Hands-On-Developer-Workshop

Welcome to our short live tour into the **Rational Jazz Platform**. With the time given it is really only possible to get a glimpse into how the Jazz Platform and the corresponding tools can help with development. The Labs will focus on core capabilities of IBM **Rational Team Concert**. We begin with a defect for a failing test detected with **Rational Quality Manager**. During the lab the integration capabilities of the Jazz Platform will be only briefly demonstrated, in an optional lab, by accessing the data in **Rational Quality Manager**. The example also provides integration with requirements management data in **Rational Requirements Composer** but does not explicitly show it.

Background and starting point

You will be part of the **Squawk** development project. The project develops the so-called Squawker application using the Jazz-based tools Requirements Composer, Team Concert, and Quality Manager. The existing Squawker application's purpose is to demonstrate the sounds "things in this world" for example animals, people and machines produce.

The Squawk project consists of several teams, each one responsible for separate aspects of development such as requirements, testing or developing the UI, documentation or maintaining released versions. You will work as different roles in the **Core Team** that develops core components for the project.

The hands-on is split into three parts. You will perform the work for these team members:

Scott – Scrum Master and Team Lead of the Core Team

Deb – A new developer in the Core Team

The material provided deliberately exceeds what is doable in the given time. The intent is to provide a complete scenario and some interesting optional information. The material is illustrated and can provide a reference even if not all labs are successfully finished. Please see https://jazz.net for additional information.

- 1 shows how Jazz based products help increase efficiency and quality providing feedback data in dashboards. In addition it shows agile planning from Scott's perspective which will result in a critical defect being assigned to Deb.
- **2** shows how Rational Team Concert help Deb fixing the defect. Because she is new in the project an approval for the change is required.
- **3** shows how Rational Team Concert helps Scott to review Deb's changes, accessing the relevant source code base, applying Deb's change, building and testing the new code.
- **Optional Labs**
    - Optional **4** shows how Rational Team Concert helps to access data from other tools such as test results from within Rational Team Concert.
    - Optional **2.35** shows how Rational Team Concert helps to collaborate on development artifacts using instant messaging.

Lab 1 Project Status and Planning

1.1 The Problem

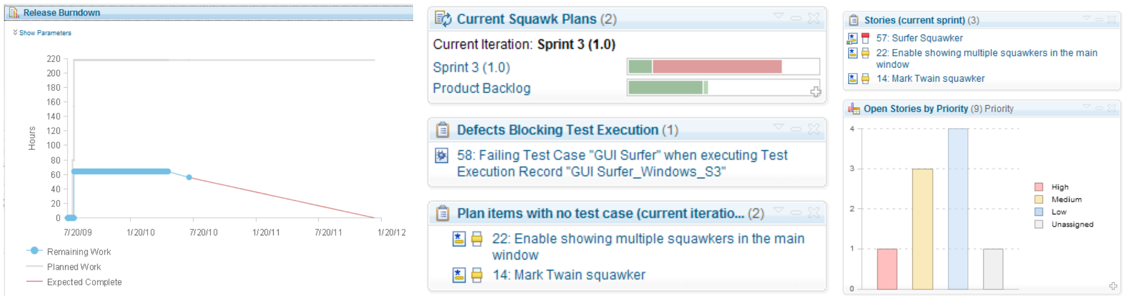
In most environments today, planning and keeping track of the plan, is one of the most frustrating activities, although it is most important and usually required. One reason is that data, required for planning, tends to be spread across tools, documents and people. Traditional development environments are not designed to keep all the data required to do the planning in a consistent way. Thus planning involves exporting and collecting data from various data sources such as tools, tables and plans as well as checking with the developers about progress on a regular basis. All that data and information needs merging, validating and updating of efforts and progress and verify the result with the team. Once the data is prepared it is usually outdated and probably invalid already.

1.2 The Solution

The Jazz Platform and Rational Team Concert in contrast provide a consistent data model that allows providing and maintaining all data necessary to support and perform planning. The data model allows defining the schedule as iteration structure, milestone target dates, current iteration or milestone. It allows defining team membership, work assignment to teams and scheduled absences such as vacation. In addition users can maintain estimates and progress for planned work in Rational Team Concert.

A built in planning component provides life plans based on this data at various levels. A personal plan allows team members to manage current and future work. Dedicated plan types for sprints, iterations and releases provide insight into the current plan status for one or multiple teams. Working with plans and data contributing to the plans can be performed in a Web2.0 browser as well as in a supported IDE such as Eclipse. Thus planning data is kept consistent and changes are immediately reflected in the plans. The Plans are always up to date. This saves time to perform productive work which would otherwise have been wasted on repetitive, error prone, manual work not feeding back into the projects data.

Rational Team Concert provides customizable dashboards to allow users to view data, status information and reports. To make projects more predictable RTC provides process templates and special dashboard report views such as Burn-up, Burn-down and Team velocity out of the box. These reports run on historical data aggregated from work items. They help to understand the current status and the expected progress. If used iteratively the gathered data can make development much more predictable.



Current Squawk Plans (2)
Current Iteration: **Sprint 3 (1.0)**
Sprint 3 (1.0)
Product Backlog

Defects Blocking Test Execution (1)
58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"

Plan items with no test case (current iteratio... (2)
22: Enable showing multiple squawkers in the main window
14: Mark Twain squawker

Stories (current sprint) (3)
57: Surfer Squawker
22: Enable showing multiple squawkers in the main window
14: Mark Twain squawker

In addition dashboards provide access to other Jazz based products such as Rational Quality Manager (see “Plan items with no test cases…” above) or Rational Requirements Composer. Dashboards can display artifacts from other domains and links between artifacts enable drill down and root cause analysis.

1.3 Lab: Plan the teams work

In this section you will see how RTC helps the team with planning and understanding the status of the current work.

The test team has discovered an issue with the latest build. A defect has been created and needs to be planned and assigned to a team member.

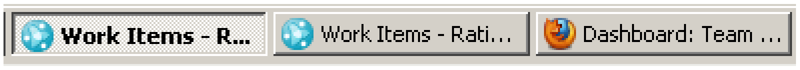
Team role

You are performing the role of Scott, the Scrum Master. In this section, you will use the dashboard Scott created to manage incoming work and use a plan to assign work to a resource based on responsibility and availability.

__1. Locate and prepare Rational Team Concert Eclipse Clients.

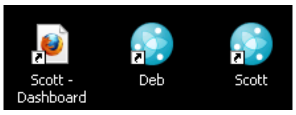
To work with Code Rational Team Concert provides a development client. Rational Team Concert's development client can run based on Eclipse or Microsoft Visual Studio.

__a. To save some time with the limited resources of the lab machines, servers and clients should be started in the beginning. The Lab instructor should have started the clients already. If this is the case you should be able to locate them in the taskbar:



__b. If they can be located skip to step __3 otherwise continue with step __4

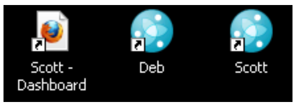
__2. To save time later start the Rational Team Concert Client for **Deb** and **Scott** using the provided Links

__a. Locate the quick-start links 

on the desktop.

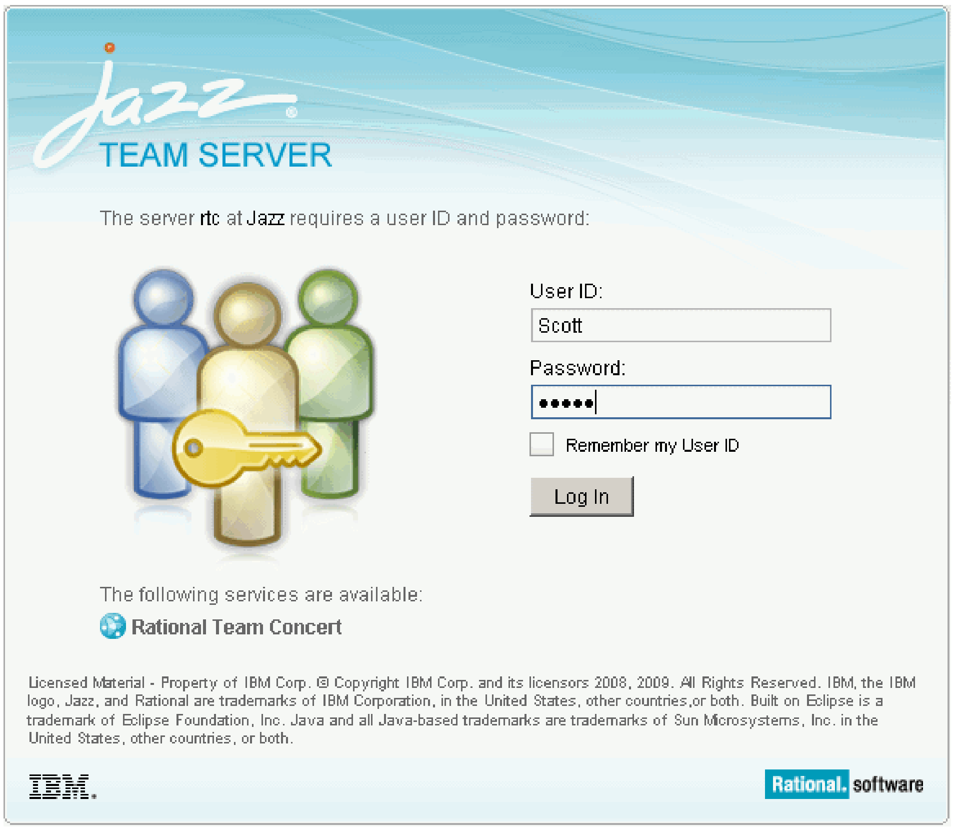
__b. Double-click on the provided links **Deb** and **Scott**

__3. The Web UI allows all roles to access and work with development data from anywhere without being required to install a client. Log in to Rational Team Concert's Web UI as Scott.

