# Agility@Scale: Agile Planning and Best Practices with IBM(R) Rational Team Concert(TM)

*Ron Estabillo – Technical Specialist for Rational Software, ASEAN*

Rational. software

© 2009 IBM Corporation

Go to IBM

# Agenda

- **Agile Planning Introduction**
- Agility @ Scale
- A Scrum Planning Episode
- How we use Agile Planning for Developing Rational Team Concert (RTC)

# What is Agile?

- An iterative and incremental (evolutionary) approach performed in a highly collaborative manner with **just the right amount of ceremony** to produce high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.

- **Core principles:**
  - "Fits just right" process
  - Continuous testing and validation
  - Consistent team collaboration
  - Ongoing customer involvement
  - Rapid response to change
  - Frequent delivery of working software

# Agile Methods

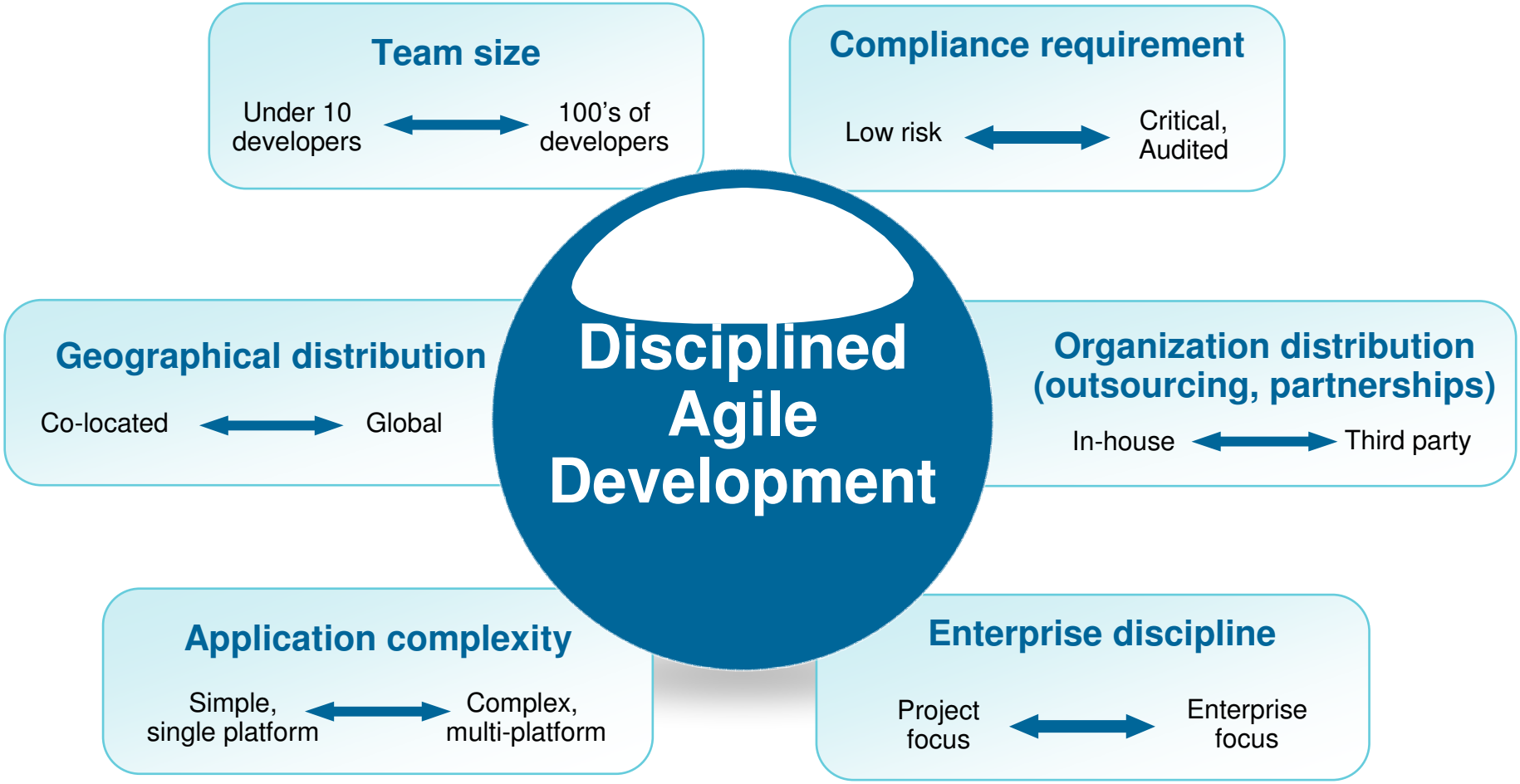
There are several Agile Methods available, however from a bird's eye view they share a common set of tooling relevant properties:

- Short development cycles (1 – 8 weeks)
- Self organizing development teams
  - Joint planning meetings
  - Stand-up meetings
- High personal responsibility of team members
  - Developers "manage" their work
  - Developers / Teams estimate
- Progress tracking
- Ongoing customer involvement
- Use of historical data to improve planning

# Agenda

- Agile Planning Introduction
- **Agility @ Scale**
- A Scrum Planning Episode
- How we use Agile Planning for Developing Rational Team Concert (RTC)

# What is Agility at Scale?



# Agile Process Maturity Model

## 1 Core Agile Development

Focus is on construction

Goal is to develop a high-quality system in an evolutionary, collaborative, and self-organizing manner

Value-driven lifecycle with regular production of working software

## 2 Disciplined Agile Delivery

Extends agile development to address full system lifecycle

Risk and value-driven lifecycle

Self organization within an appropriate governance framework

## 3 Agility at Scale

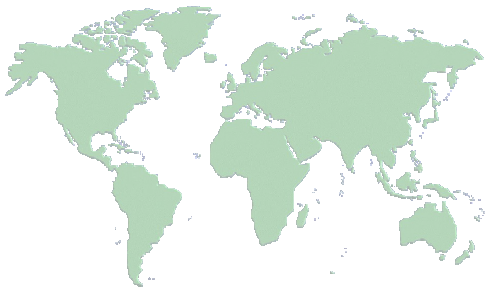
Addresses one or more scaling factors:

- Team size
- Geographical distribution
- Organizational distribution
- Regulatory compliance
- Environmental complexity
- Enterprise discipline

# Rational Team Concert's Agile Planning Support

- RTC's Agile Planning support has the following key characteristics:
  - Process neutral (works with Scrum, OpenUp …), but assumes
    - short (1 – 6 weeks) development cycles
    - supports two level planning: release, iteration
    - supports monitoring releases, iterations
  - No separation between planning/implementing of features and bug fixing
    - Both planning and defect management share a common data model
    - They are supported in the same tool and are highly integrated
    - Plans are in fact a query for work items
  - Supports:
    - Top down planning (product owner, team lead, …) **OR**
    - Bottom Up (team members) **OR**
    - Plan consolidation (Sprint planning meetings, stand-ups, …)

# Agile Best Practice: Additional Roles

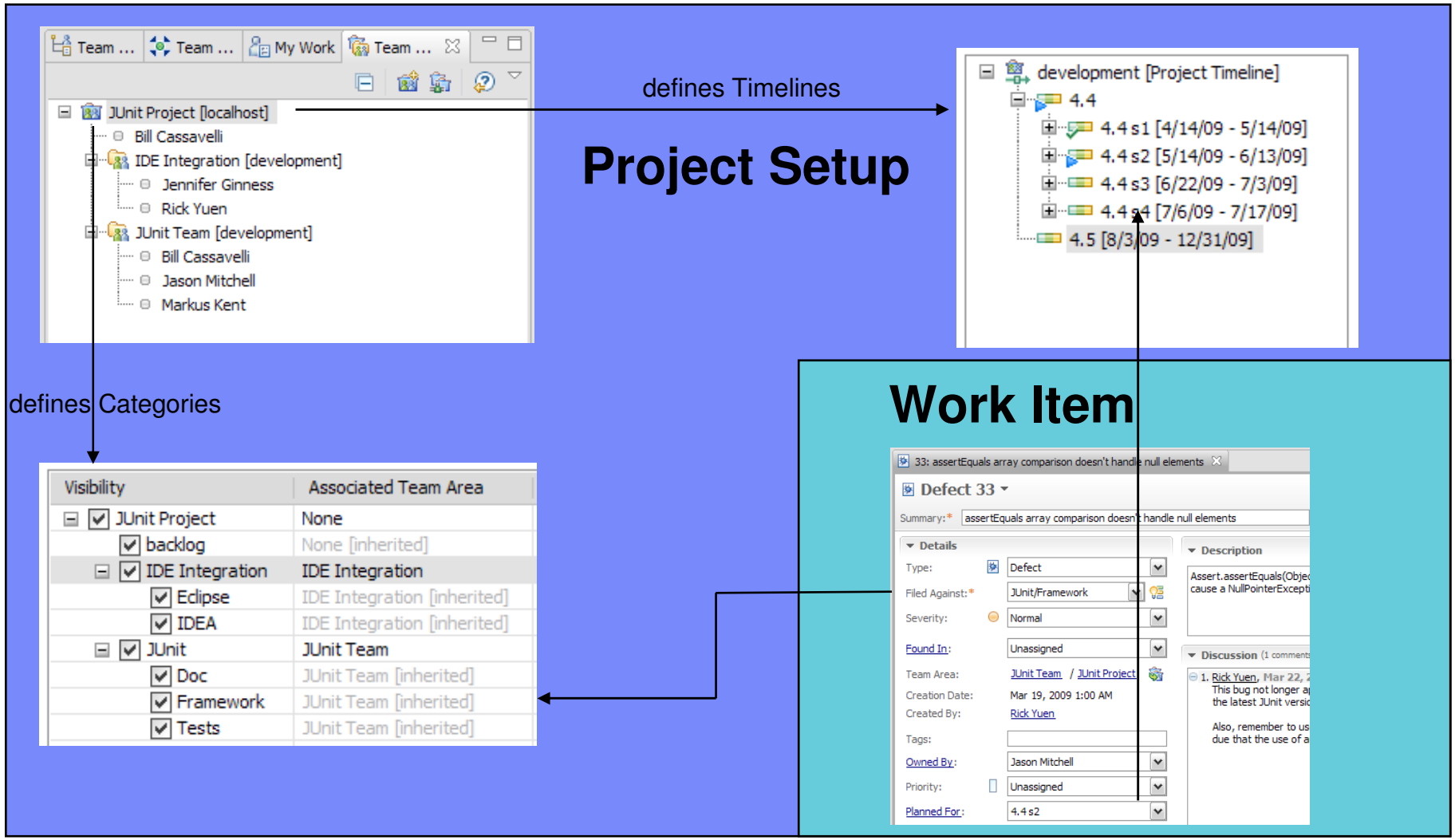


- Ambassadors
  - People who travel between sites to share information
  - Local team rooms should accommodate visitors
  - Short engagements
- Boundary Spanners
  - People specifically focused on enabling communication between sites
- Architecture Owners (Large Teams)
  - Person at each site responsible for the architecture
  - Facilitates and mentors the team through architectural issues
- Chief Architecture Owner (Large Teams)
  - Facilitates the architecture owners
  - Responsible for overall architecture
- Chief Product Owner (Large Teams)
  - Facilitates the product owners
  - Responsible for overall requirements

# Agenda

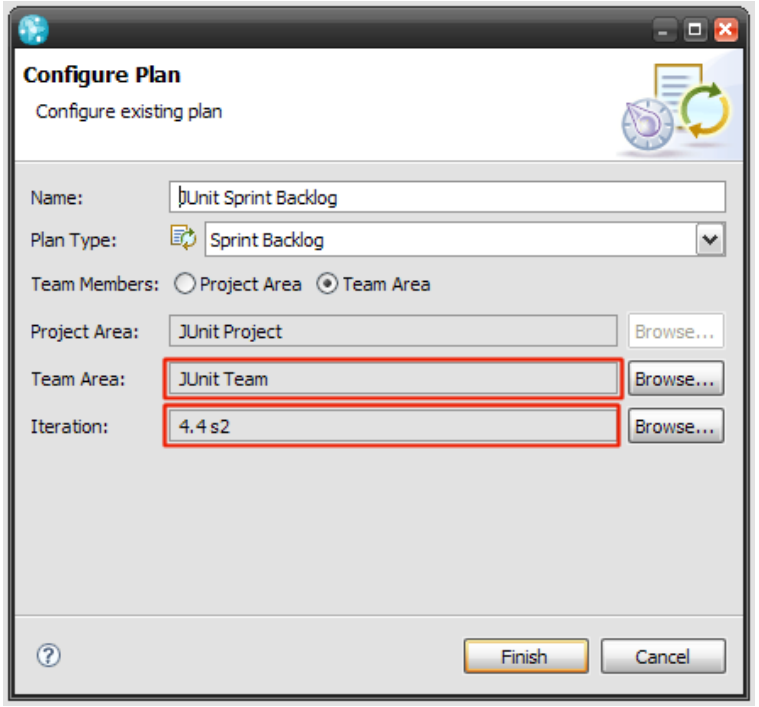
- Agile Planning Introduction
- Agility @ Scale
- **A Scrum Planning Episode**
- How we use Agile Planning for Developing RTC

