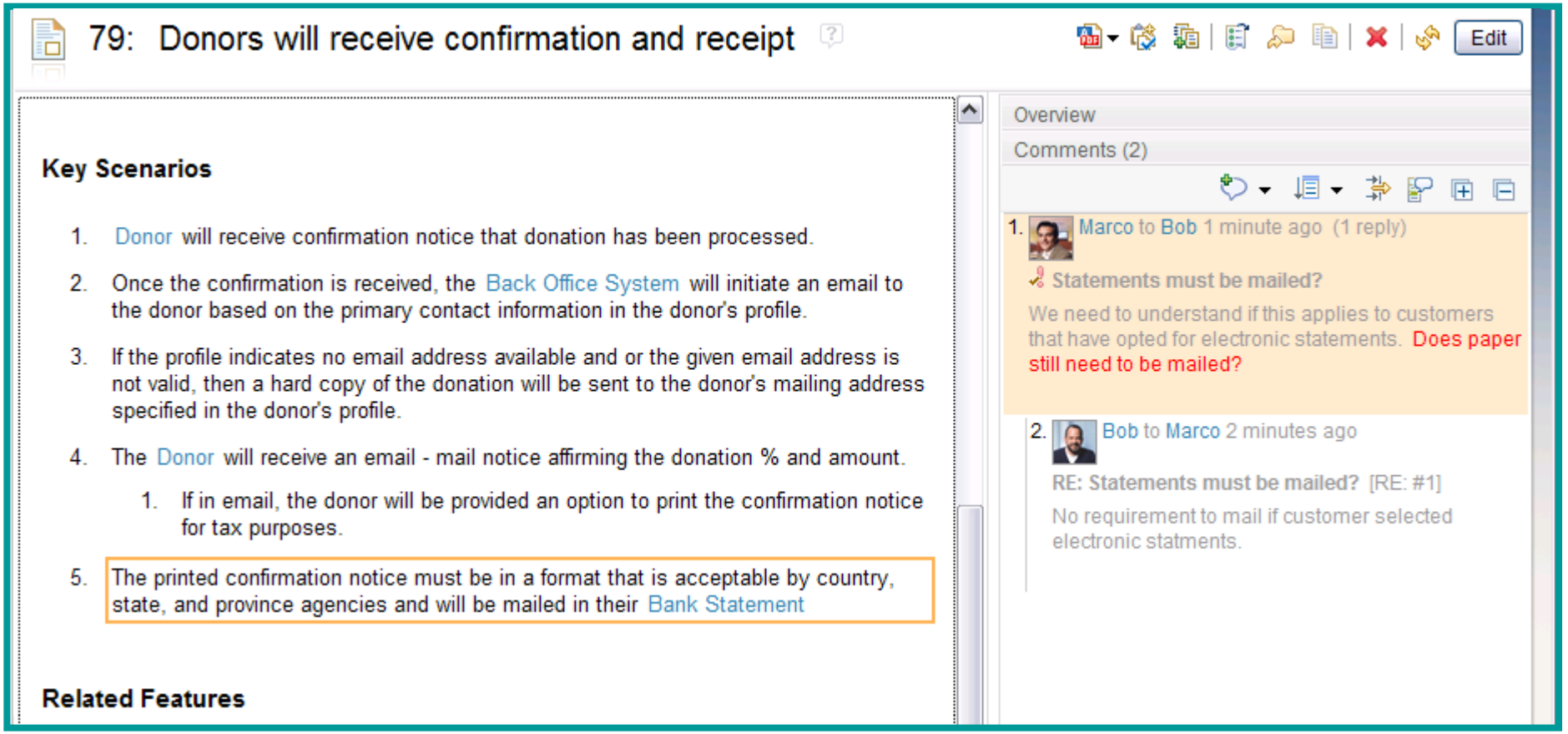
Requirements comment updates

Feeds focused on Requirement comments focus stakeholders on changes and strip out the noise of attributes changes etc..

Any changes to a Requirement

Feeds on all changes give project leads visibility into all requirements changes so that know exactly what is changing by whom and can access immediately assess the impact for the team.

Comments on requirements details are in-context



A click on the comment highlights the text under discussion.

Test your In-Context Collaboration capabilities

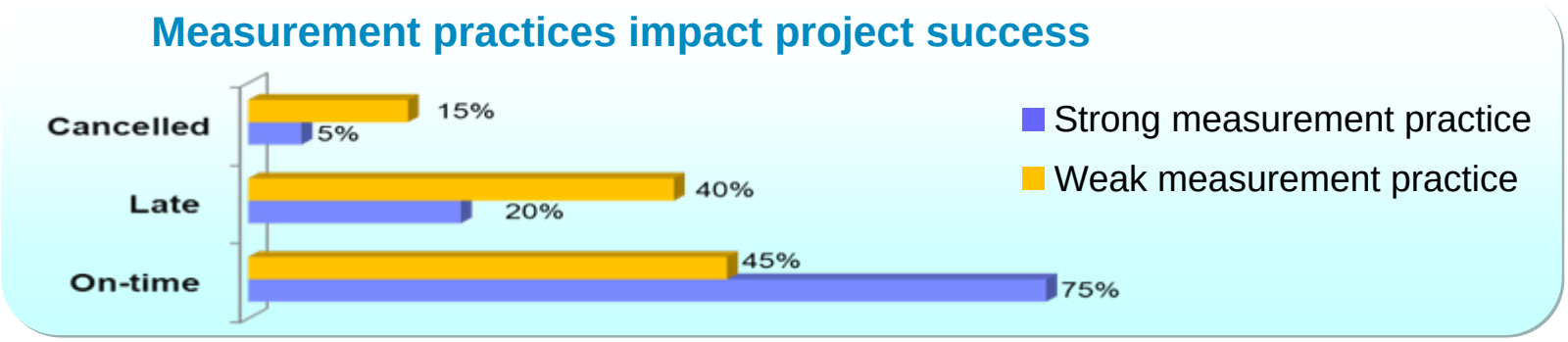
Can your team…

- Easily access the “single source of the truth” through linked artifacts?
- Quickly grasp the “who, what, when and why” of team activities?
- Bring new team members up-to-speed quickly?
- Overcome the barriers of multiple time zones when working with outsourced and distributed team members?