__a. Locate the quick-start links  on the Desktop

__b. Use the link to start the browser

__c. Enter `Scott` for both the **User ID** and **Password**, and then press **Log In**.



The server rtc at Jazz requires a user ID and password:

User ID: Scott

Password: •••••

Remember my User ID

The following services are available:

Rational Team Concert

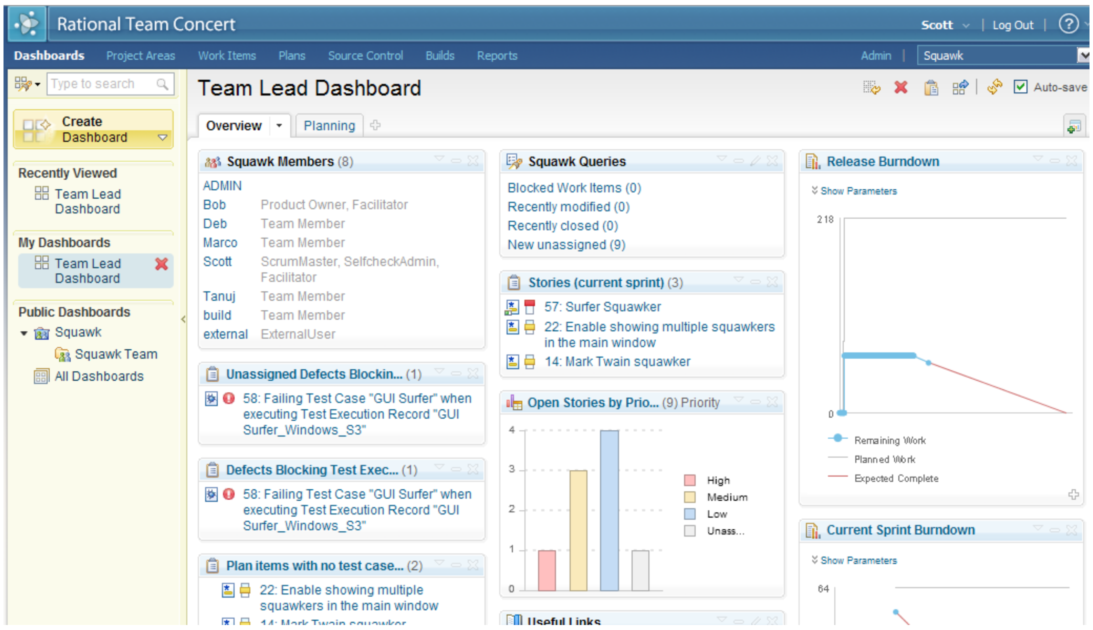
Licensed Material - Property of IBM Corp. © Copyright IBM Corp. and its licensors 2008, 2009. All Rights Reserved. IBM, the IBM logo, Jazz, and Rational are trademarks of IBM Corporation, in the United States, other countries,or both. Built on Eclipse is a trademark of Eclipse Foundation, Inc. Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

__4. Rational Team Concert displays the dashboard Scott created to get a better overview into the status of his team in the **Squawk** project. The viewlets of the dashboard have been configured to show various aspects interesting for the team lead. Dashboards can be created for projects, teams and individual users. They can be changed for various needs by any role with the required permission, adding tabs and adding or changing viewlets.

Browse the Team Lead Dashboard to get an overview of the project status.

__a. In our example, the Overview tab shows 3 columns of viewlets showing project members, query results, queries, graphical query representations and reports such as burn-down information.

All this information is key for a team lead to understand what is going on in the team.



Rational Team Concert — Scott | Log Out

Dashboards · Project Areas · Work Items · Plans · Source Control · Builds · Reports — Admin | Squawk

Team Lead Dashboard — Auto-save

Overview | Planning

Create Dashboard

Recently Viewed: Team Lead Dashboard

My Dashboards: Team Lead Dashboard

Public Dashboards: Squawk, Squawk Team, All Dashboards

Squawk Members (8)

ADMIN	
Bob	Product Owner, Facilitator
Deb	Team Member
Marco	Team Member
Scott	ScrumMaster, SelfcheckAdmin, Facilitator
Tanuj	Team Member
build	Team Member
external	ExternalUser

Unassigned Defects Blockin... (1)
- 58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"

Defects Blocking Test Exec... (1)
- 58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"

Plan items with no test case... (2)
- 22: Enable showing multiple squawkers in the main window
- 14: Mark Twain squawker

Squawk Queries
- Blocked Work Items (0)
- Recently modified (0)
- Recently closed (0)
- New unassigned (9)

Stories (current sprint) (3)
- 57: Surfer Squawker
- 22: Enable showing multiple squawkers in the main window
- 14: Mark Twain squawker

Open Stories by Prio... (9) Priority

Useful Links

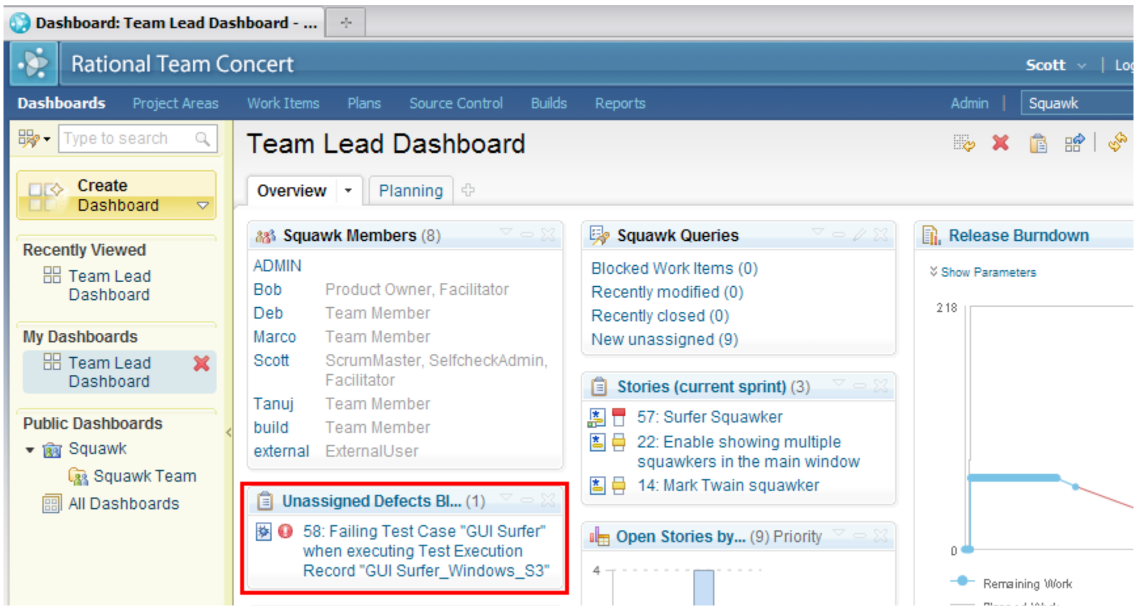
Release Burndown

Show Parameters

Current Sprint Burndown

Show Parameters

__b. The **Overview** tab shows "Unassigned Defects Blocking Test Execution".



Dashboard: Team Lead Dashboard - ...

Rational Team Concert — Scott | Log

Dashboards · Project Areas · Work Items · Plans · Source Control · Builds · Reports — Admin | Squawk

Team Lead Dashboard

Overview | Planning

Create Dashboard

Recently Viewed: Team Lead Dashboard

My Dashboards: Team Lead Dashboard

Public Dashboards: Squawk, Squawk Team, All Dashboards

Squawk Members (8)

ADMIN	
Bob	Product Owner, Facilitator
Deb	Team Member
Marco	Team Member
Scott	ScrumMaster, SelfcheckAdmin, Facilitator
Tanuj	Team Member
build	Team Member
external	ExternalUser

Unassigned Defects Bl... (1)
- 58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"

Squawk Queries
- Blocked Work Items (0)
- Recently modified (0)
- Recently closed (0)
- New unassigned (9)

Stories (current sprint) (3)
- 57: Surfer Squawker
- 22: Enable showing multiple squawkers in the main window
- 14: Mark Twain squawker

Open Stories by... (9) Priority

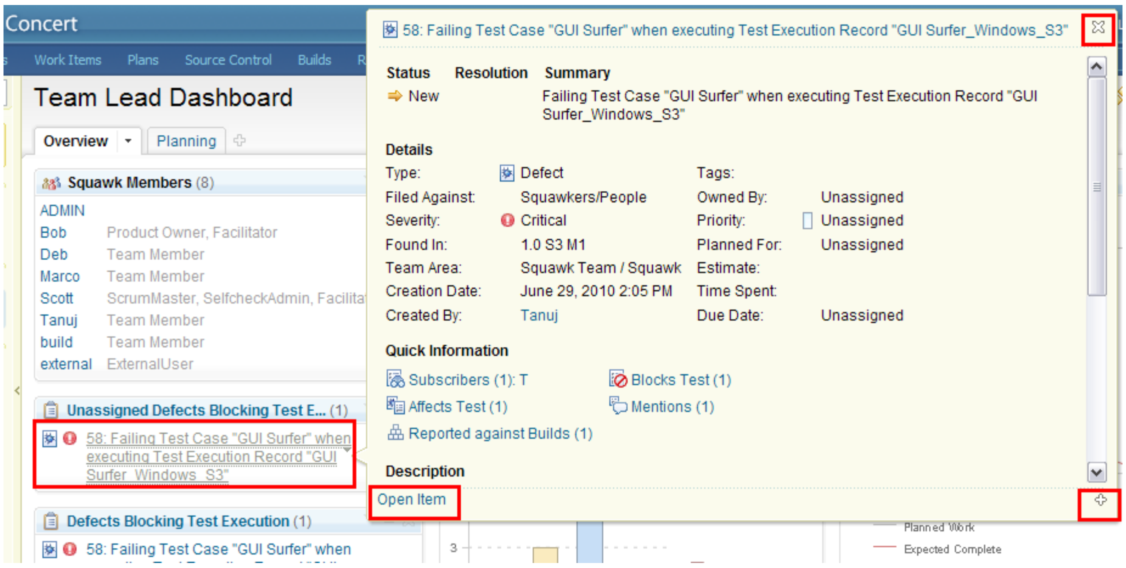
Release Burndown

Show Parameters

__5. RTC Dashboards provide tool tip hovers. These tool tips allow previews of artifacts to better understand the situation, without navigating away from the current page.