# Project Areas, Timelines, Iterations and Categories



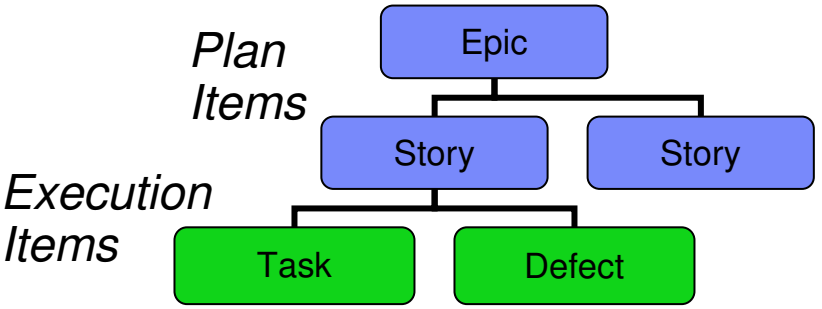
# Concept: Plan

- Plans are owned by a team or project and are associate with an iteration
- Plan contents is derived from this:
  - All work items which have a Filed Against value set to a category owned by the plan's team or project and are planned for the plan's iteration.
- They have a plan type, defining the "onion" type of a plan

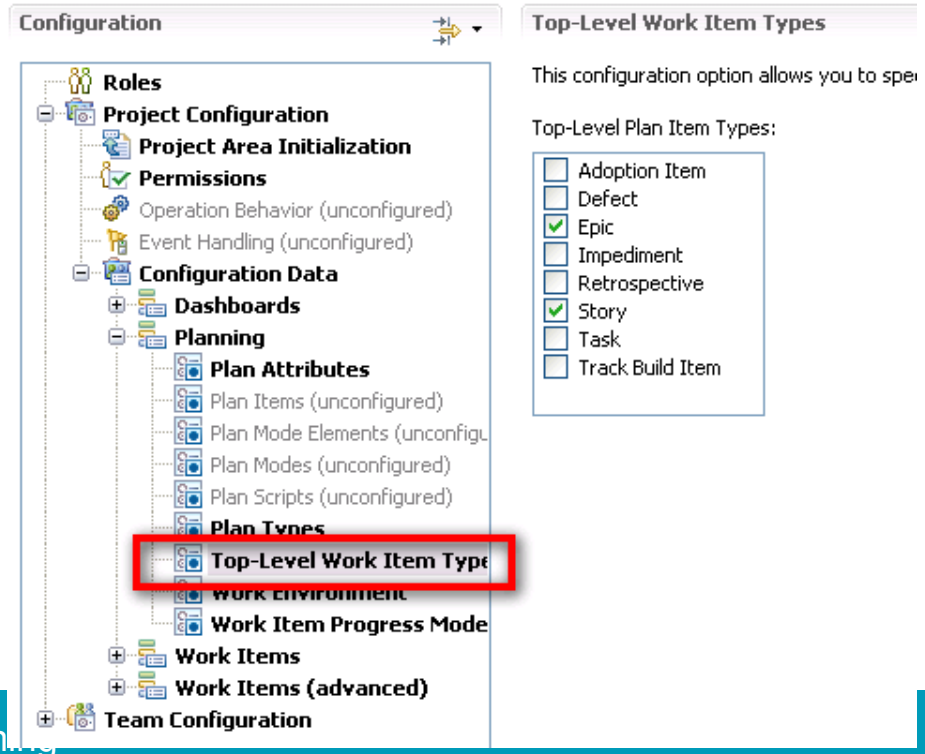


# Concept: Plan and Execution items

- Work items have two flavors:
  - Execution Items: work items which have work assigned. In Scrum these are task, defects, …
  - Plan Items: work items that are relevant for the planning. In Scrum these are Epics and Stories




- RTC allows to customize which work item types are plan items and which types are execution items.



# Concept: Plan Types

## Release Planning:

- contains planning work items for
  - The plan's iteration and all its child iterations
  - The plan's team area or project area and all its child team areas
- Support for teams of teams

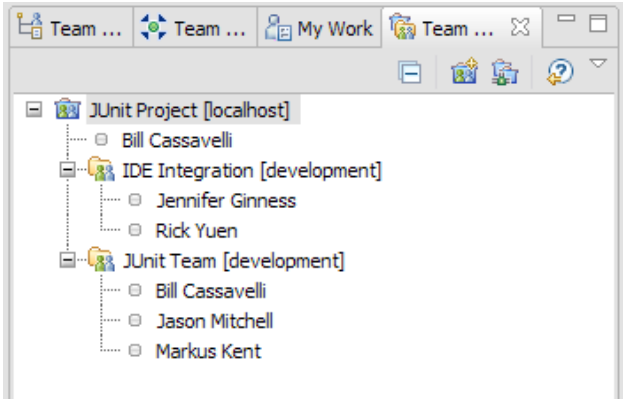
## Iteration Planning:

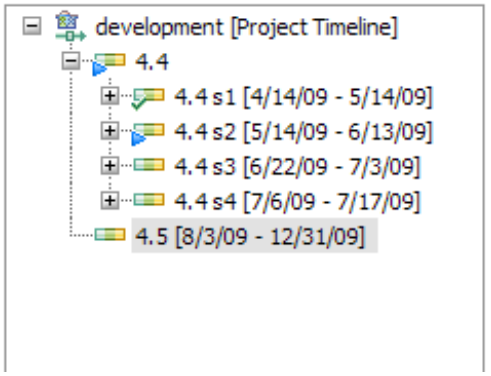
- contains execution and planning work items for
  - The plan's iteration
  - The plan's team or project area

## Daily planning

- My Work View

# Plan Types - continued



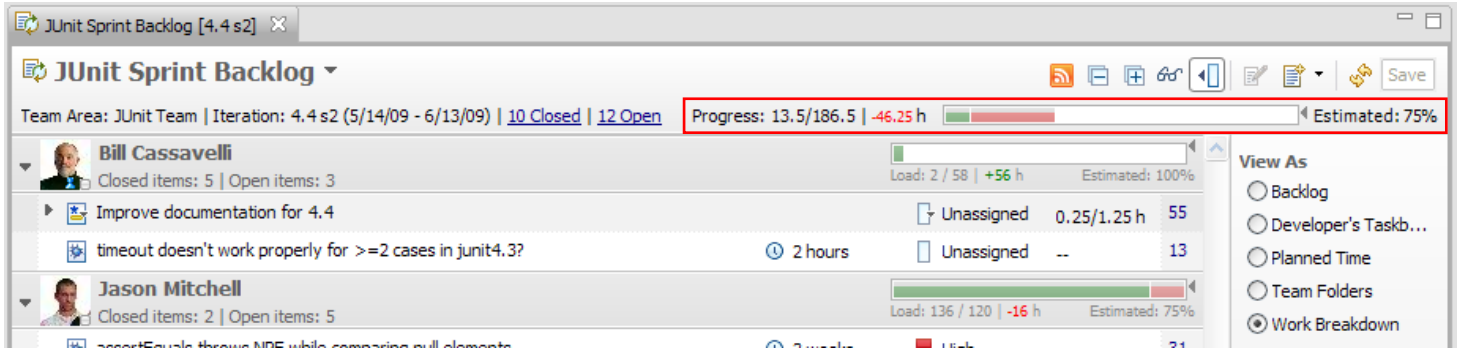

- Release plans
  - **JUnit Project** / **4.4** (***for projects***): shows all plan items associated with the JUnit Project, IDE Integration team or JUnit team and planned for for 4.4, 4.4s1, 4.4s2, 4.4s3 or 4.4s4
  - **IDE Integration team** / **4.4** (***for teams***): shows all plan items associated with the IDE Integration team and planned for 4.4, 4.4s1, 4.4s2, 4.4s3 or 4.4s4. In addition is shows all execution items associated with the IDE integration team and planned for 4.4
- Iteration plan for JUnit Team / 4.4s2
  - Shows all work items associated with the JUnit team and planned for 4.4s2

# Concept: Estimation and Sizing

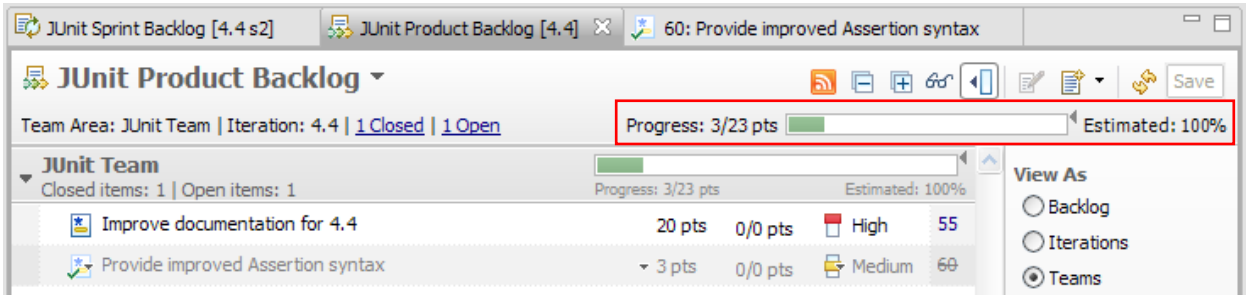
- Agile planning separates **estimating the size** from **estimating the duration**
- The **size** is often estimated in relative values. Scrum uses **Story Points**
- The **effort** is typically estimated in **Ideal Hours/Days**
- **Velocity** measures a teams rate of progress
  - Completing 2 Stories of 5 story points in one iteration gives a velocity of 40
  - Good guess is to assume the same velocity for the next iteration
  - Team velocity is what matters