What is Development Intelligence?

Development Intelligence improves predictability by:

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

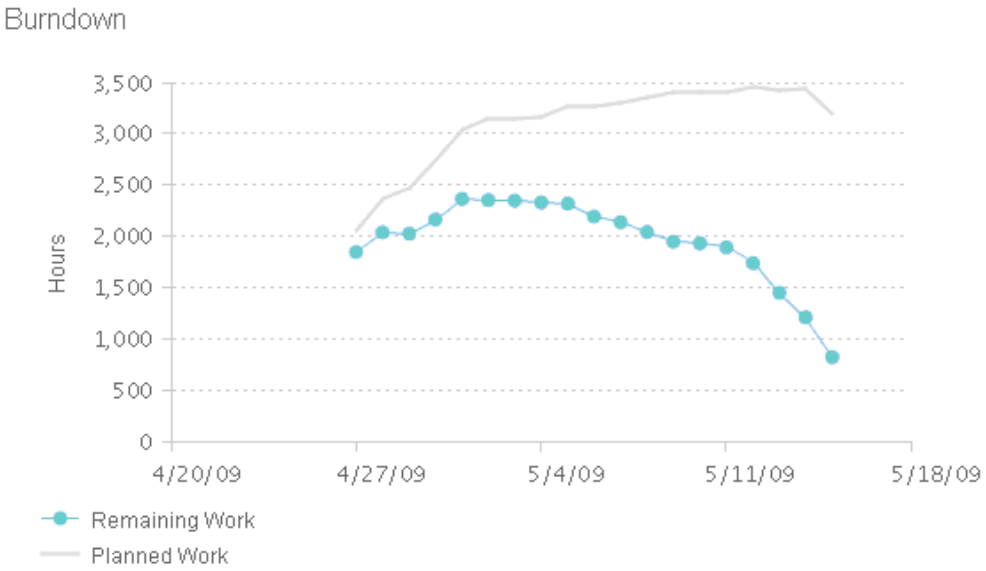


Source: Capers Jones, Measurement, Metrics and Industry Leadership, 2009 and Software Engineering Best Practices, McGraw Hill, 2010.

Measuring Time-to-Value (Schedule)

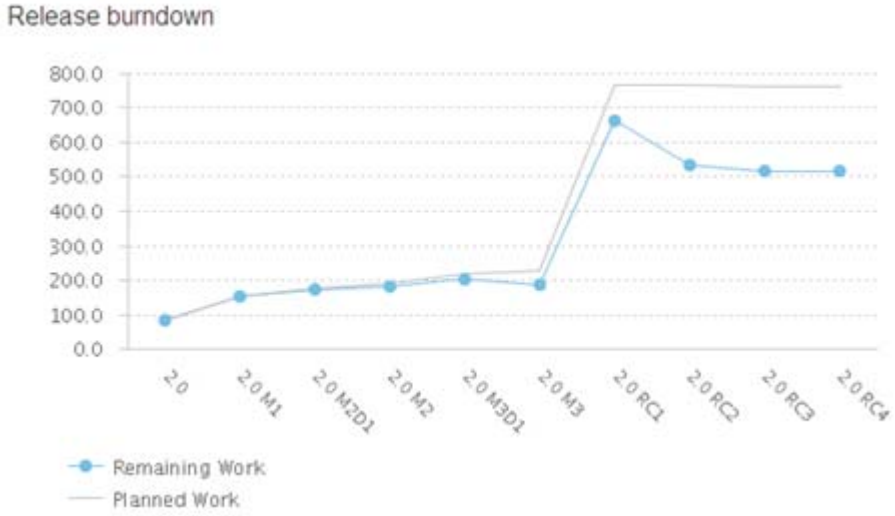
Iteration Burndown

Showing how much work is left to do in an iteration. It enables the team to adjust scope or resources to finish the iteration successfully.



Release Burndown

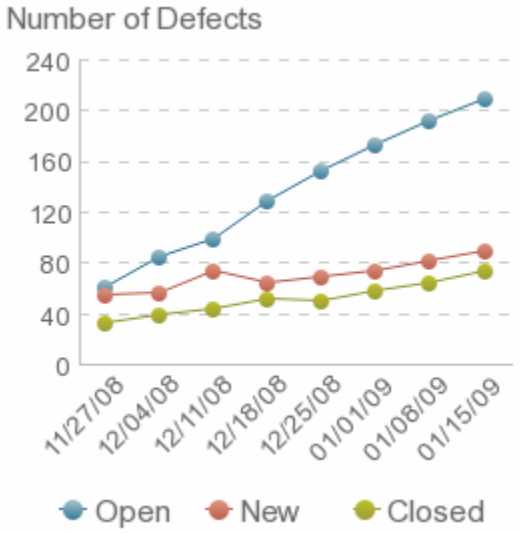
Shows the estimated functionality remaining to complete the current release.