__a. Move the mouse over the Defect 58 in the viewlet showing "Unassigned Defects Blocking Test Execution". To make working and navigation in the browser more efficient Rational Team Concert shows a Tool Tip hover as preview where possible.

Press the ✥ button to pin the tool tip for browsing. The Details section shows the defect is critical and has been found in the current Milestone, has no owner and is not planned yet.


58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"

Status	Resolution	Summary
New		Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"

Details

Type:	Defect	Tags:	
Filed Against:	Squawkers/People	Owned By:	Unassigned
Severity:	Critical	Priority:	Unassigned
Found In:	1.0 S3 M1	Planned For:	Unassigned
Team Area:	Squawk Team / Squawk	Estimate:	
Creation Date:	June 29, 2010 2:05 PM	Time Spent:	
Created By:	Tanuj	Due Date:	Unassigned

Quick Information

Subscribers (1): T — Blocks Test (1)
Affects Test (1) — Mentions (1)
Reported against Builds (1)

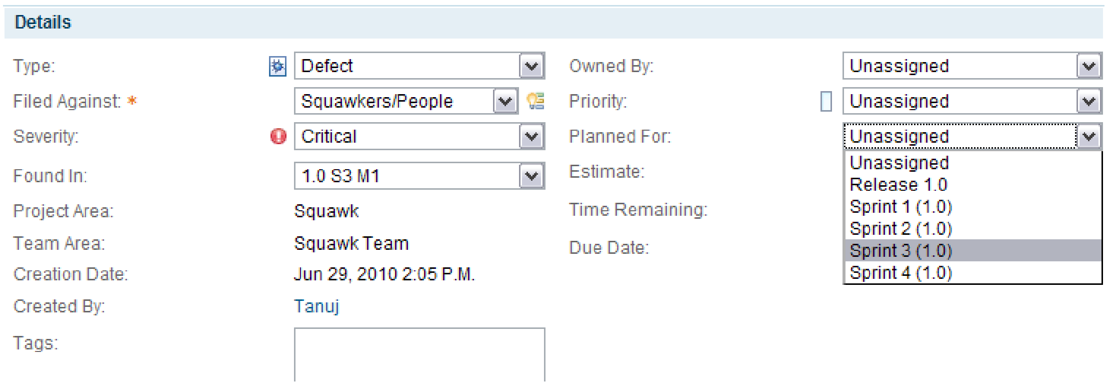
Description

Open Item

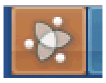
Concert
Work Items
Plans
Source Control
Builds
Team Lead Dashboard
Overview
Planning
Squawk Members (8)
ADMIN
Bob Product Owner, Facilitator
Deb Team Member
Marco Team Member
Scott ScrumMaster, SelfcheckAdmin, Facilita
Tanuj Team Member
build Team Member
external ExternalUser
Unassigned Defects Blocking Test E... (1)
58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"
Defects Blocking Test Execution (1)
58: Failing Test Case "GUI Surfer" when
Planned Work
Expected Complete

__b. Click **Open Item** to have a closer look. The defect needs to be fixed as soon as possible. Because Rational Team Concert knows the project schedule and milestones it is easy to plan work accordingly.

In the work Item editor set the **Planned For** attribute to the current **Sprint 3 (1.0)** and **Save** Save your work.


Details

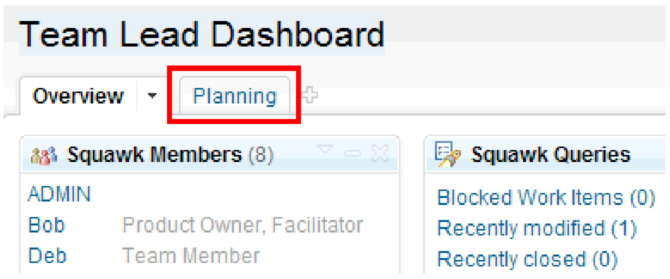
Type:	Defect	Owned By:	Unassigned
Filed Against: *	Squawkers/People	Priority:	Unassigned
Severity:	Critical	Planned For:	Unassigned
Found In:	1.0 S3 M1	Estimate:	
Project Area:	Squawk	Time Remaining:	
Team Area:	Squawk Team	Due Date:	
Creation Date:	Jun 29, 2010 2:05 P.M.		
Created By:	Tanuj		
Tags:			

__c. Use the Browser Navigation button or the button  in the page header to go back to the dashboard.

__d. Please note that Rational Team Concert also supports working with multiple browser tabs.

__6. Dashboards allow integrating viewlets for various purposes. To make it easier to manage different views they provide tabs for categorization.

__a. We still need to plan and assign the defect to a user to fix. Click the **Planning Tab** of the Dashboard.



Team Lead Dashboard

Overview | Planning

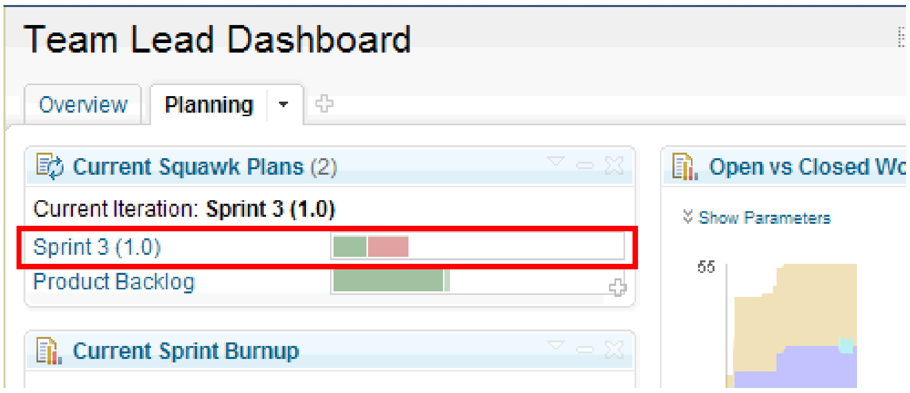
Squawk Members (8)

ADMIN	
Bob	Product Owner, Facilitator
Deb	Team Member

Squawk Queries

Blocked Work Items (0)
Recently modified (1)
Recently closed (0)

__b. The dashboard tab shows the plans related to the current sprint.



Team Lead Dashboard

Overview | Planning

Current Squawk Plans (2)

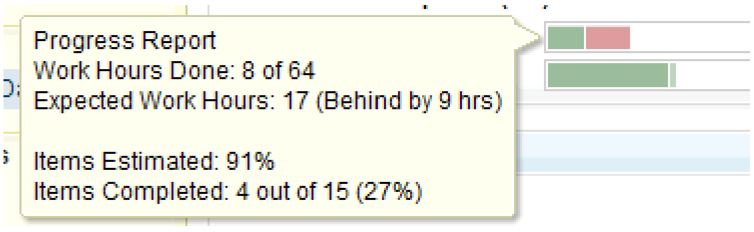
Current Iteration: **Sprint 3 (1.0)**

Sprint 3 (1.0)
Product Backlog

Current Sprint Burnup

Open vs Closed Wo

__c. Again tool tips help to get a first impression about the progress of the plan without navigating. Hover the mouse over the colored bar behind the **Sprint 3 (1.0)** plan. Because the progress report is calculated from all the data available to RTC including the current time and date you will see different numbers. The progress in this example shows in green that some estimated 8 hours of work has been done. Given the staff and assignment the expected work hours should be 17 so we are behind the plan. In addition you can see that 91% of all items on this plan are estimated. With this value the plan should be quite reliable. You can also see immediately that 27% of the planned work is already completed.

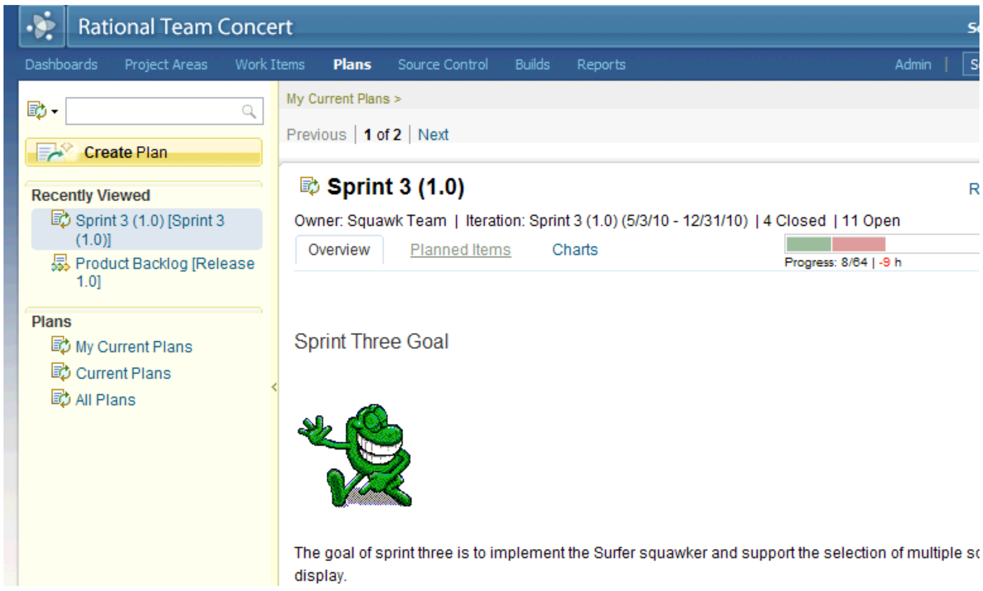

Progress Report
Work Hours Done: 8 of 64
Expected Work Hours: 17 (Behind by 9 hrs)

Items Estimated: 91%
Items Completed: 4 out of 15 (27%)

__7. To plan the new work of fixing the defect Rational Team Concert uses several mechanisms. One is the unique integrated planning component.