# Estimation and Sizing

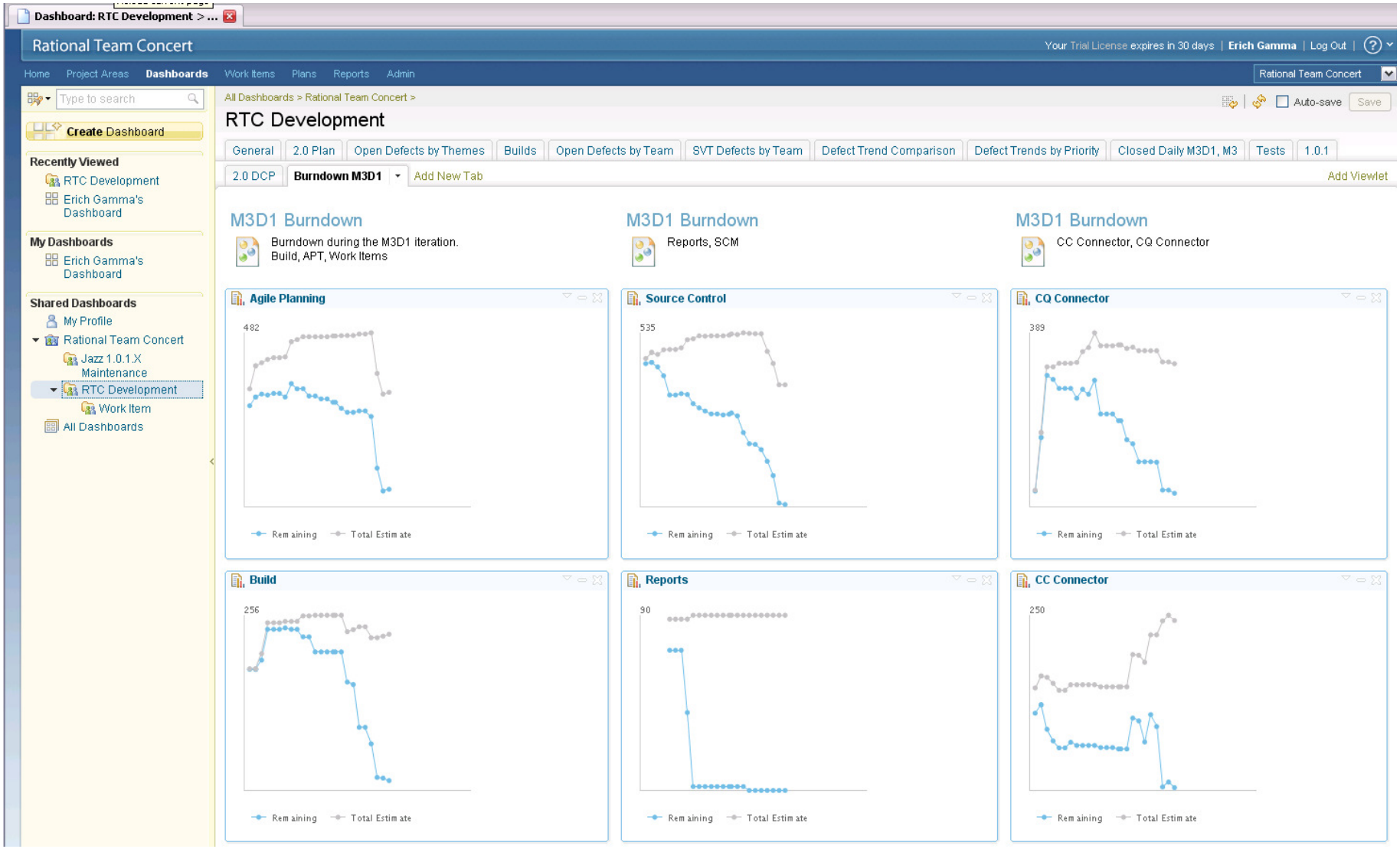
- Effort: Execution items are estimated in ideal hours.




- Size: Plan items are estimated using a configurable numerical sizing attribute
  - Scrum uses Story points. Other sizing attributes could be: lines of code, function points, …

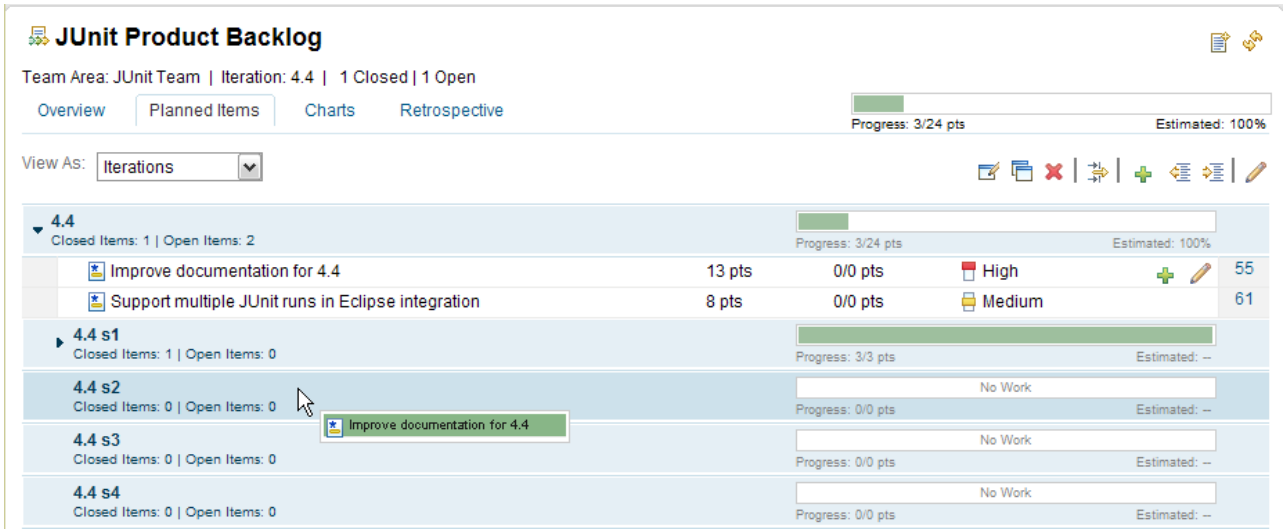


# Concept: Effort Burn Down



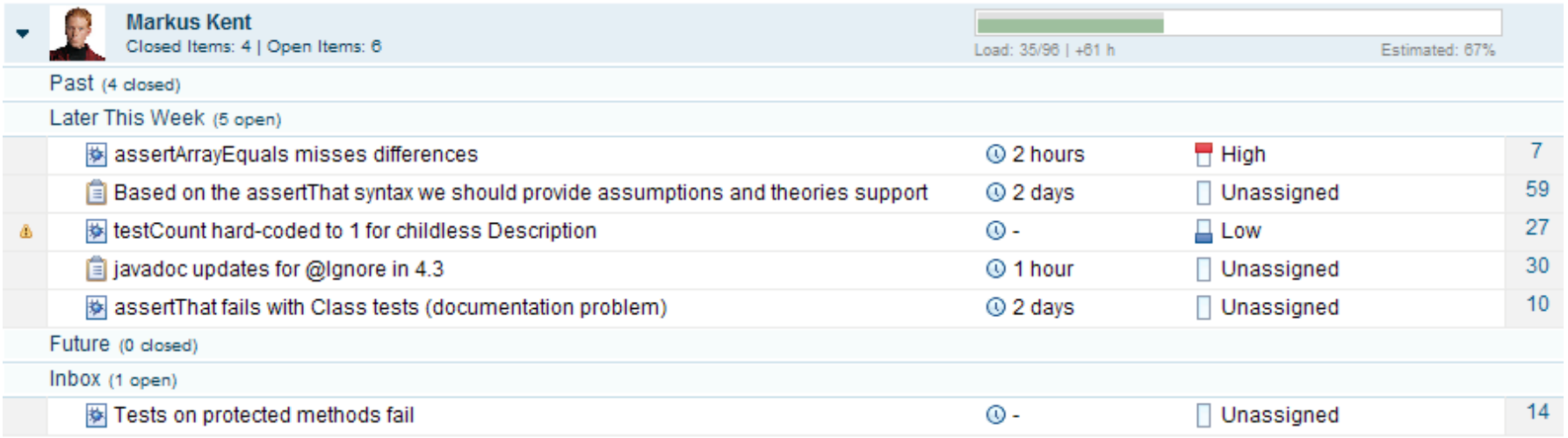
# Concept: Release Planning

- Configure the project areas time line
- Create a plan of type release plan / product backlog
- Take velocity reports or other historical data to find out how many story points a team can handle per iteration.
- Assign the plan items to the iterations, based on the backlog order and the team's velocity.



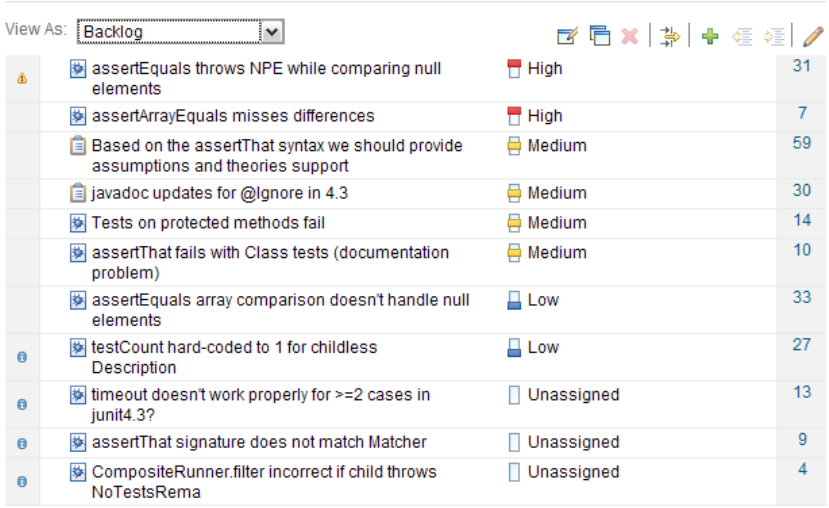
# Concept: Iteration Planning

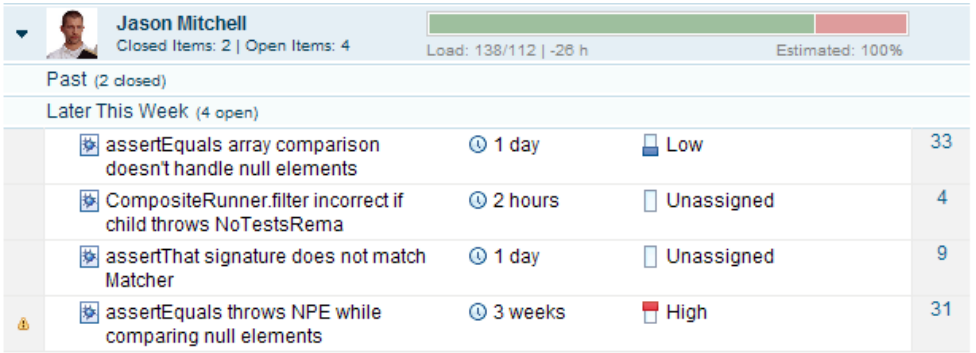
- RTC provides different view modes
  - Breaking plan items into tasks
  - Tracking progress by owner during an iteration
  - …
- Use **Load Bars** for capacity planning
- Use **Progress Bars** to track progress



# Concept: Backlog and Planned Time

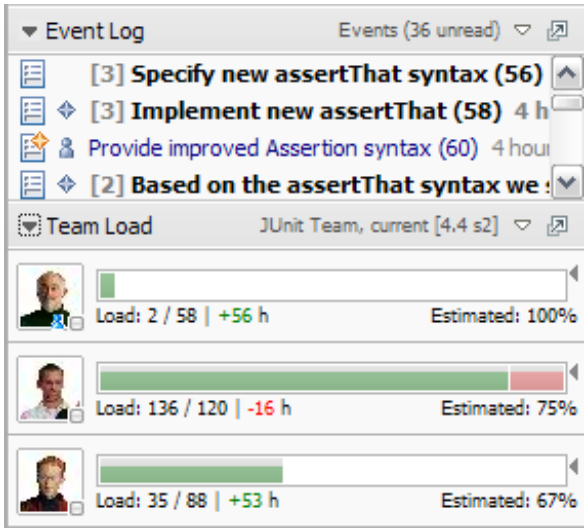
- Backlog
  - Sorts the work items in backlog order. Rearranging work items in this sort mode will change its ranking on the backlog
- Planned Time
  - Sorts the work items in the order in which the owner of the work item intends to work on them. The owner defines this sequence typically in the My Work View.