Measuring Product Quality

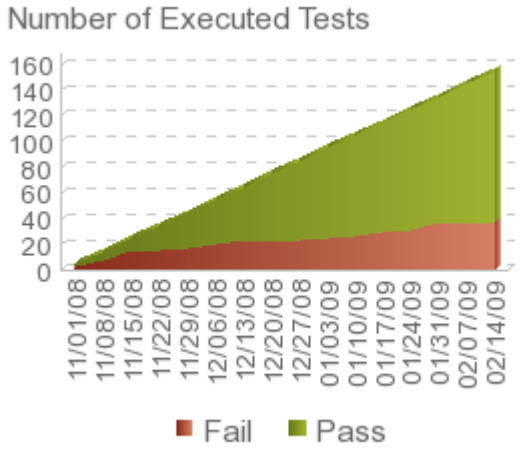
Defect Trends

Shows defect arrival and closure rates, determines the remaining defect backlog, projects the future defect arrival/close rate up to and post-ship

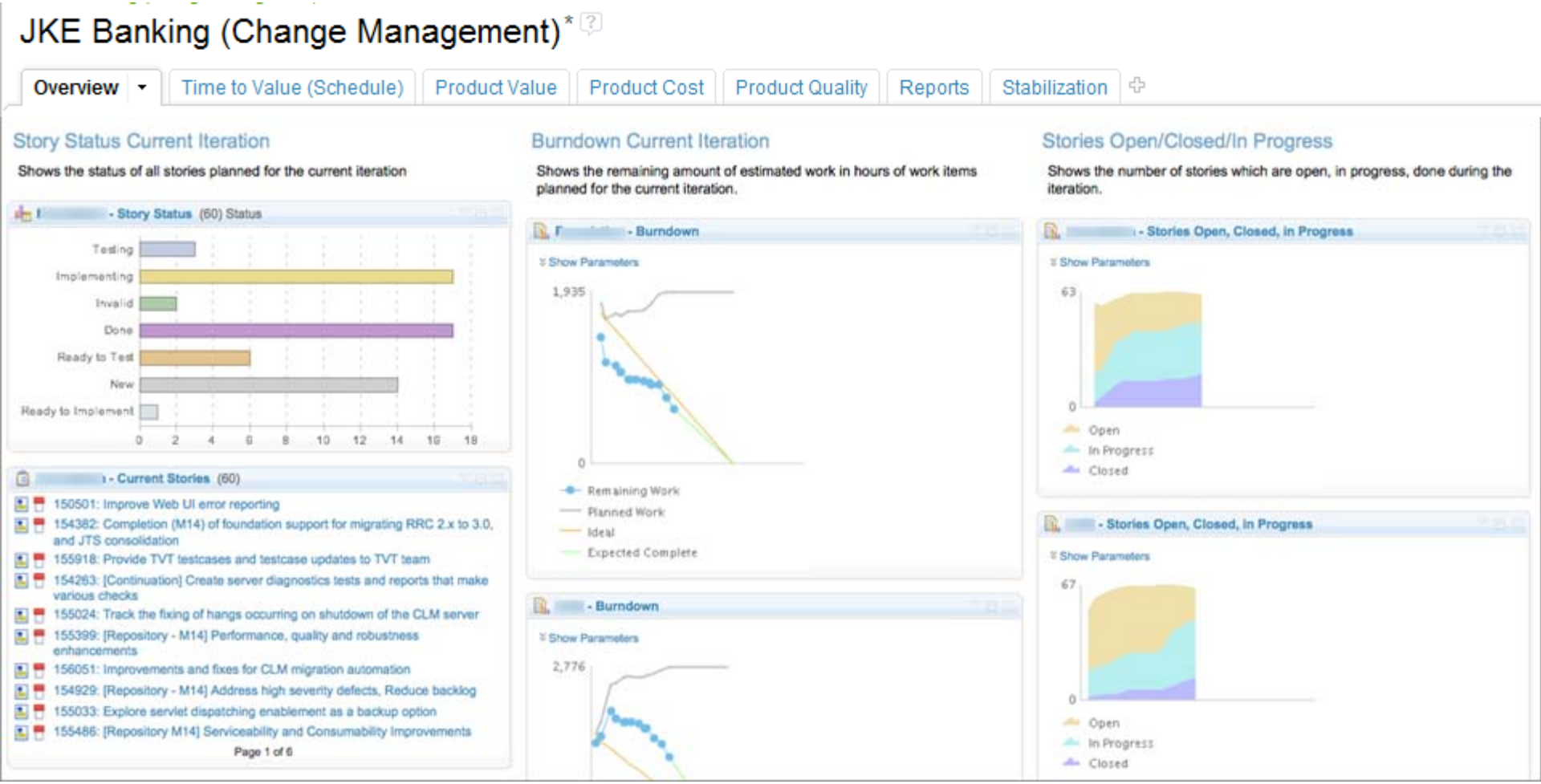


Test Execution Status

Monitors test completion and success



Use dashboards to provide that 1 view of project health



Test your development intelligence

Can your team…

- **Time-to-Value**
  - Produce the right capabilities according to the committed schedule?
- **Product Value**
  - Deliver a valuable product?
- **Product Cost**
  - Measure what we spending to deliver the system?
- **Product Quality**
  - Build a high quality system?
- **Predictability**
  - Manage the risk and uncertainty?

What is Continuous Improvement?