__a. Click the link named **Sprint 3 (1.0)** to open the plan.

__b. The plan opens at the **Overview Tab**. This tab is a Wiki used to document the goals of the plan for example. It is also possible to have links to important plan items and other artifacts on this page. This allows getting a quick overview. Additional Wiki Tabs can be created e.g. to document meetings, decisions or other team related information.



Sprint 3 (1.0)

Owner: Squawk Team | Iteration: Sprint 3 (1.0) (5/3/10 - 12/31/10) | 4 Closed | 11 Open

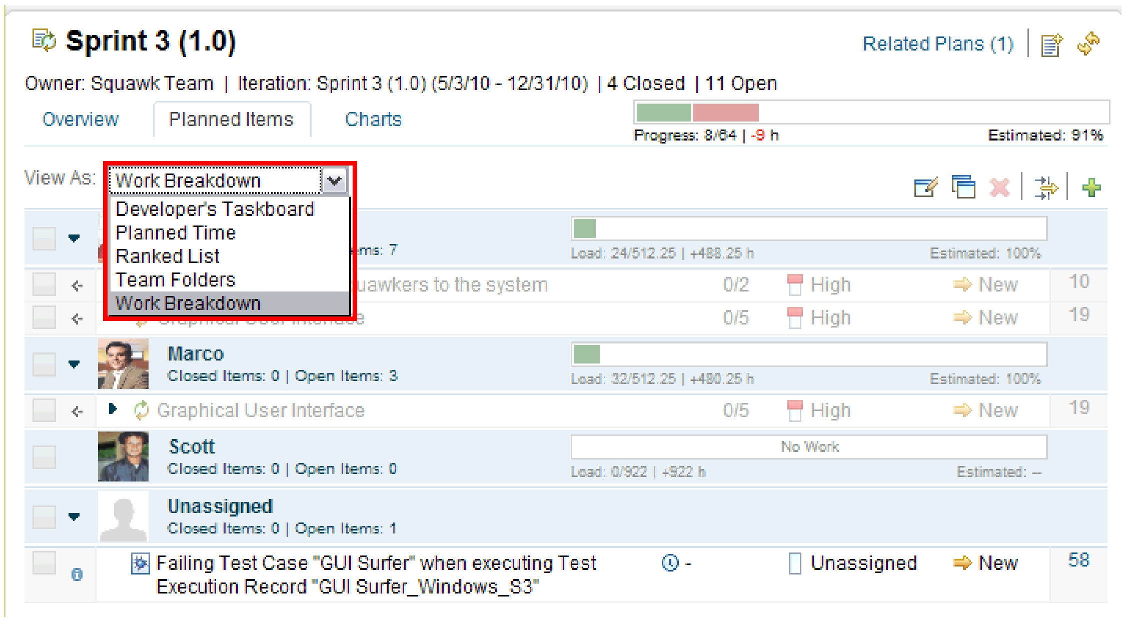
Progress: 8/64 | -9 h

Sprint Three Goal

The goal of sprint three is to implement the Surfer squawker and support the selection of multiple sc display.

__c. Click the **Planned Items** tab. The plan shows one of several possible views on the planning data. Having different views allows focusing on certain aspects and showing different attributes. It is possible to create new and customize existing views.

Switch to the **"Work Breakdown"** view if another view is selected, to see the parent child breakdown hierarchy of work items.



Sprint 3 (1.0) Related Plans (1)

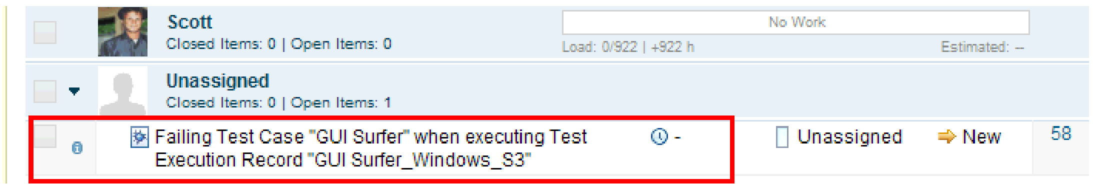
Owner: Squawk Team | Iteration: Sprint 3 (1.0) (5/3/10 - 12/31/10) | 4 Closed | 11 Open

Overview | Planned Items | Charts — Progress: 8/64 | -9 h — Estimated: 91%

View As: Work Breakdown
- Developer's Taskboard
- Planned Time
- Ranked List
- Team Folders
- Work Breakdown

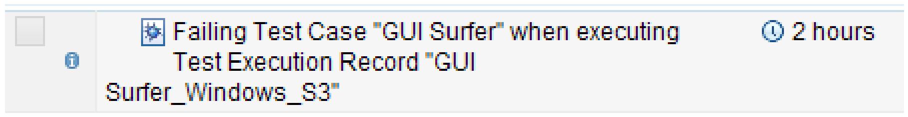
...ms: 7	Load: 24/512.25 | +488.25 h	Estimated: 100%		
...uawkers to the system	0/2	High	New	10
Graphical User Interface	0/5	High	New	19
Marco Closed Items: 0 | Open Items: 3	Load: 32/512.25 | +480.25 h	Estimated: 100%		
Graphical User Interface	0/5	High	New	19
Scott Closed Items: 0 | Open Items: 0	No Work — Load: 0/922 | +922 h	Estimated: --		
Unassigned Closed Items: 0 | Open Items: 1				
Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"	-	Unassigned	New	58

__d. The defect 58 needs to be estimated, prioritized and assigned to a user in the team.



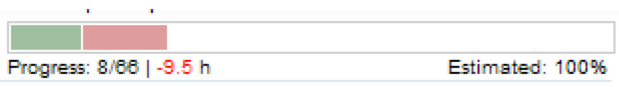
Scott Closed Items: 0 | Open Items: 0	No Work — Load: 0/922 | +922 h	Estimated: --		
Unassigned Closed Items: 0 | Open Items: 1				
Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"	-	Unassigned	New	58

__e. To **estimate** the effort hover the mouse over the clock shaped icon - . Left click and select 2 hours from the menu. The entry should look like below.



Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3" — 2 hours

Please note when assigning an estimate the progress bar shows the change immediately in the bar size as well as in the numbers. It would be obvious if the newly planned workload work load exceeded the remaining capacity.

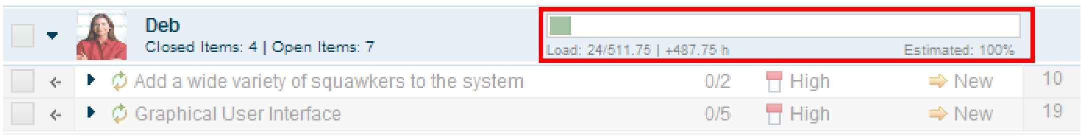


Progress: 8/66 | -9.5 h — Estimated: 100%

__f. Hover the mouse over the **Priority** symbol Unassigned , left click and select high priority High __g. The defect needs to be assigned to a responsible owner. This can be done in the defect or in the plan.

To understand the situation at first glance, the plan shows a load bar for each user. The individual load bar is calculated from the scheduled time remaining, the individual assignment to the team, scheduled absences and the estimated work load assigned to the user.

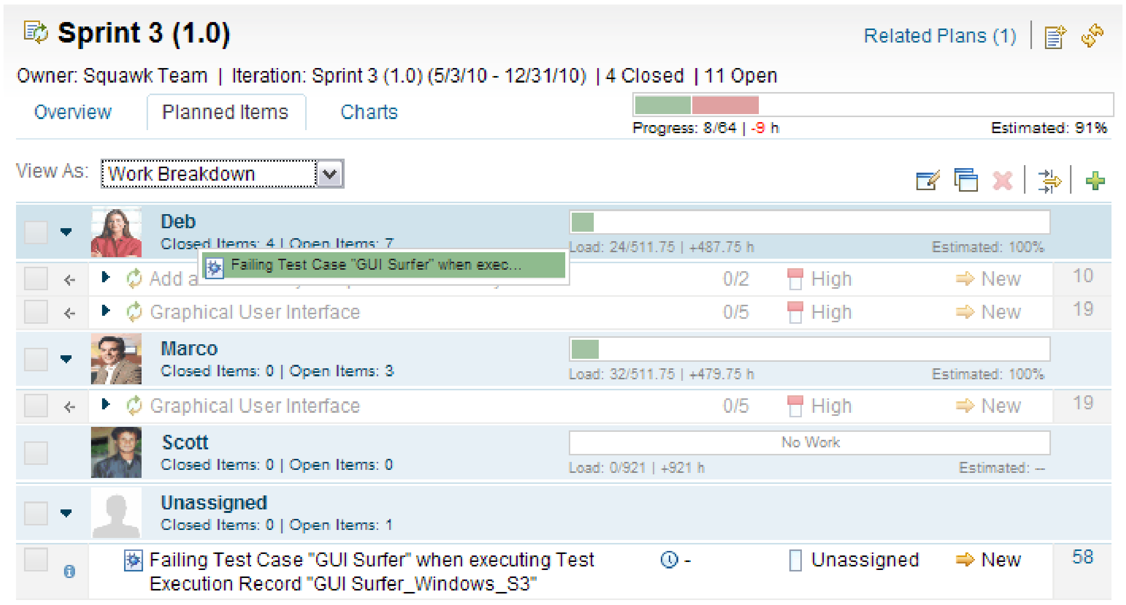
Deb apparently has enough time left – the bar is still green and shorter than the box.



Deb
Closed Items: 4 | Open Items: 7
Load: 24/511.75 | +487.75 h Estimated: 100%

Add a wide variety of squawkers to the system	0/2	High	New	10
Graphical User Interface	0/5	High	New	19

__h. To assign the defect to Deb, left click the defect and drag it over **Deb**'s section. Watch Deb's Load bar when dropping the work item. It is not required to save the change, this is performed automatically.