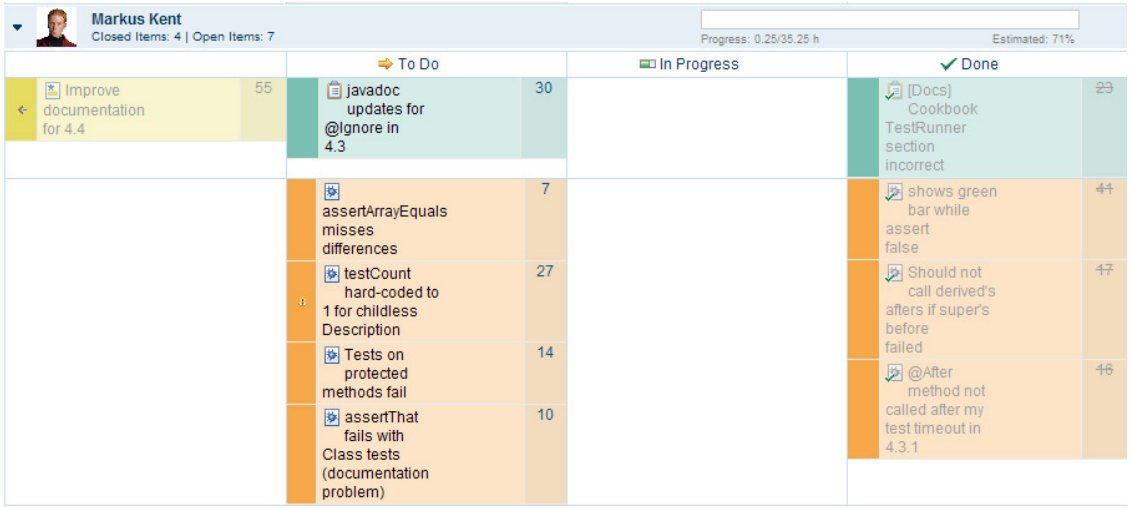


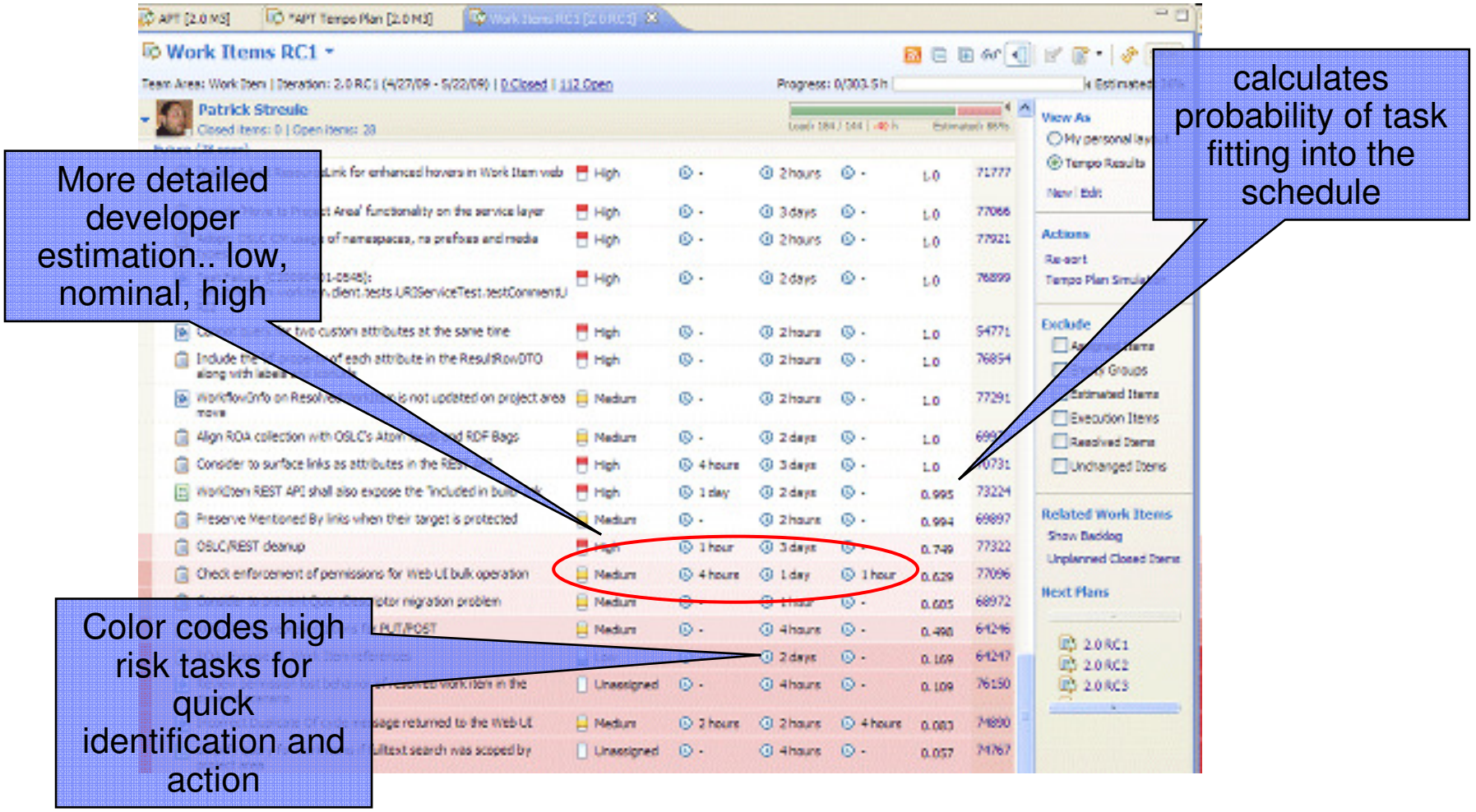
# Concept: Daily Planning

- Triage important defects onto the Iteration Plan
  - Team Central, Team Load and Iteration Plan
- Run daily stand ups
  - Developer Taskboard to see what people are working on
  - Planned Time mode to check overall schedule



# Concept: Plan Risk Assessment

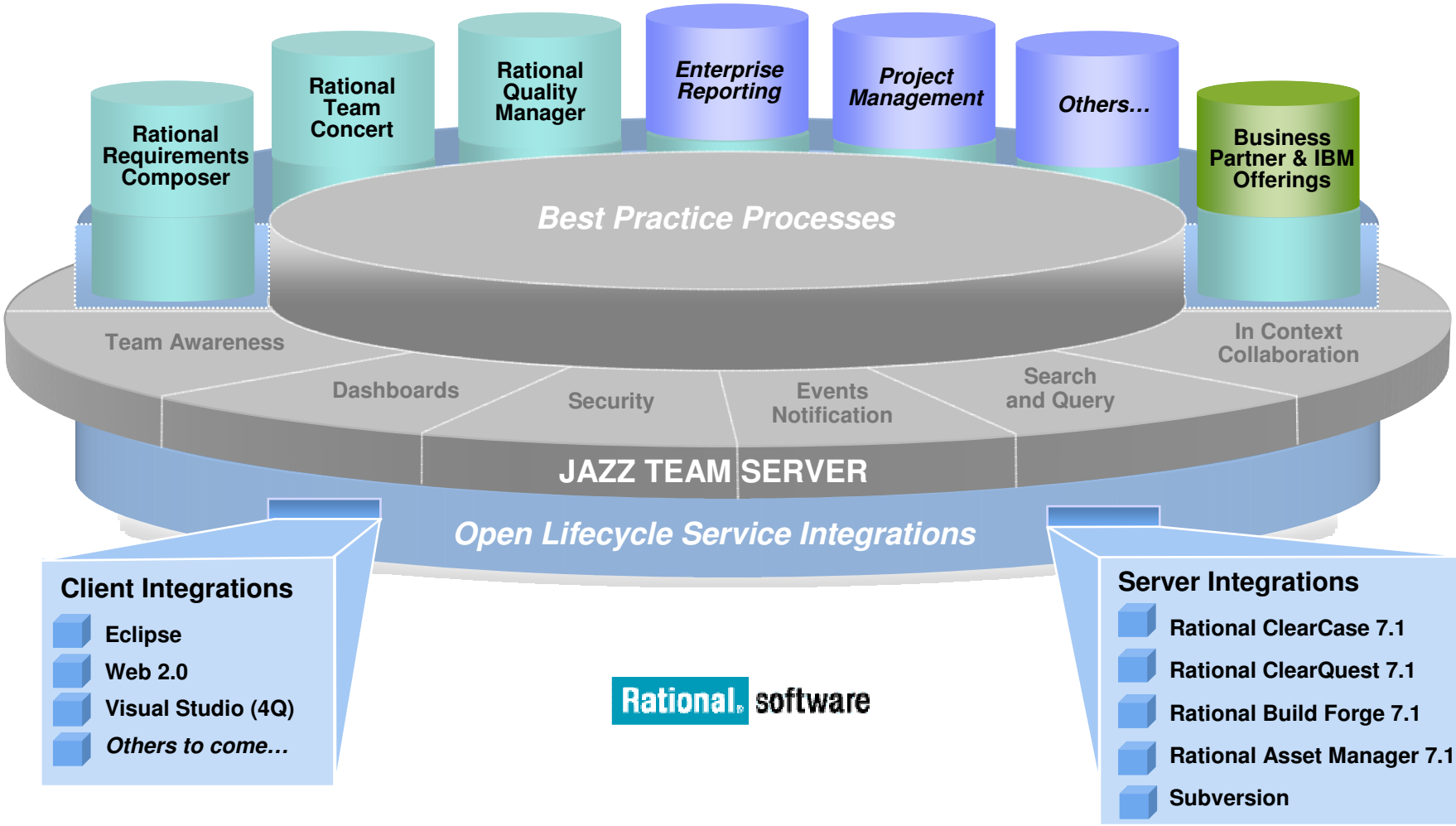


DEMO

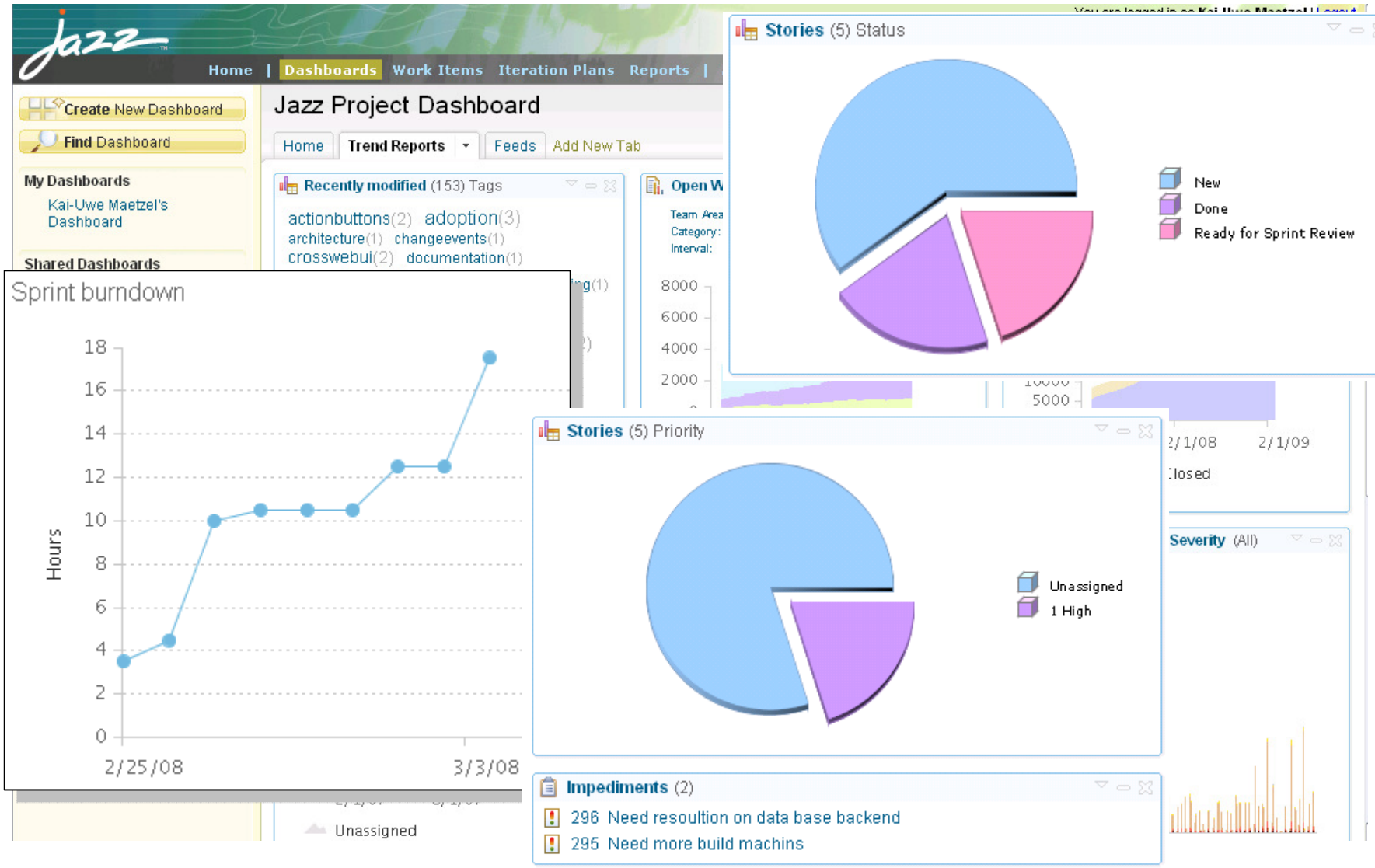
# Agenda

- Agile Planning Introduction
- A Scrum Planning Episode
- **How we use Agile Planning for Developing RTC**
- What's new in 2.0