Continuous Improvement reduces cost by:

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

> *"Successful analytics requires taking it beyond software and reporting, and into the realm of management practices and operations improvement"*
> Information Management Online, February 23, 2011

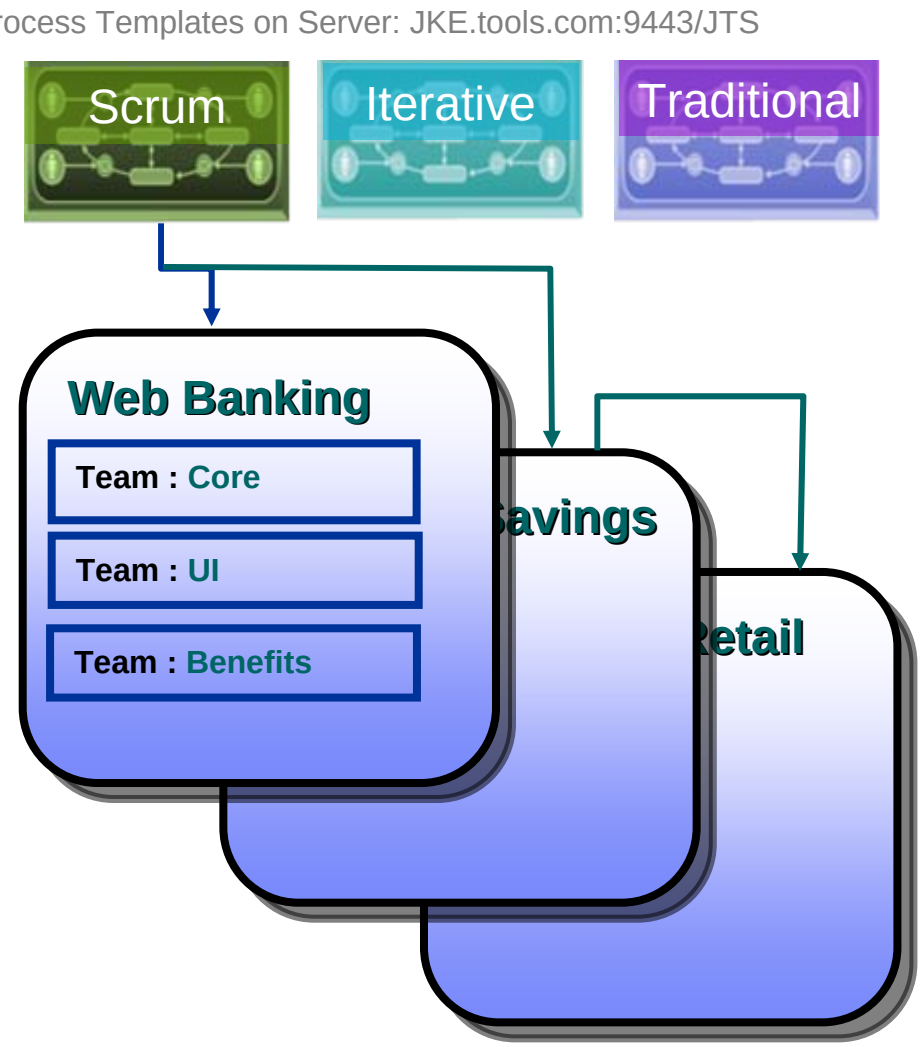
Adopt - Process Sharing

Leverage the Best practices from Rational and other teams

1. **Import/export Templates**
   Provides the ability to leverage best practices from Rational and other teams

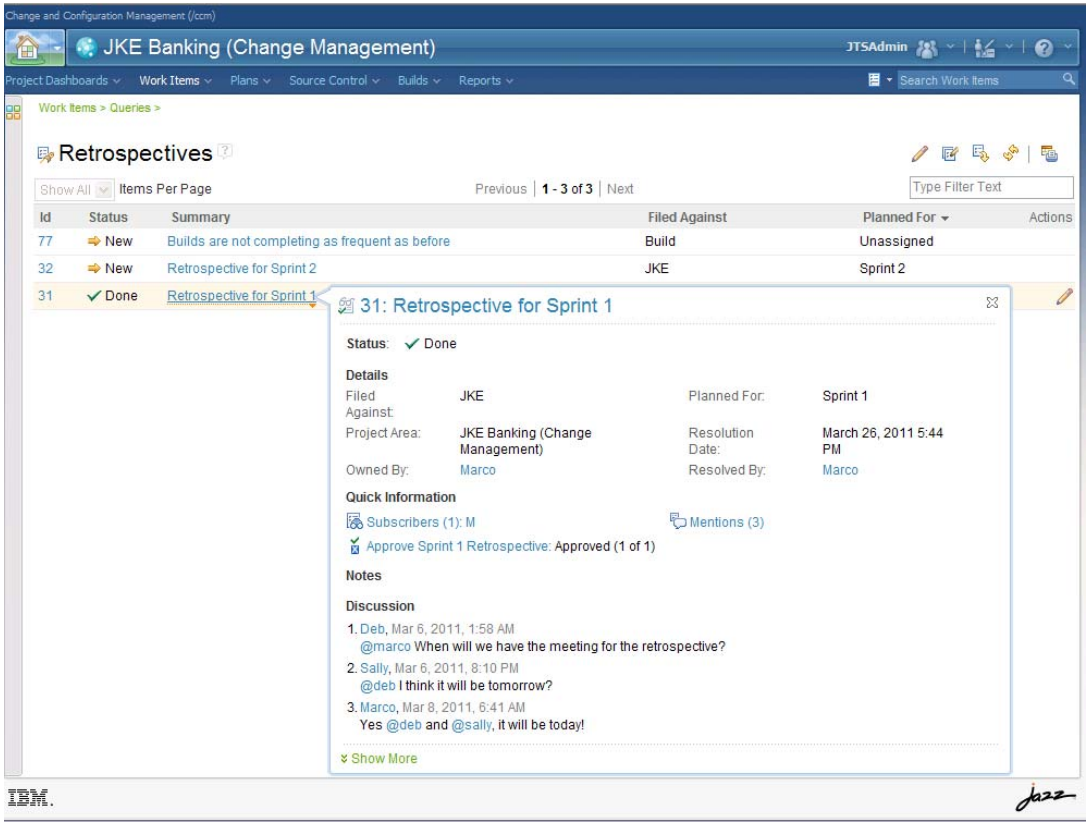
2. **Servers support multiple templates**
   Allowing administrators to support different processes.
   Teams improve their process independently.