Sprint 3 (1.0) Related Plans (1)

Owner: Squawk Team | Iteration: Sprint 3 (1.0) (5/3/10 - 12/31/10) | 4 Closed | 11 Open

Overview | Planned Items | Charts

Progress: 8/64 | -9 h Estimated: 91%

View As: Work Breakdown

Deb
Closed Items: 4 | Open Items: 7
Load: 24/511.75 | +487.75 h Estimated: 100%

Failing Test Case "GUI Surfer" when exec...

Add a	0/2	High	New	10
Graphical User Interface	0/5	High	New	19

Marco
Closed Items: 0 | Open Items: 3
Load: 32/511.75 | +479.75 h Estimated: 100%

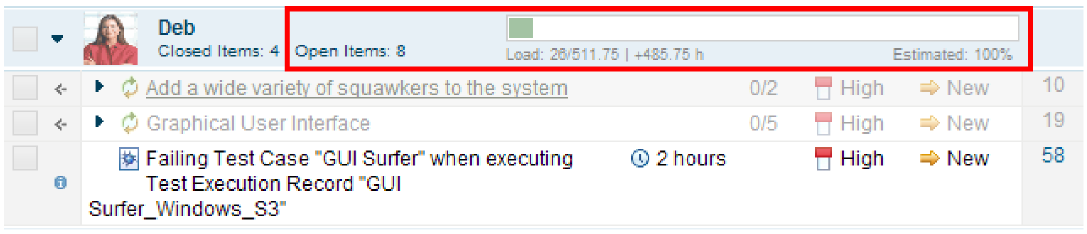
Graphical User Interface	0/5	High	New	19

Scott
Closed Items: 0 | Open Items: 0
No Work
Load: 0/921 | +921 h Estimated: --

Unassigned
Closed Items: 0 | Open Items: 1

Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"	-	Unassigned	New	58

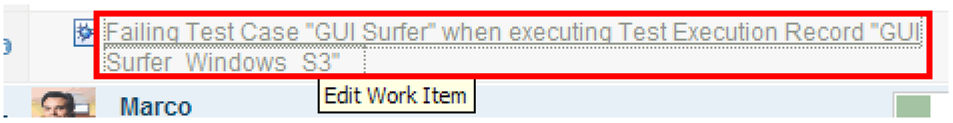
__i. The defects owner is now set to Deb. This adds one more item to Debs list of open items and increases her load by 2 hours. All this is immediately visible in the plan.



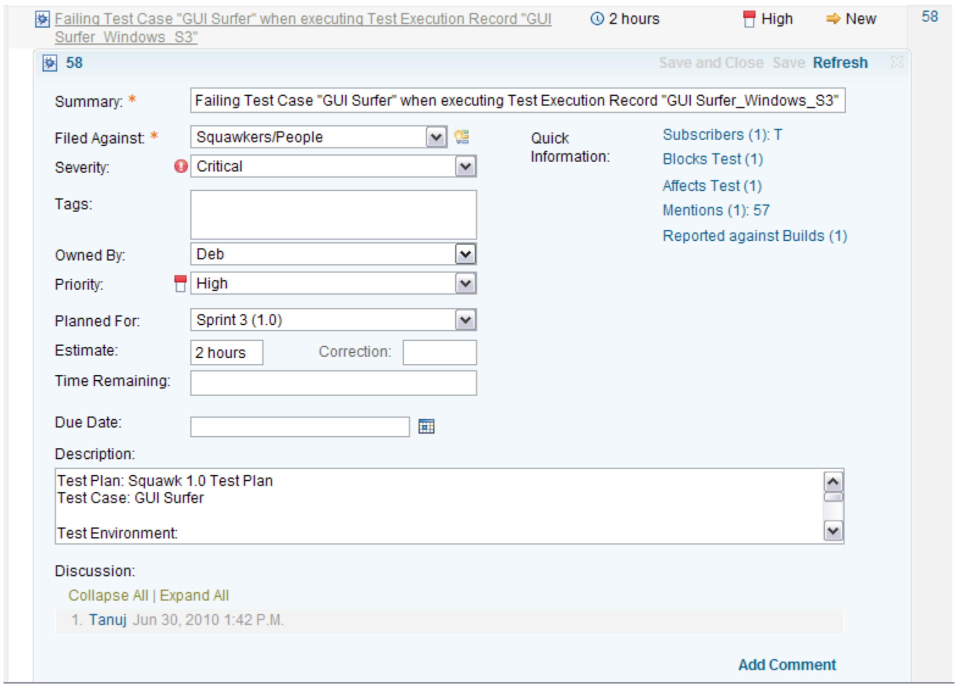
Deb
Closed Items: 4 | Open Items: 8
Load: 26/511.75 | +485.75 h Estimated: 100%

Add a wide variety of squawkers to the system	0/2	High	New	10
Graphical User Interface	0/5	High	New	19
Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"	2 hours	High	New	58

__j. Rational Team Concert's Web2.0 UI allows a working model similar to a rich client and greatly enhances the user experience with simplifying gestures that are often hard to do in Web UI's. Tool tips and Drag and drop are just some examples. Another one is the configurable in-plan editors giving access to the most important attributes. Click on the defects **Summary**.


Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"
Edit Work Item
Marco

An editor unfolds giving access to the most important attributes.


Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3" 2 hours High New 58

58 Save and Close Save Refresh

Summary: * Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"

Filed Against: * Squawkers/People

Severity: Critical

Tags:

Owned By: Deb

Priority: High

Planned For: Sprint 3 (1.0)

Estimate: 2 hours Correction:

Time Remaining:

Due Date:

Quick Information: Subscribers (1): T; Blocks Test (1); Affects Test (1); Mentions (1): 57; Reported against Builds (1)

Description:

Test Plan: Squawk 1.0 Test Plan
Test Case: GUI Surfer

Test Environment:

Discussion:

Collapse All | Expand All

1. Tanuj Jun 30, 2010 1:42 P.M.

Add Comment

Click on the Summary again to close the editor.

__k. This concludes the planning Lab.

1.4 Summary

In contrast to other environments Rational Team Concert supports planning based on consistent data available to the tool. The data includes planned schedule, assignment, current date, estimated and performed work. Instead of having to manually consolidate data from several data sources again and again, the data is consistently maintained during day to day work. Rational Team Concert can calculate and display plan and status in real-time. It helps planning by displaying the plan progress and the individual load and allows detecting and dealing with problems in the first possible moment. This supports the team working on a sustainable pace. The Team Lead is able to detect when developers get assigned too much work and react accordingly, either distributing the load to members with open capacity, or moving work into a later plan.

Historical data provides feedback and makes planning more accurate. It enables teams to analyze process related issues and refine their working model to optimize how they work leading to a better productivity while overtime is reduced. RTC supports calculating and displaying the team velocity (the amount of work/complexity a given team has proven to be able to handle in past iterations). In conjunction with Burn-up charts, this allows for better estimation as well as milestone projection. In addition anti-pattern like frequent reprioritizing, over-committing and over-loading of developers as a result are easy to discover. They can be dealt with and thus much less likely to happen. This again increases planning predictability and trustworthiness.

Lab 2 Defect Fixing Process and Review

2.1 The Problem

Teams are often dislocated, even within a single company. In addition more and more teams include external resources or submissions supplied by other companies. Working with an informal ad-hoc process is not an option in such environments. New methods such as agile development while being perceived as informal, in reality require quite a lot of discipline following a process. Many industries are recently imposed to safety and quality standards which increase the need to support processes.

To be efficient processes need to be supported and automated as much as possible. It is also necessary to constantly adapt and improve the process to increase efficiency. Most environments today are built on several tools. In most cases the tools involved have their own architecture, data model, workflow and user interface. Integrations are built later are hard to maintain and usually provide only limited capabilities for process support. The process definition is also often distributed across tools and hard to maintain and very complicated to use.

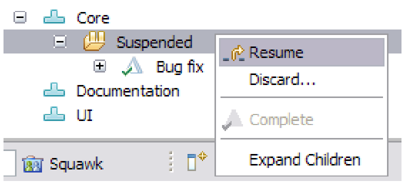
Software configuration management capabilities seem to be, at first glance, commoditized and simple to use. But often enough simplicity has downsides, e.g. a lack of isolation forces to either keep untested changes on the local machine or merge them into the team repository, potentially breaking the software.

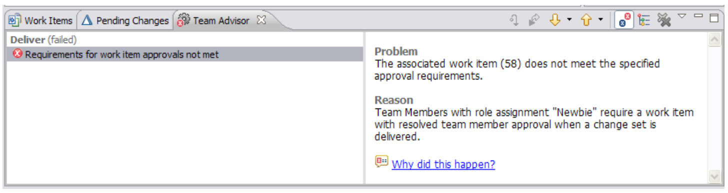
2.2 The Solution

The Jazz Platform and Rational Team Concert provide a consistent data and link model. Platform and tools have been designed for cross activity integration and process support. The process component allows flexible configuration on project, team, role level and even temporal level e.g. to have slightly different rules while development and close to a release. It can be configured and imposed on line from any team member with the required permissions. The Process can be extended if required and be adjusted over time.

The process component has built in Advisors that help user to follow the process with information, guidance and quick fixes. This allows to significantly reduce overhead and training cost for instance when bringing new user into a project.

The Rational Team Concert software configuration management capabilities are optimized for collaboration and data sharing. As a result they are built on a architecture with two stages on the server. This architecture provides private or public repository workspaces on the server that get automatically synchronized with local data. The repository workspaces connect to streams that are used by teams to share code.





Deliver (failed)

Requirements for work item approvals not met

Problem

The associated work item (58) does not meet the specified approval requirements.

Reason

Team Members with role assignment "Newbie" require a work item with resolved team member approval when a change set is delivered.

Why did this happen?