# Rational Team Concert Platform Architecture

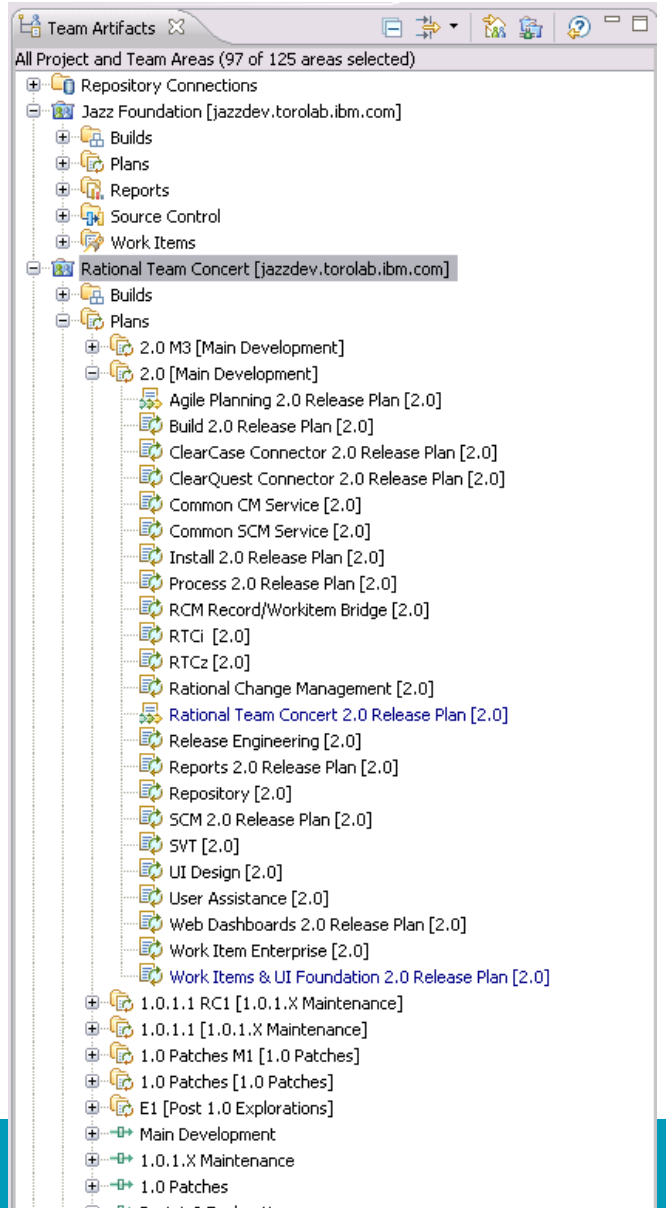


# Continuous Project Monitoring

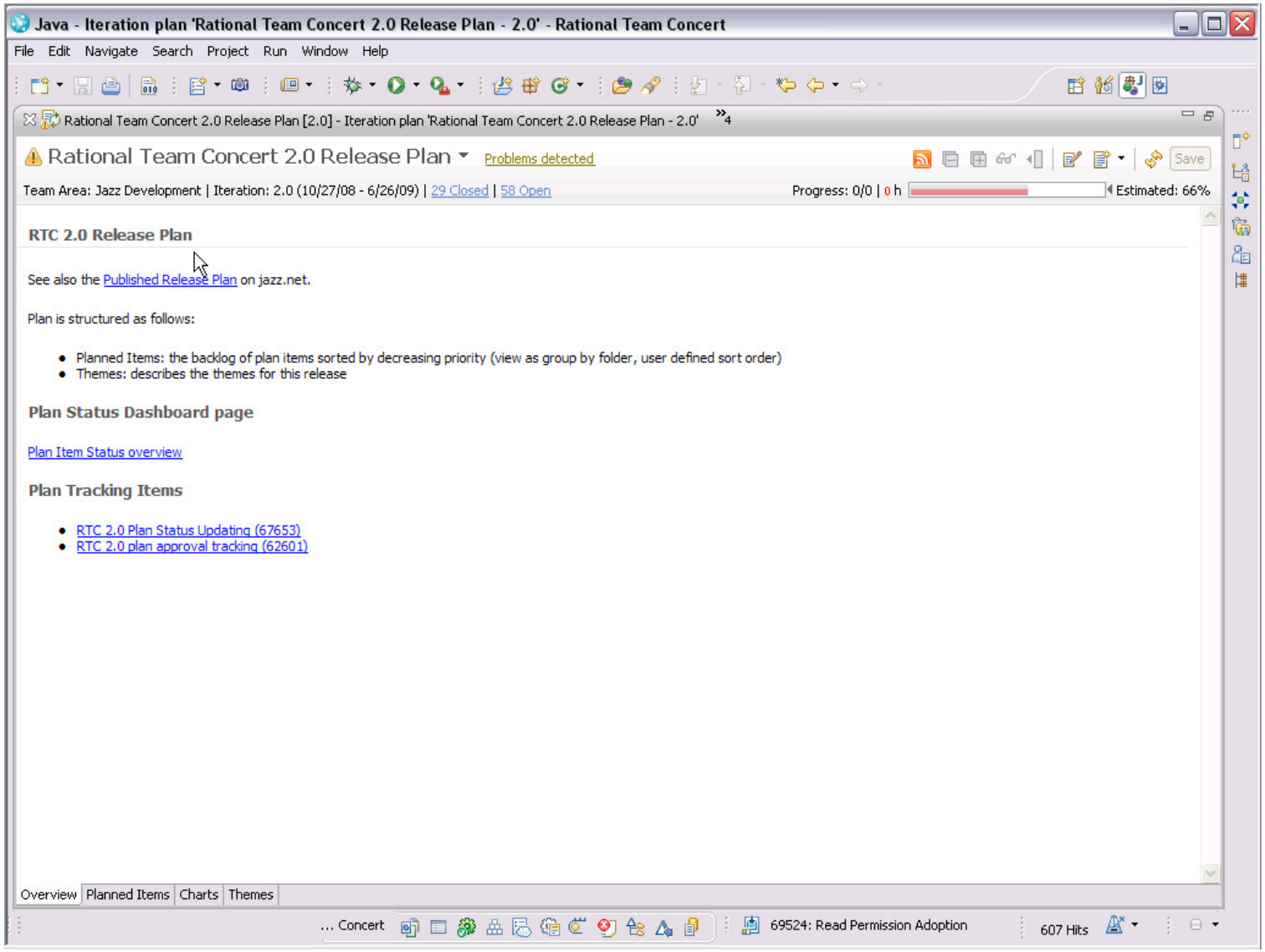


# Using RTC to Plan and Deliver **RTC 2.0 - Overview**

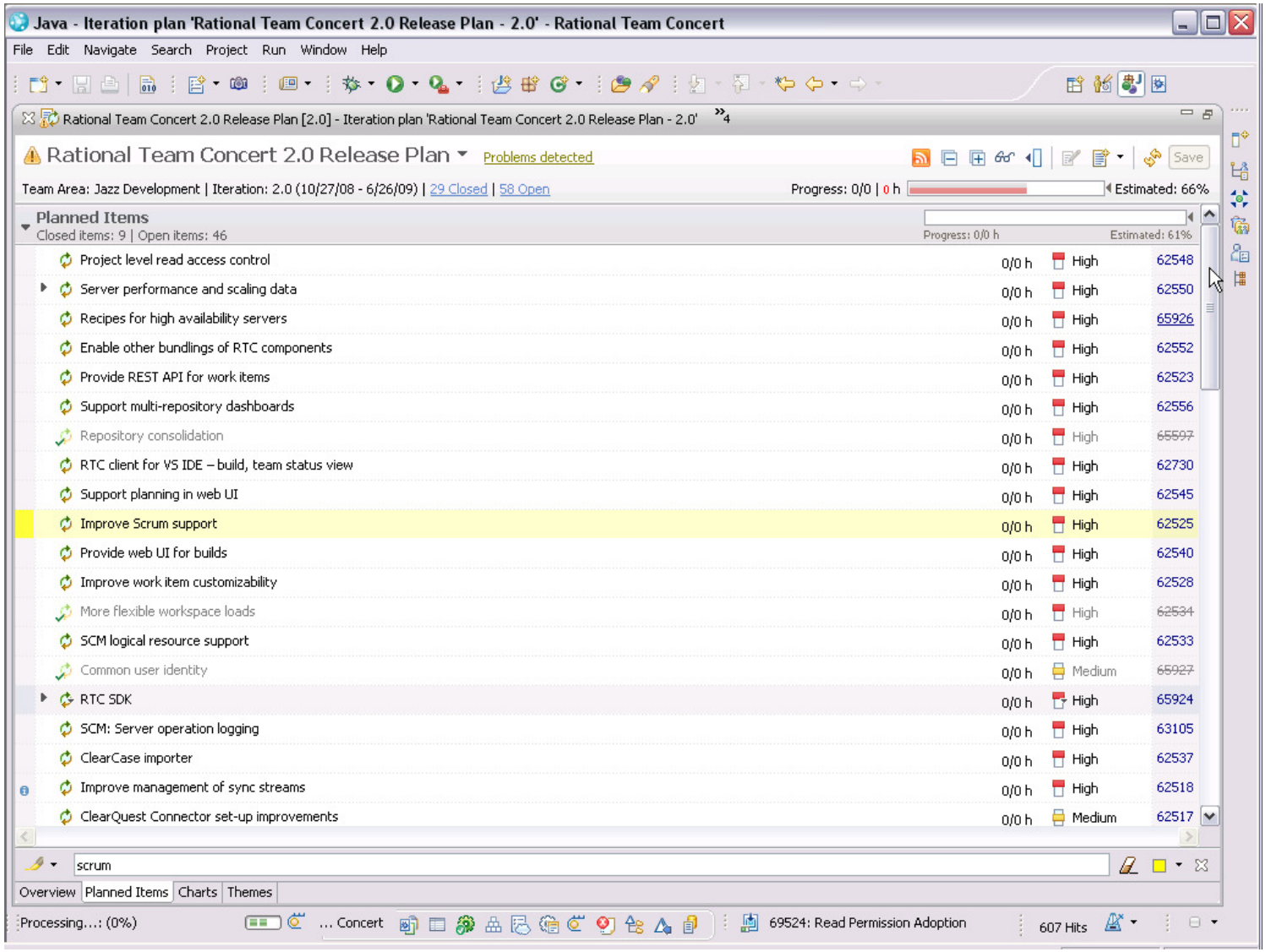
- Release Plan
  - Define themes
  - Define plan items
  - Plan Item Work item type
- Tracking a Release
  - Effort less status tracking
  - Progress
- Iteration Planning (Scrum of Scrum)
  - Tracking cross team efforts
  - Tracking burn down across teams
  - Retrospectives
- Tracking Dependencies across teams