3. **Process updates can be shared**
   Allows teams to control when they want to accept improvements.

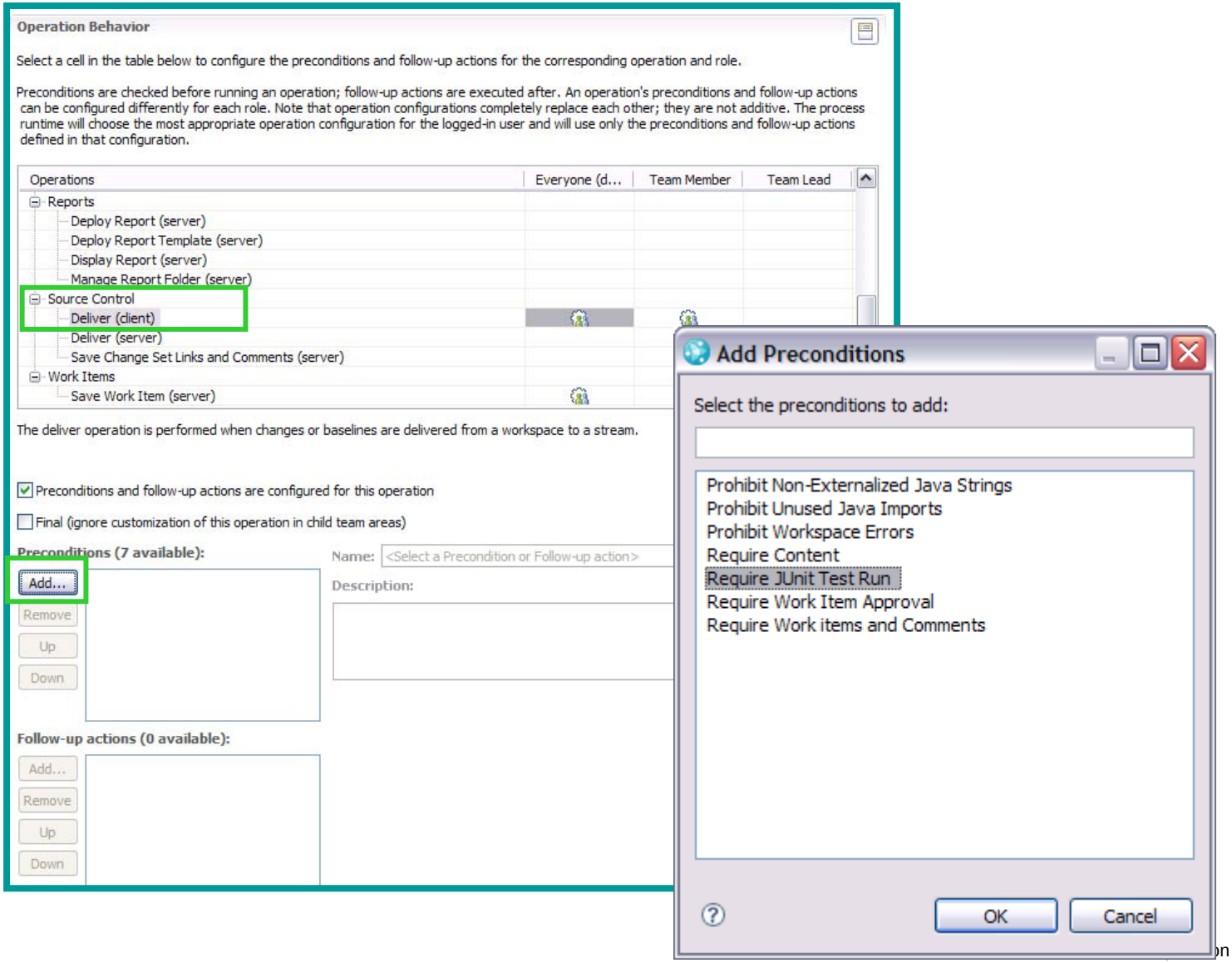


Team Retrospectives help identify areas of improvement

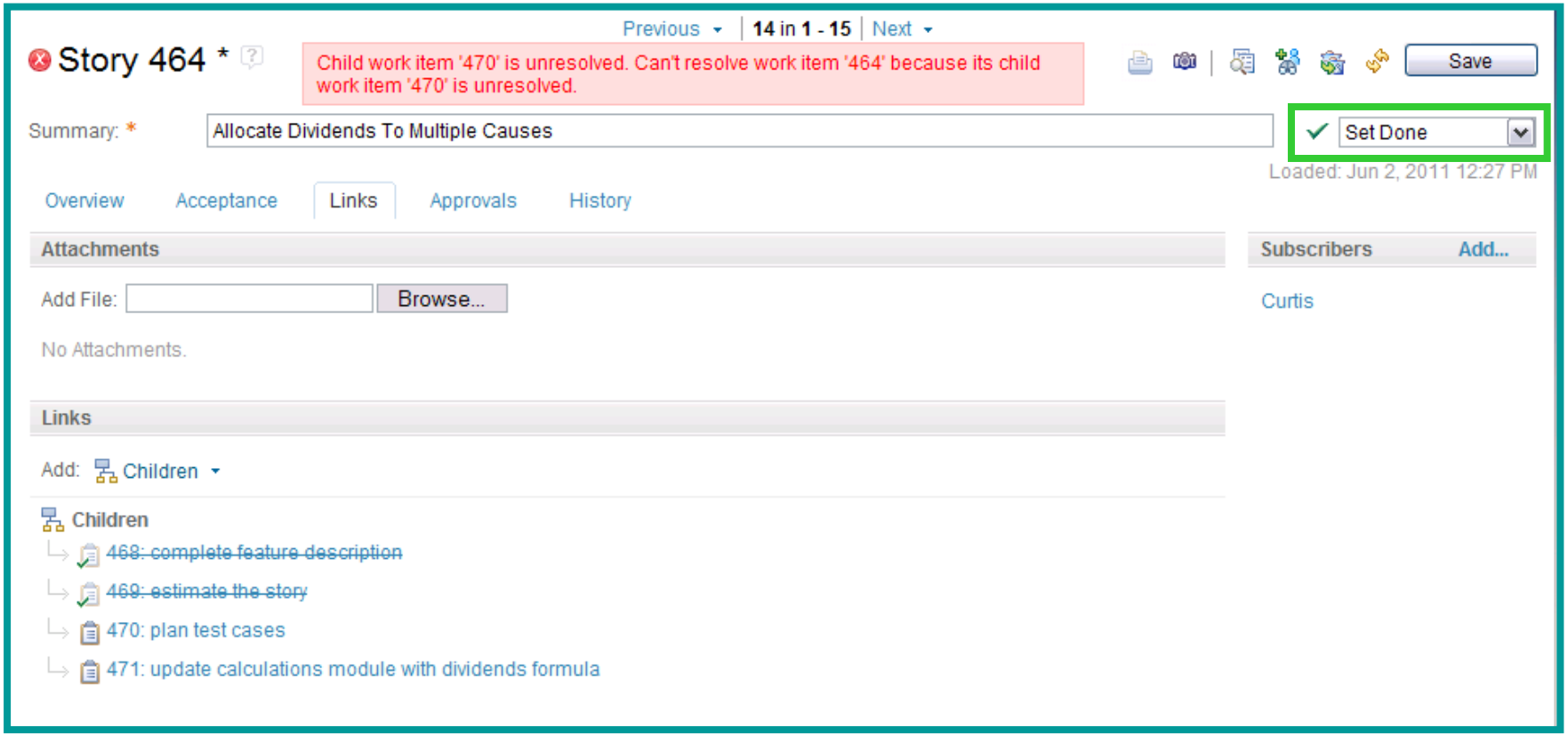
- Tracking Retrospectives help team members to collaborate on improvements.
  - Ranking of process improvements help to focus on where the team feels the biggest pain.
  - Metrics where available help support process improvements and show results.



Implement : Require Unit test before deliver



Plan tasks are not being completed before Story completed



Test your Continuous Improvement capabilities

Can your team…

- Leverage out-of-the-box process templates for traditional and agile workflows?
- Change process “on the fly” as part of a continuous feedback loop?
- Allow team leads to determine how strict or lax the “rules of the road” should be?
- Modify process enforcement over the life of a project, to encourage early-stage experimentation and end-game stability?

Criteria for effective lifecycle management: ALM imperatives

1. Accelerate time to delivery with **Real-time Planning**
2. Improve quality with **Lifecycle Traceability**
3. Maximize product value with **In-Context Collaboration**
4. Refine predictability with **Development Intelligence**
5. Reduce costs with **Continuous Improvement**

Learn more at: https://jazz.net/library/article/637

1 Overview

2 IBM Rational Collaboration Lifecycle Management

3 Delivering the 5 ALM imperatives

4 Proven results

Business results

*Automated status reporting derived from evolving engineering artifacts can **improve productivity** by*