2.3 Lab Software Change and Process

In this section you will see how RTC helps a team member fixing a defect with easy access to related information such as a test result. In addition you will see advanced software configuration management and process control capabilities.

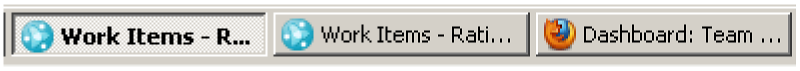
A defect discovered by the test team has been created and assigned to you now a fix needs to be created.

Team role

You are performing the role of Deb, the new developer. In this section, you will use the Rational Team Concert Eclipse IDE to identify your work and create a fix. The fix requires an approval which is enforced so you will ask an approver to perform it.

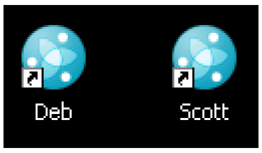
__1. Open the Rational Team Concert client using Deb's workspace.

__a. The Lab instructor should have started the clients already. If this is the case you should be able to locate them in the taskbar:



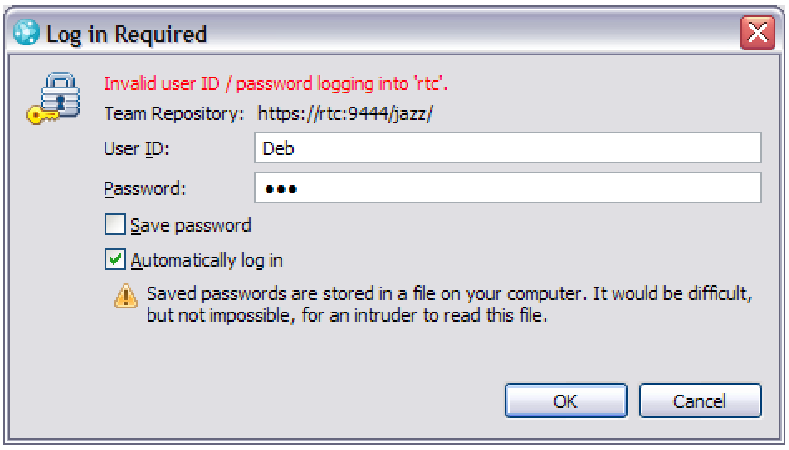
__b. If they can be located restore them until you see Deb's log-in screen and skip to step __3. Otherwise continue with step __2

__2. Double-click on the provided link for **Deb** and wait for the log-in screen to appear.



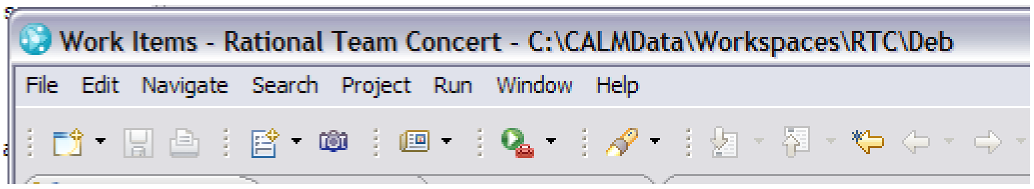
__3. In the log-in Screen

__a. Enter `Deb` for the **Password**, and then press **OK**. The connection process will take some time.



Invalid user ID / password logging into 'rtc'.
Team Repository: https://rtc:9444/jazz/
User ID: Deb
Password: •••
☐ Save password
☑ Automatically log in
Saved passwords are stored in a file on your computer. It would be difficult, but not impossible, for an intruder to read this file.

__4. Verify the headline shows **Work items – ….. – C:\CALMData\Workspaces\RTC\Deb**.

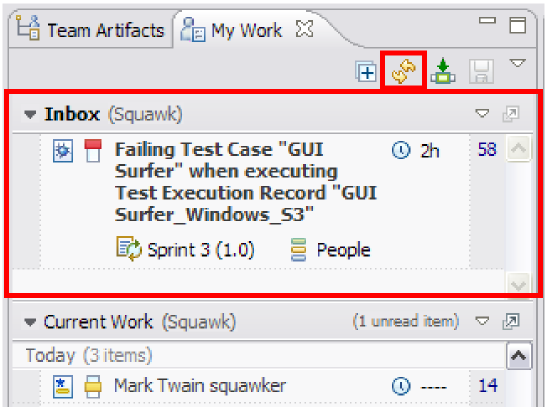

Work Items - Rational Team Concert - C:\CALMData\Workspaces\RTC\Deb
File Edit Navigate Search Project Run Window Help

__5. RTC provides various views on common artifacts such as defects and other work items. This helps presenting the right information immediately instead of forcing to search for it. Deb sees the new defect in her **My Work** view.

__a. On the left side of the Rational Team Concert window, switch to the **My Work** view.

__b. You should see the new defect in the Inbox. Click the **Refresh** icon at the top of the **My Work** view if not. Team Concert refreshes these views automatically after some time.

__c. Deb does not have to search for her work. Team Concert provides it for easy access. Her **Inbox** in the special personal planning view **My Work** contains the new blocking defect assigned to her.



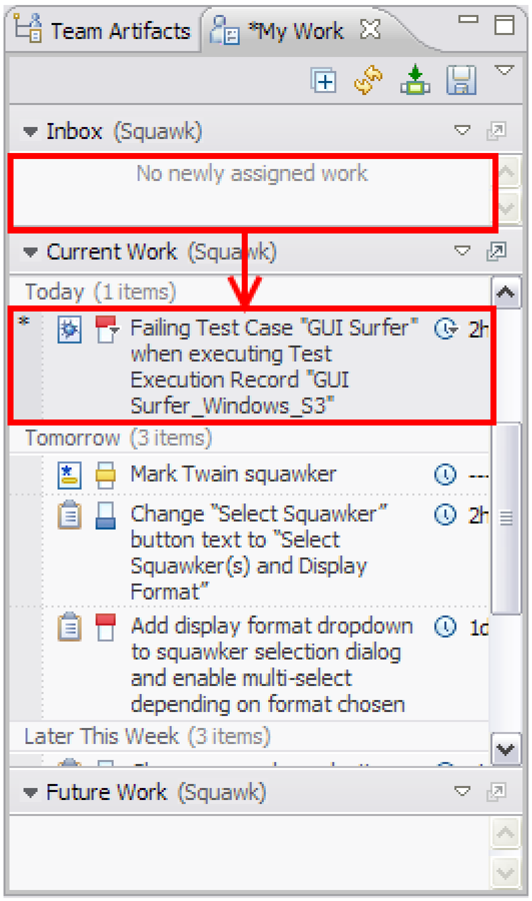
Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3" 2h 58
Sprint 3 (1.0) People

Mark Twain squawker ---- 14

__6. Deb decides that she needs to work on this right away.

__a. In the **My Work** view, drag and drop the defect to the top of the list under **Today**. Note that if the work item is still open in an editor, you will need to save it.

Please note that all the work previously planned for today is moving into the future based on the estimation.


Team Artifacts
*My Work
Inbox (Squawk)
No newly assigned work
Current Work (Squawk)
Today (1 items)
Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"
2h
Tomorrow (3 items)
Mark Twain squawker

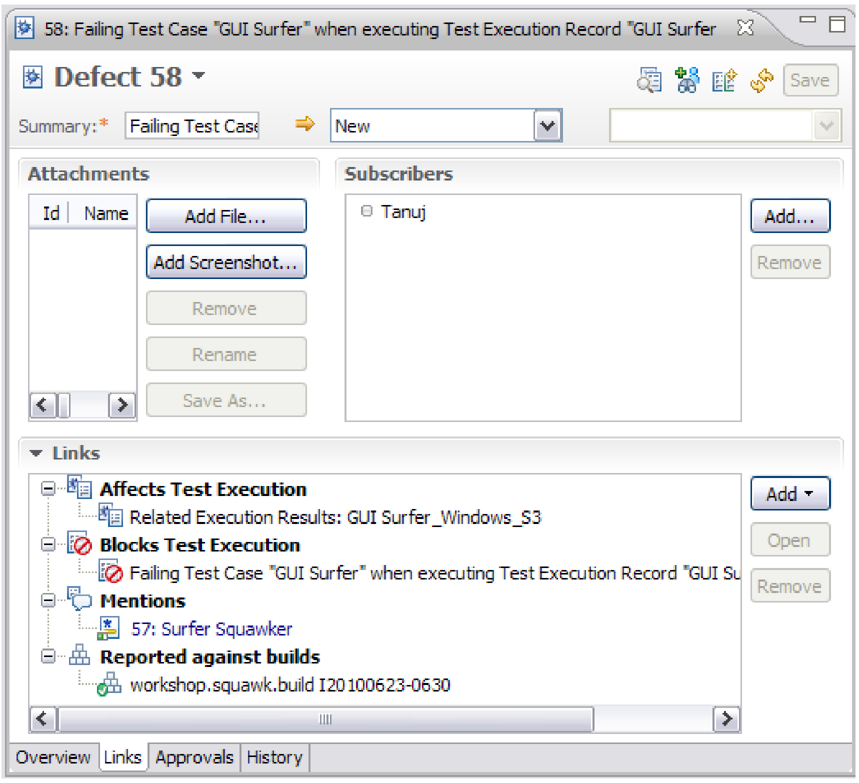
Change "Select Squawker" button text to "Select Squawker(s) and Display Format"
2h
Add display format dropdown to squawker selection dialog and enable multi-select depending on format chosen
1d
Later This Week (3 items)
Future Work (Squawk)

__7. Deb needs to understand what went wrong. RTC makes it very easy to navigate to the defect to find more details.

__a. In the **My Work** view, double-click the defect to open the work item editor.

__b. Read the defect description.

__c. Click the **Links** tab. In the links section RTC provides all other artifacts this defect is related to. In the links section you can see that the defect is related to a test execution result that failed. Because of this failure the defect blocks further tests of the test case linked below. The test execution record and the test case actually belong to a related test project in **Rational Quality Manager**. The data is stored on a server and can be directly accessed from here. The optional 4 shows the details


58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer

Defect 58

Summary:* Failing Test Case — New

Attachments — Subscribers: Tanuj

Links

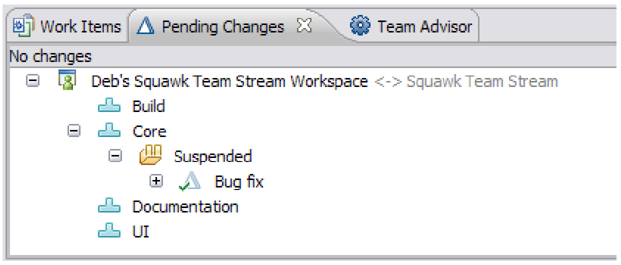
- Affects Test Execution
  - Related Execution Results: GUI Surfer_Windows_S3
- Blocks Test Execution
  - Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Su
- Mentions
  - 57: Surfer Squawker
- Reported against builds
  - workshop.squawk.build I20100623-0630

__8. Now Deb needs to fix the problem. Afterward she would have to test and deliver the change to her team and resolve the defect.