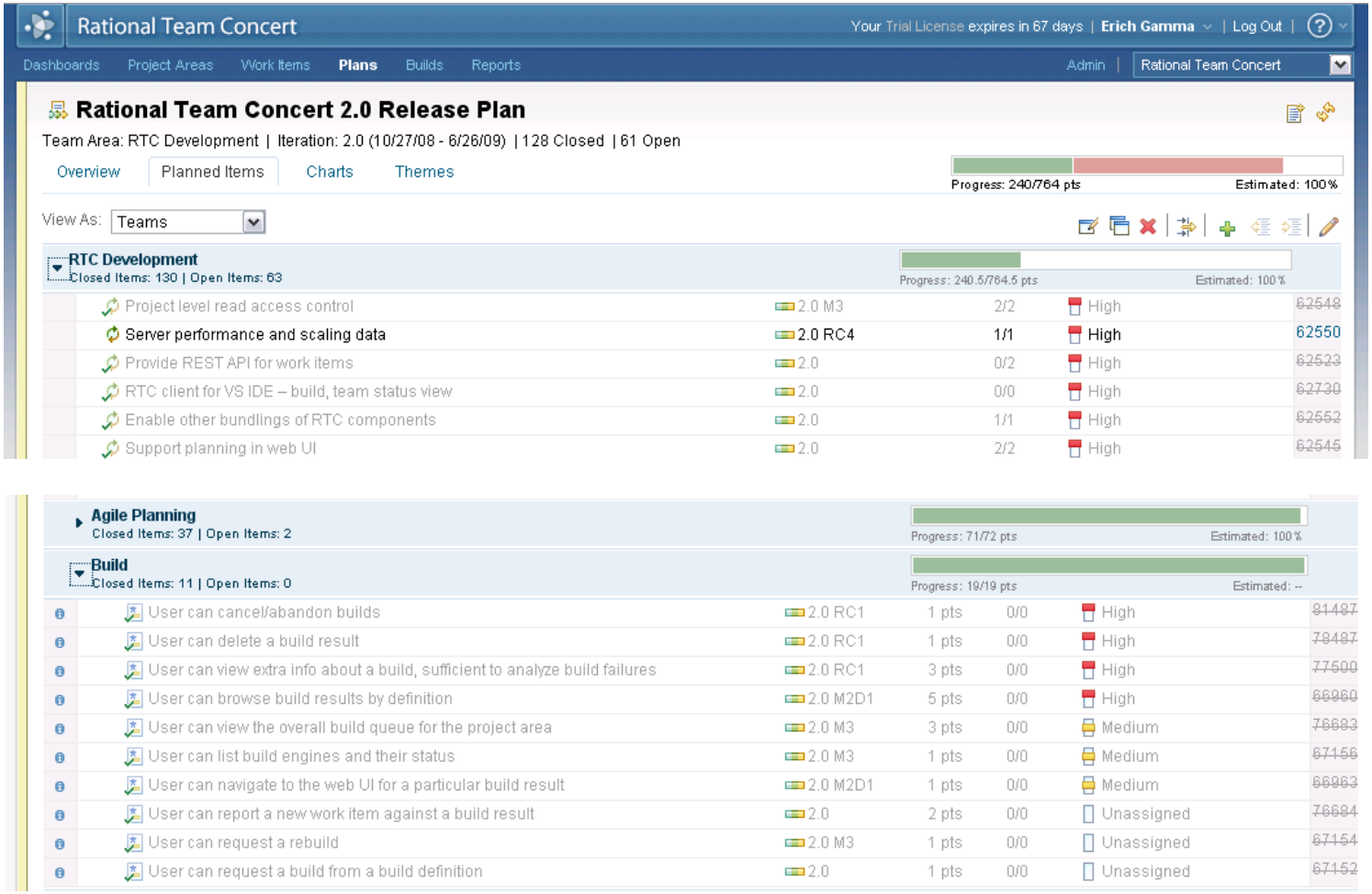
# Release Plan Overiew



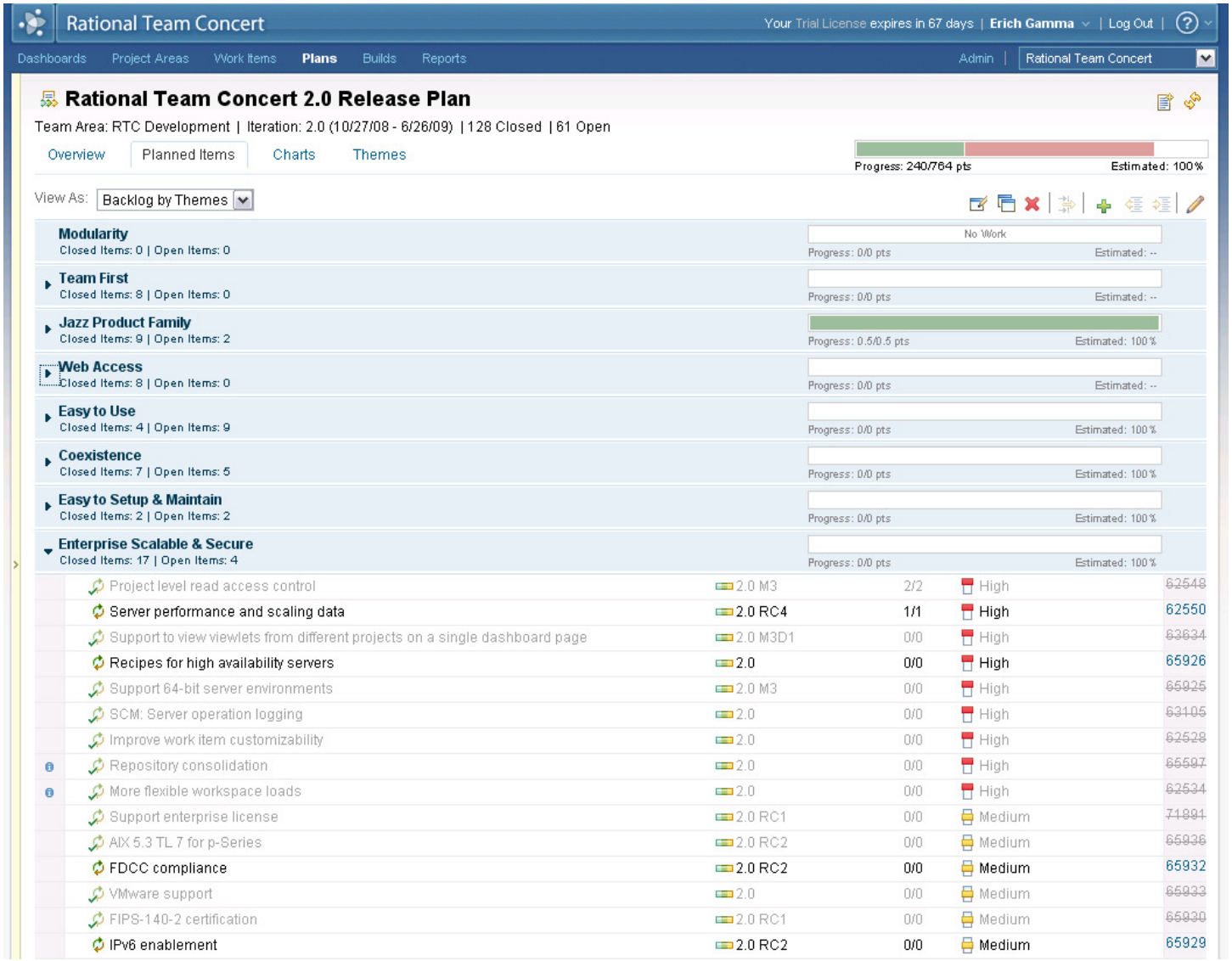
# Release Backlog



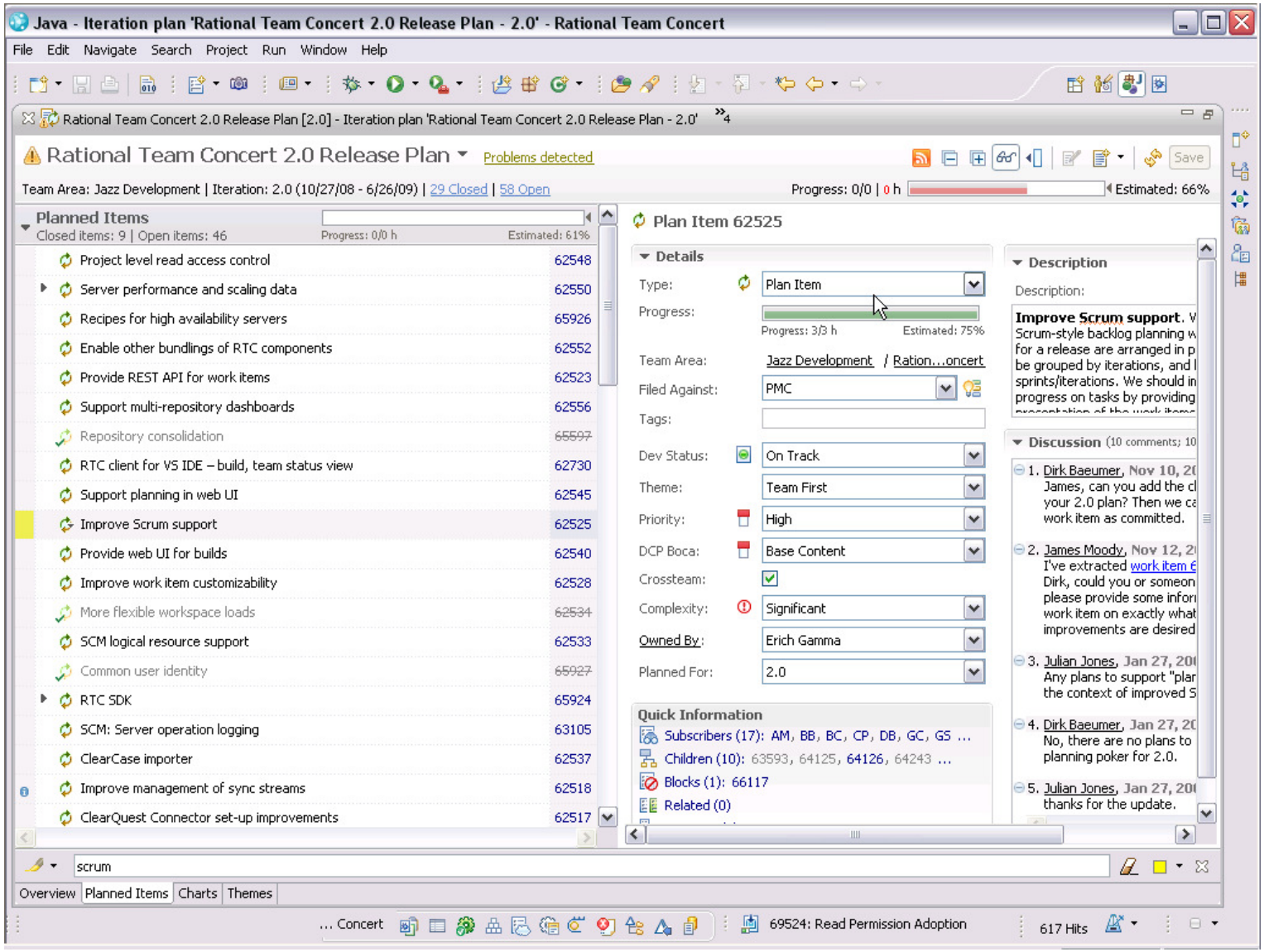
# Project Release Plan: Team of Team Release Plan