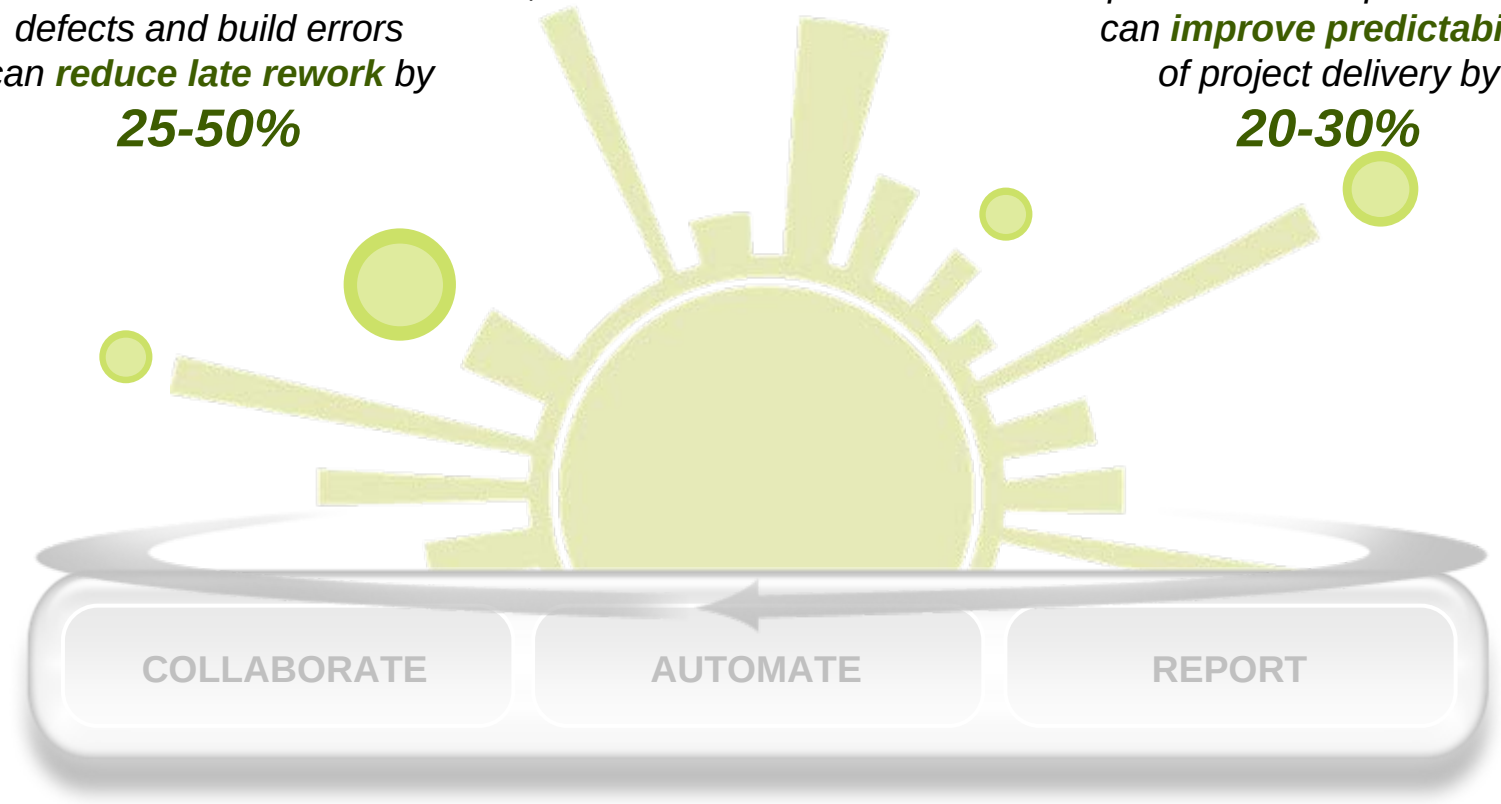
5-10%

*Being able to collaborate on work items, defects and build errors can **reduce late rework** by*

25-50%

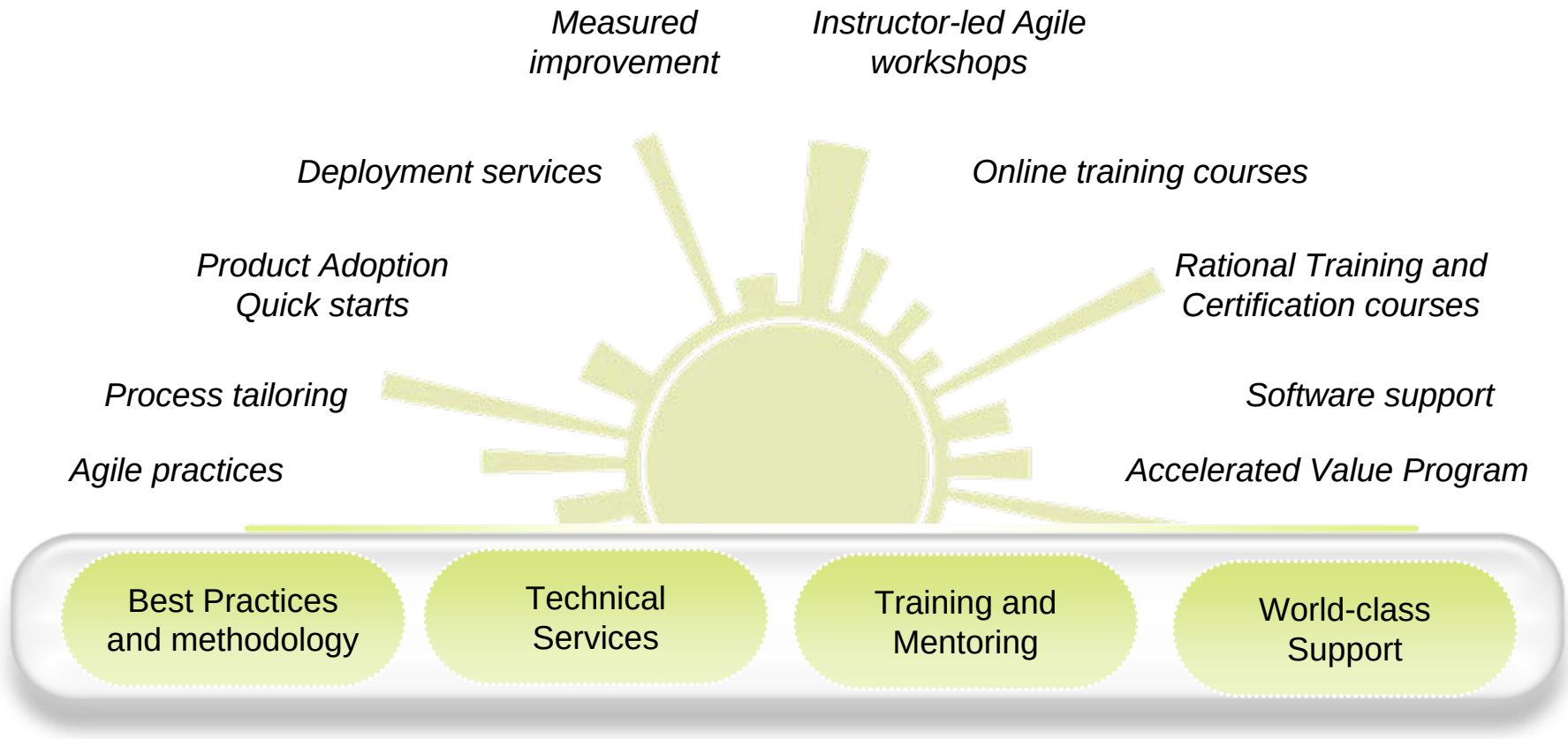
*Best practices in scope management can **improve predictability** of project delivery by*