Note that normally Deb would copy the gif into the Java™ project and write some code to fix this defect; however, this has already been done in advance. A change set (all the required file additions and changes) has been prepared which reflects the bug fix in the code (just to save some time for this lab).

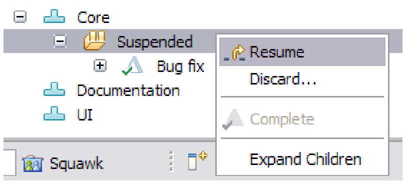
__a. To get to the fix switch to the **Pending Changes** view (in the lower middle section of the IDE) and click the **Expand to Change sets** icon ( ).

The change is available in Debs private work area on the server. It has been **suspended**. This capability of the RTC Jazz SCM system allows stopping the current unfinished work and switch to something more demanding. The suspended changes are pulled out and keep on the server.

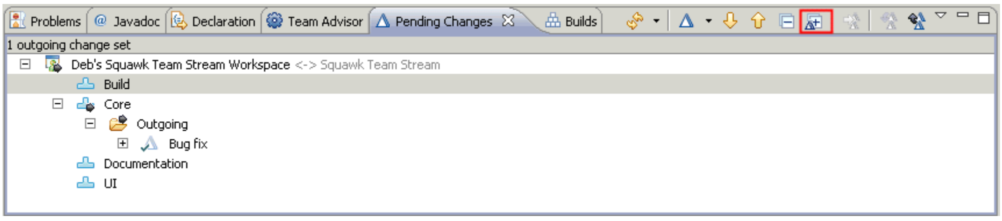


This is very useful, when working on bigger tasks. Incomplete changes can be pulled out and stored on the server to be resumed later. In the meantime an emergency fix can be performed without mixing up with the suspended work. There is also no need to deliver untested changes, which would affect the whole team, just for backup.

__b. To reactivate the fix right-click on the **Suspended** node Suspended and **Resume** the change.



__c. Rational Team Concert automatically reapplies the changes to your local workspace and stores them in the server side copy in your private repository workspace. Because it is private and on the server your changes are safe and included in backups. In addition you can collaborate on changes, e.g. send them to a colleague for review.

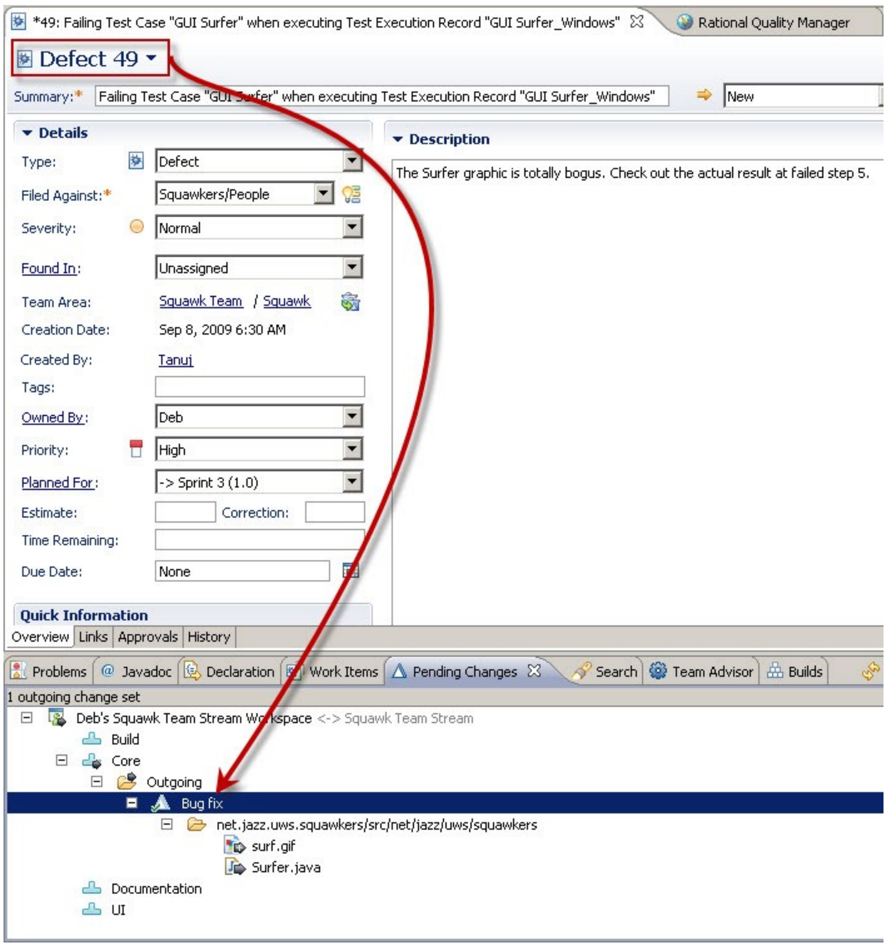


__9. Now the change has to be delivered to make it available to the team. Currently the change is only in Deb's private repository workspace. Rational Team Concert supports to deliver the change right away. However in the current setup this would fail due to the team process configuration. In order to document the reason for the change and to comply with the team process delivering changes require to provide the reason for the change. This is done by associating the source code change with the defect that was the reason for the change.

Why is this extra work worthwhile? Important reasons are collaboration, tracking, documentation and maintenance. The team has decided it requires every code change having a reason – at least in the current phase of the project. The team lead changed the process with some mouse clicks preventing anyone from delivering work to the team without a work item associated.

There are several ways to do the association. Rational Team Concert has also additional automation such that the developer can "Start working" on a work item and RTC automatically records all changes against this work item until the developer "Stops working" on the work item. However this doesn't apply to suspended change sets.

__i. One way is from the open work item editor for the surfer defect, drag and drop the **Defect <number>** to the change set in the **Pending Changes** view.


*49: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows"

Defect 49

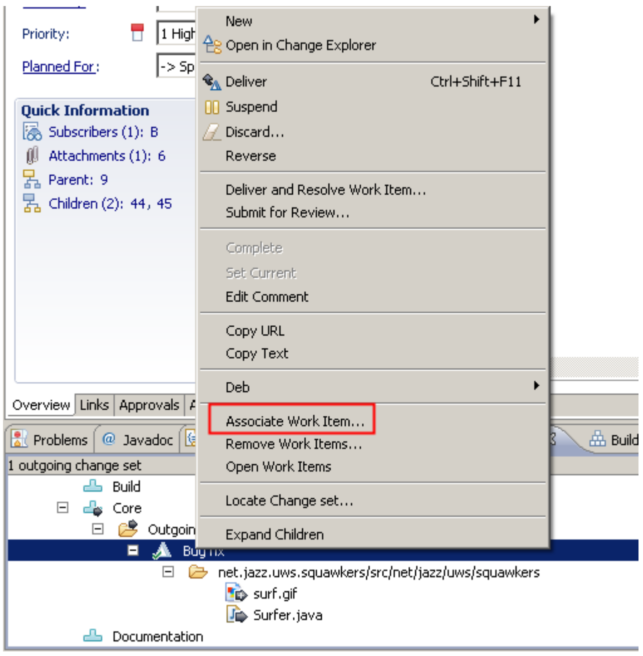
Summary: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows" — New

Type:	Defect
Filed Against:	Squawkers/People
Severity:	Normal
Found In:	Unassigned
Team Area:	Squawk Team / Squawk
Creation Date:	Sep 8, 2009 6:30 AM
Created By:	Tanuj
Tags:	
Owned By:	Deb
Priority:	High
Planned For:	-> Sprint 3 (1.0)
Estimate:	Correction:
Time Remaining:	
Due Date:	None

Description: The Surfer graphic is totally bogus. Check out the actual result at failed step 5.

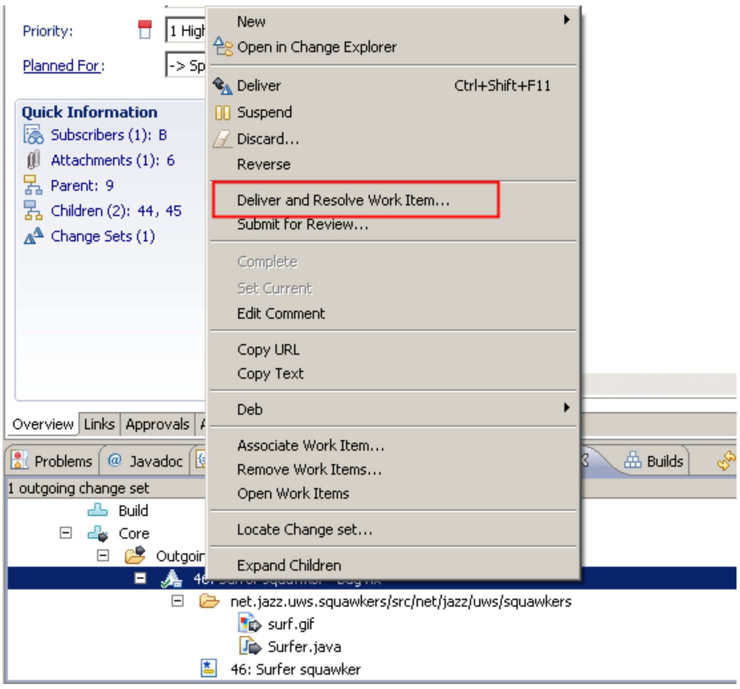
```
1 outgoing change set
Deb's Squawk Team Stream Workspace <-> Squawk Team Stream
    Build
    Core
        Outgoing
            Bug fix
                net.jazz.uws.squawkers/src/net/jazz/uws/squawkers
                    surf.gif
                    Surfer.java
    Documentation
    UI
```

__ii. Another way is to right-click the "change set in the **Pending Changes** view and use the **Associate Work Item…"** action from the menu.