# Backlog Grouped by Themes

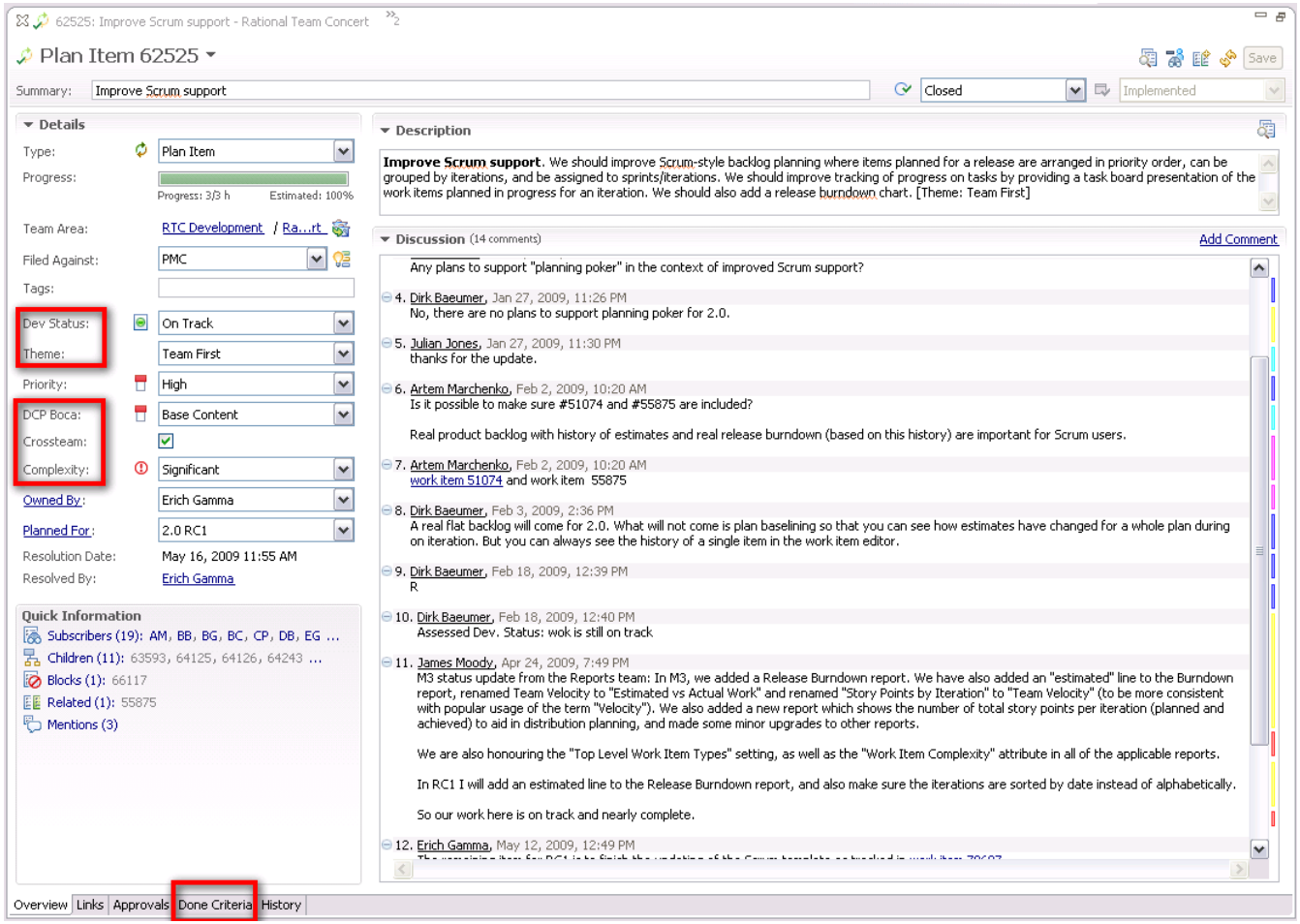


# Backlog with Plan Item Preview



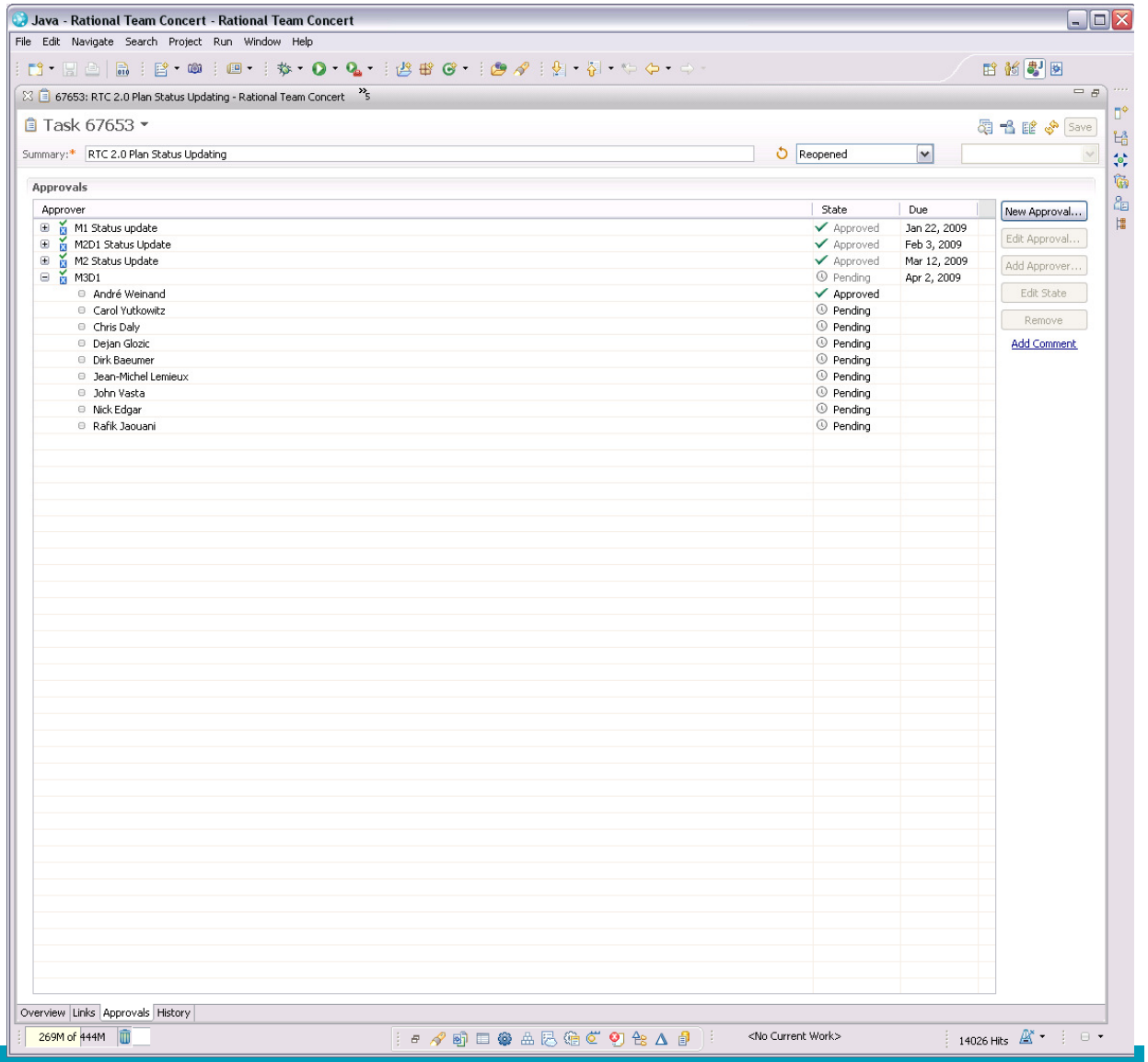
# Plan Item

- Custom Attributes
  - Complexity
  - Development status:
    - on track
    - at risk
    - behind
  - Done criteria
- Workflow
  - Proposed
  - Committed
  - Done

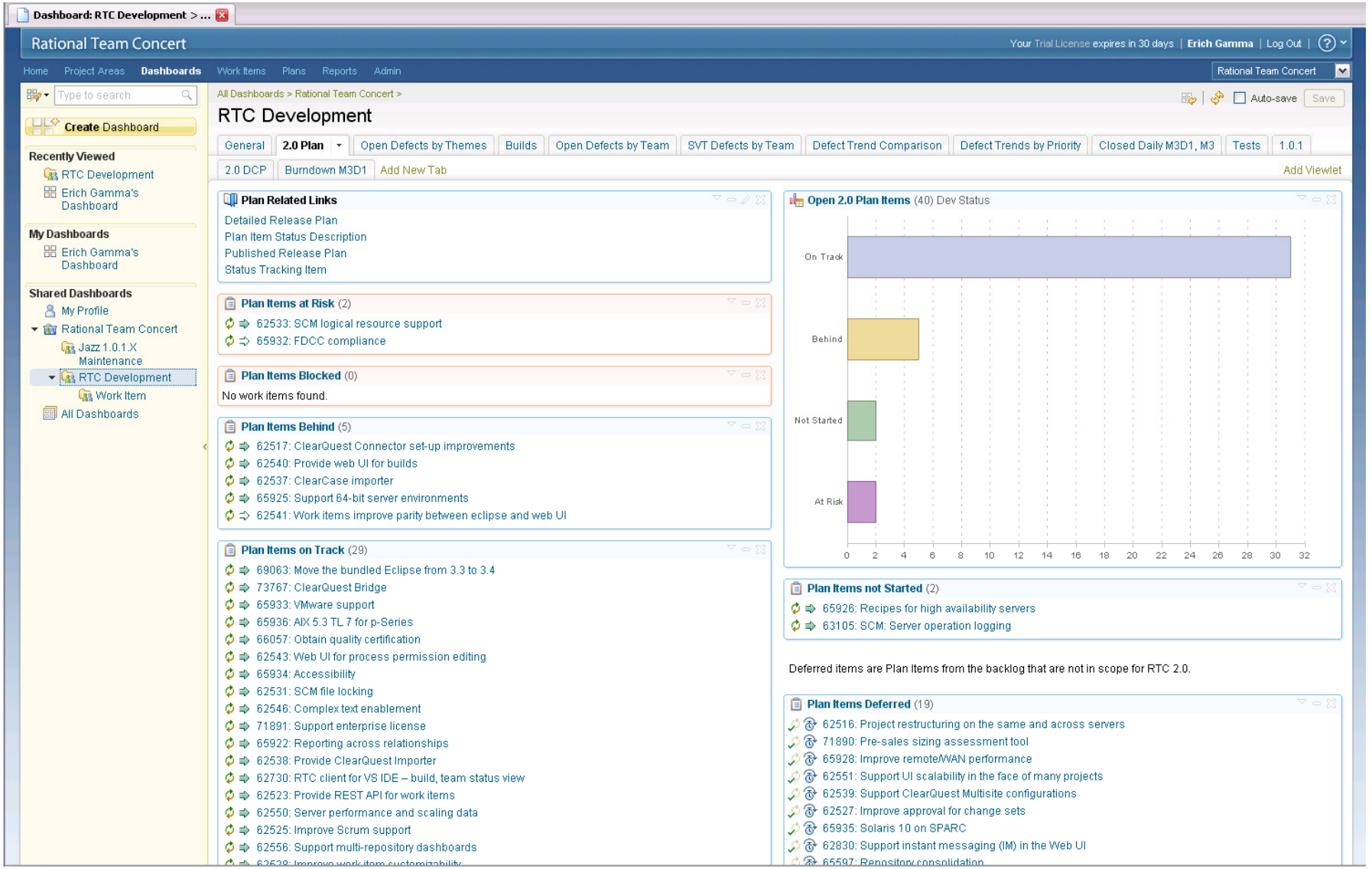


# Release Plan Status Tracking

- Painless status tracking
- Status updating is tracked in a task with approvals
- Status updated at end of each iteration