20-30%



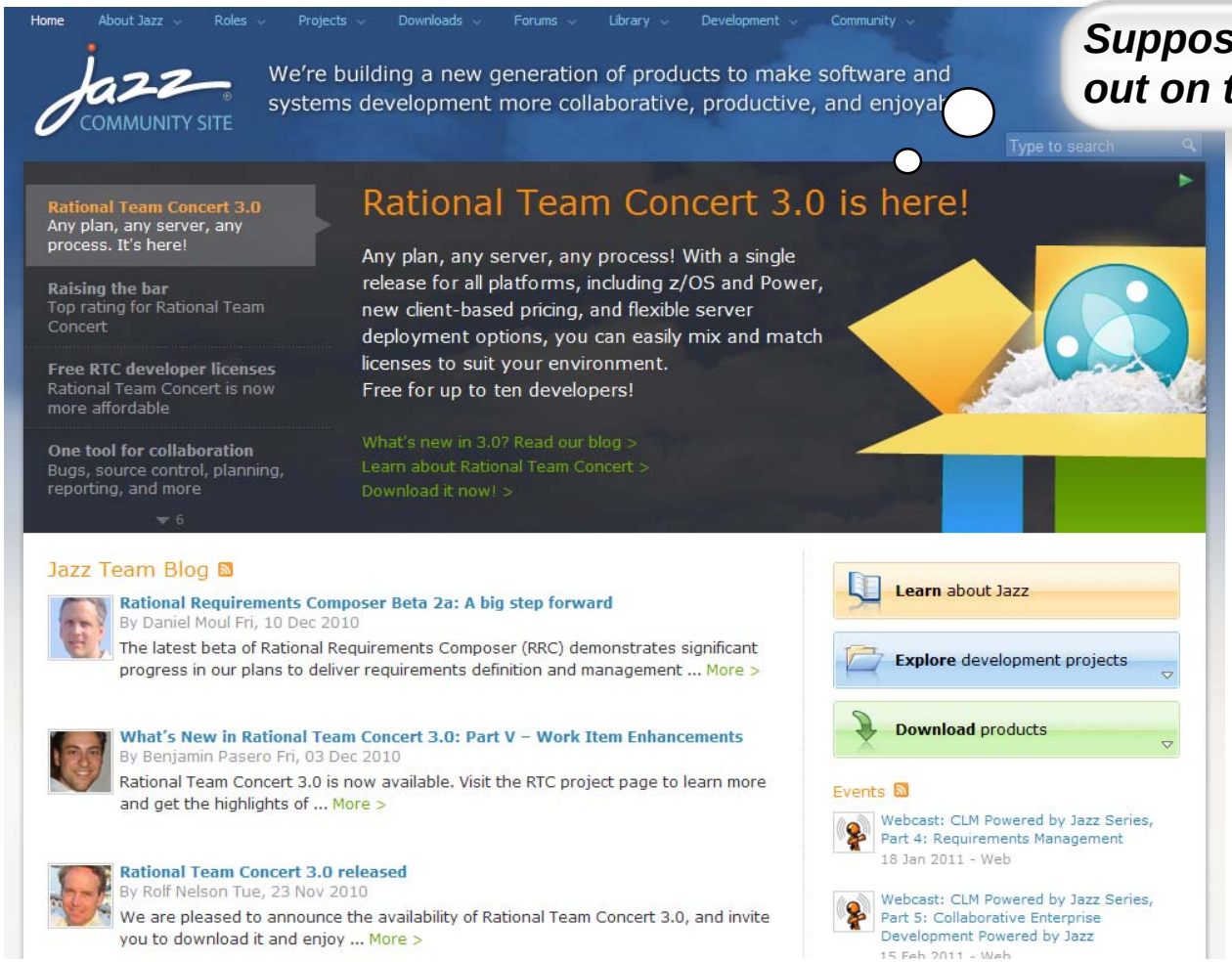
Source: IBM analysis based on services research.

Ensure Success with Rational Collaborative Lifecycle Management



jazz.net

Creating a higher-fidelity connection to our customers



Suppose we did our development out on the Internet?

A transparent software delivery laboratory where you can...

- ✓Get answers and insights directly from engineering
- ✓Communicate with the development team
- ✓Track the progress of builds and milestones
- ✓Get the latest product trials and betas
- ✓Join developers and product managers in discussion groups
- ✓Submit defect and enhancement requests

www.ibm.com/software/rational