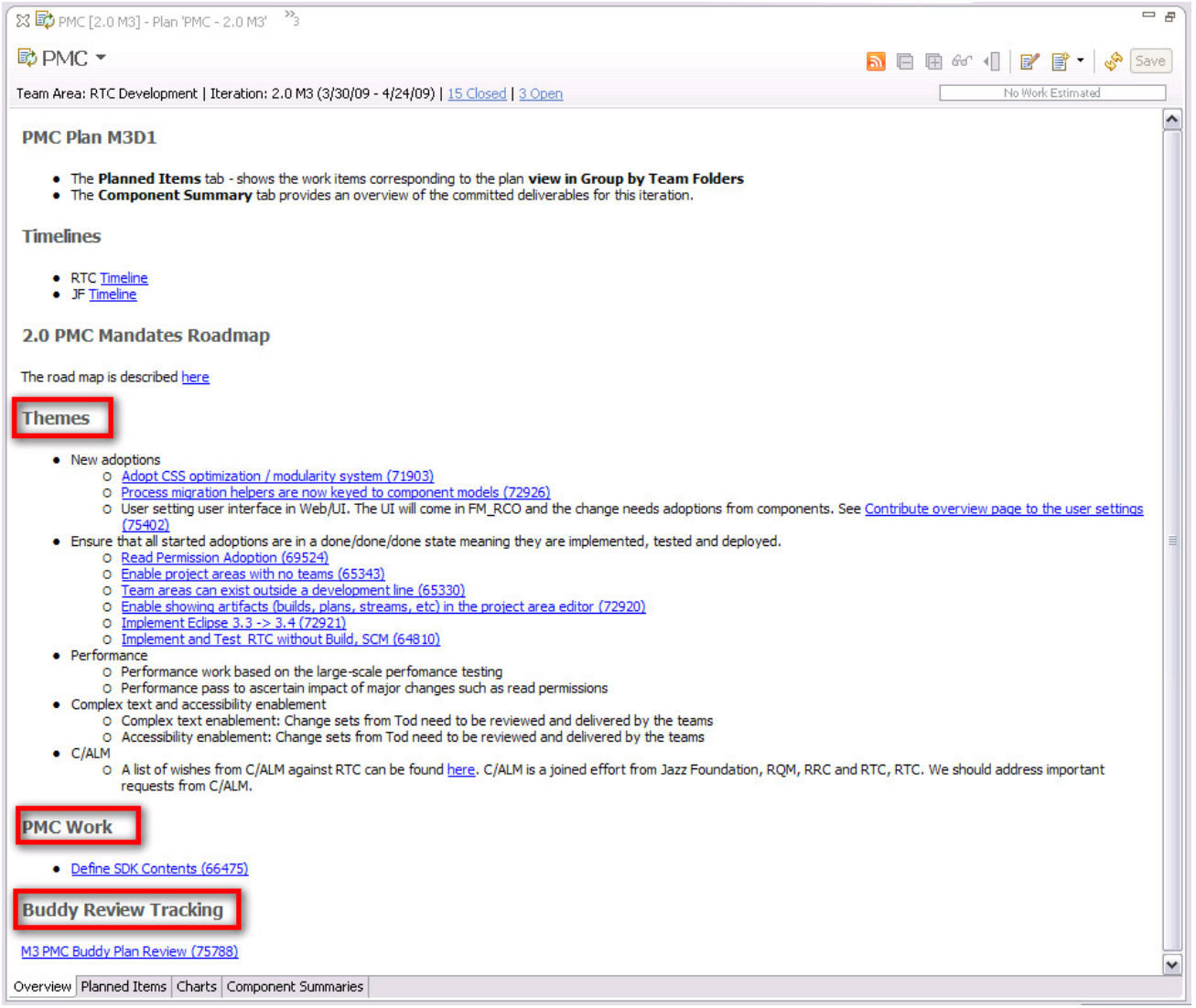


# Dashboard with Plan Item Status

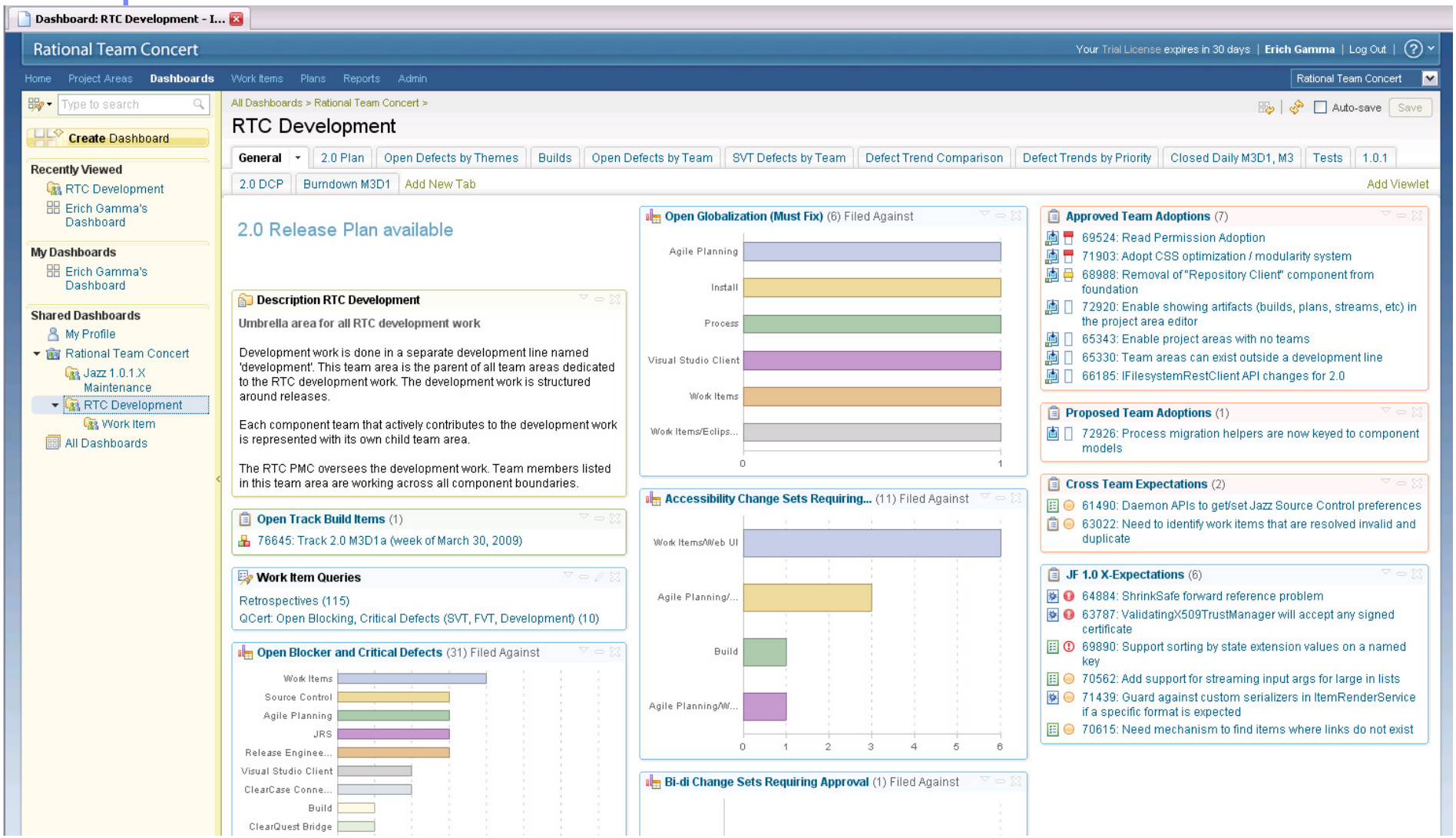


# Project Management Iteration Plan

- Defines mandates from the project management team
  - Themes
- Buddy Review Tracking



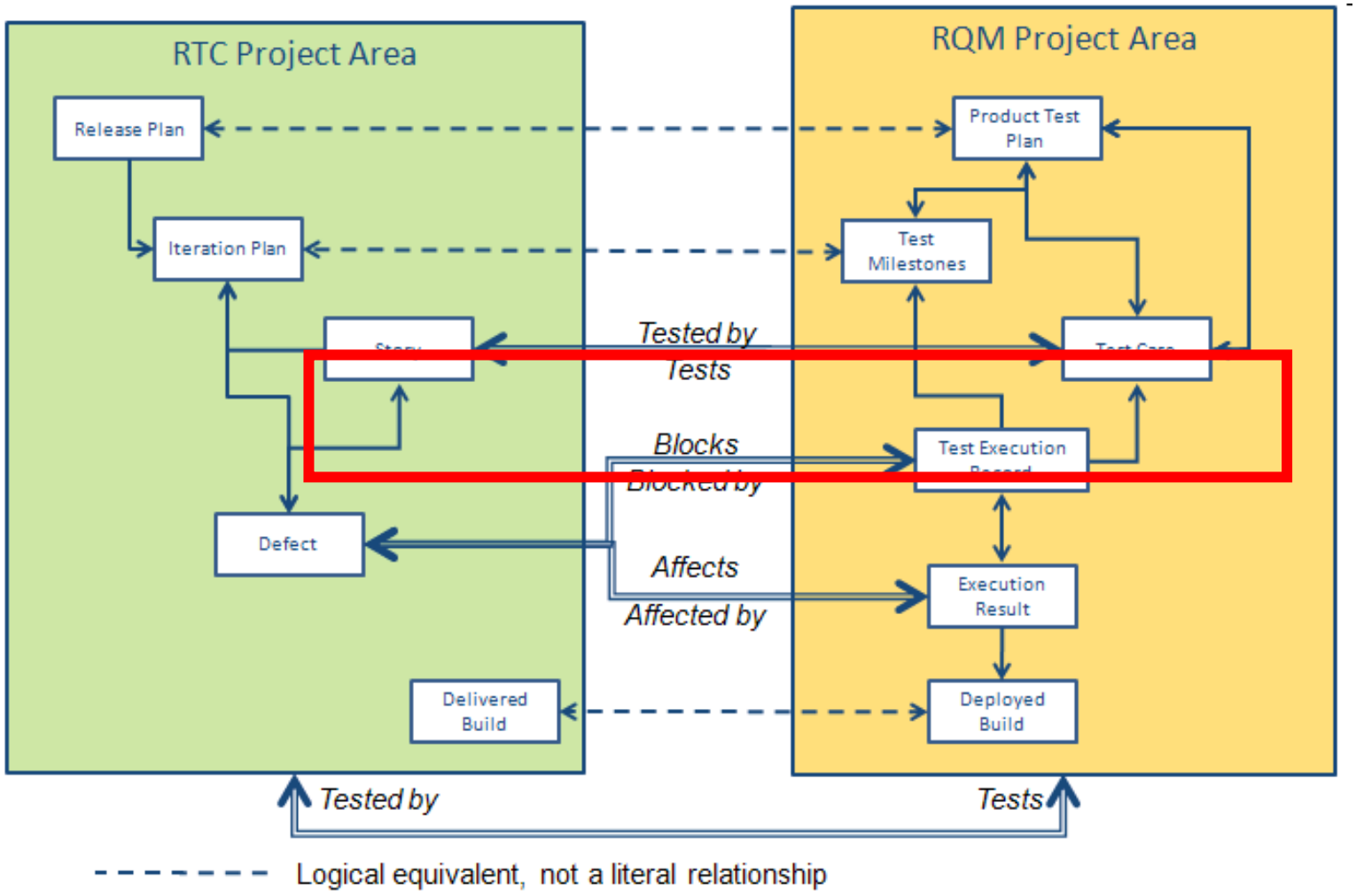
# Adoption Items on the Dashboard



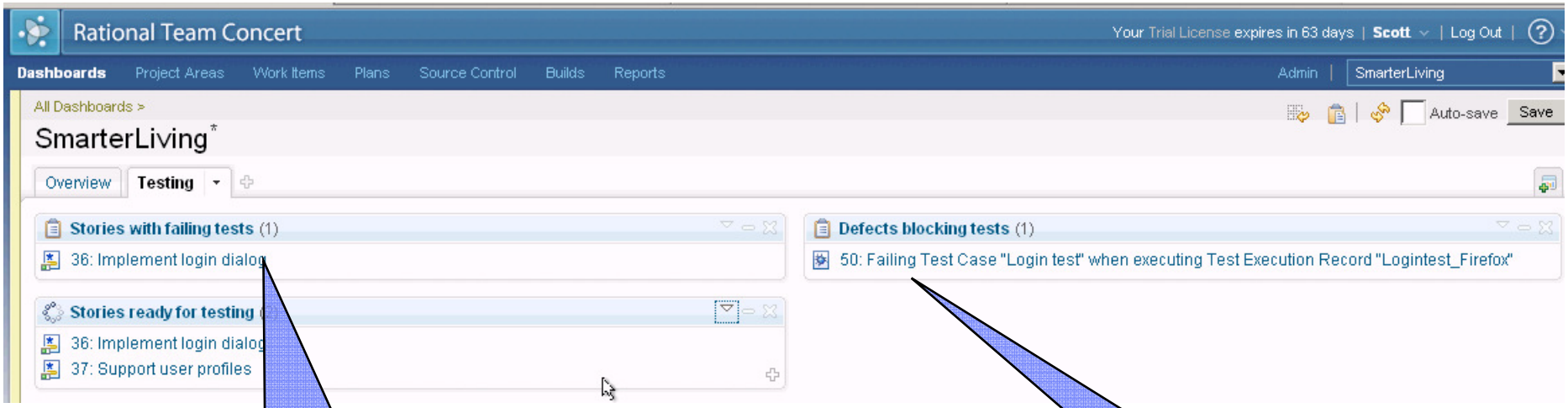
# More support for Agile at Scale

- RTC Build
  - personal builds
  - build snapshots
- Buildforge Integration
- Jazz SCM
  - isolation/integration using stream flow hierarchies => show our flow structure
  - private versions for code exploration
  - suspend resume for task isolation
- Dashboards
  - for transparency

# Agile Planning & Collaborative AL M



# C/ALM Link Queries




Stories with failed tests

Defects blocking test execution

# Summary

- RTC 2.0 covers the planning onion
- Much improved scrum and scrum of scrum support
- Planning for everybody in eclipse and web

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.

# Recommended Resources

- www.ibm.com/rational/agile/
- www.ibm.com/developerworks/
- www.ibm.com/software/info/developer/solutions/gdd/
- www.ibm.com/developerworks/blogs/page/ambler

# References


- Mike Cohn, Agile Estimating and Planning
- Webcast: Using Rational Team Concert in a Globally Distributed Team
  - https://jazz.net/blog/index.php/2009/01/06/webcast-using-rational-team-concert-in-a-globally-distribute-team/