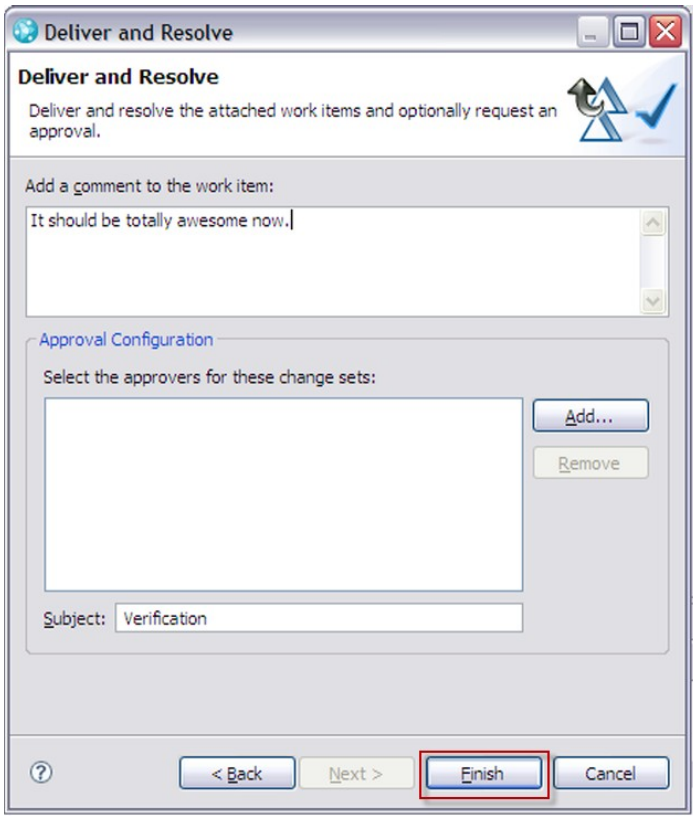


__10. Finally Deb should be able to deliver the change to the team.

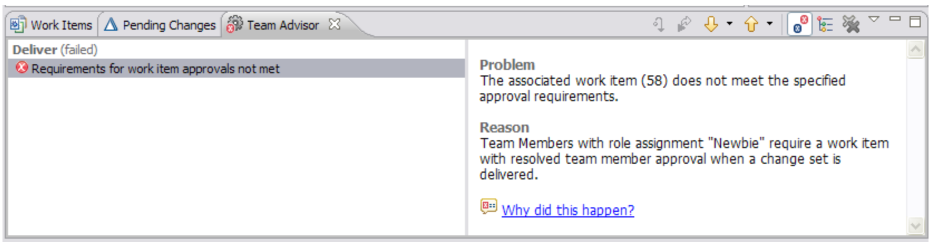
__a. Right-click the change set and run **Deliver and Resolve Work Item**.



__b. In the **Deliver and Resolve** dialog, enter `It should be totally awesome now.` In the **Add a comment to the work item** field and click **Finish**.


Deliver and Resolve
Deliver and Resolve
Deliver and resolve the attached work items and optionally request an approval.
Add a comment to the work item:
It should be totally awesome now.
Approval Configuration
Select the approvers for these change sets:
Add...
Remove
Subject: Verification
< Back
Next >
Finish
Cancel

__c. However, there seems to be a problem that needs to be investigated in the next step.



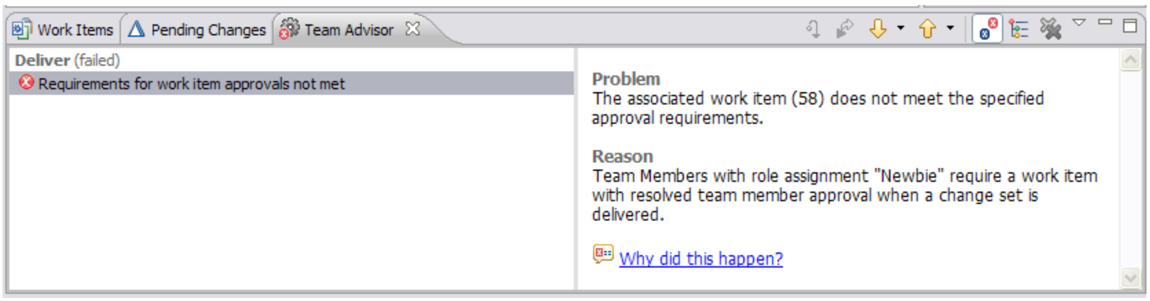
Problem
The associated work item (58) does not meet the specified approval requirements.

Reason
Team Members with role assignment "Newbie" require a work item with resolved team member approval when a change set is delivered.

__11. Every project needs a process to some extent. Often the process is not obvious or known to everybody. This typically results in strange tool errors and may require extensive consulting with an experienced colleague or process violations occur unintentionally and unnoticed. RTC provides process automation and guidance that helps following the process.

The team has decided that in the current sprint an approval is required before a “Newbie” can deliver. For example because it is intended to be a “hardening” sprint and the result is intended to be released. The process was configured with some mouse clicks to support that by the team lead and Scrumm Master Scott. Deb has the “Newbie” role and thus can’t deliver without an approval to the change. When trying to deliver her work without one, the process would be violated and RTC prohibits this.

Instead of submitting just an error message RTC provides a Team Advisor that explains the reason and helps following the process. Messages like the one below are configured in the process and triggered if violated.



Deliver (failed)
Requirements for work item approvals not met

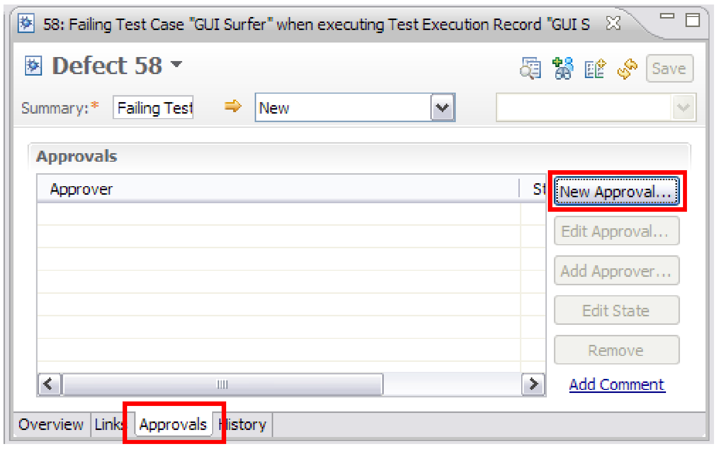
Problem
The associated work item (58) does not meet the specified approval requirements.

Reason
Team Members with role assignment "Newbie" require a work item with resolved team member approval when a change set is delivered.

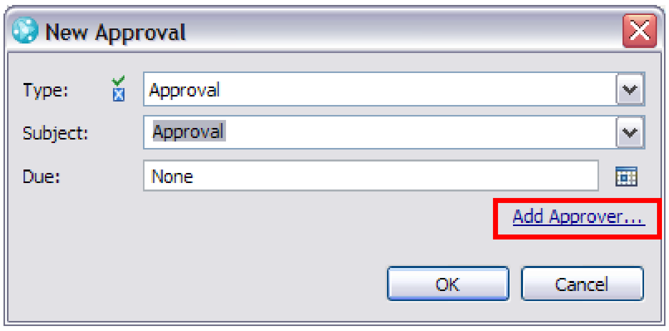
Why did this happen?

__a. To follow the process and add an approval, go to the open work item **defect 58**. If you closed the work item already you can reopen it from the **My Work** view as well as from the **Pending Changes** view.

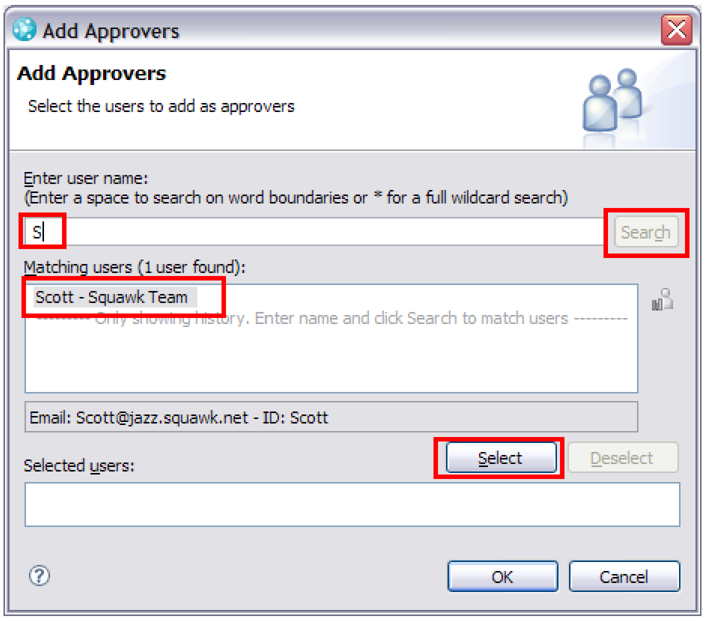
__b. Switch to the **Approvals** tab and press the **New Approval** button to add an approval.



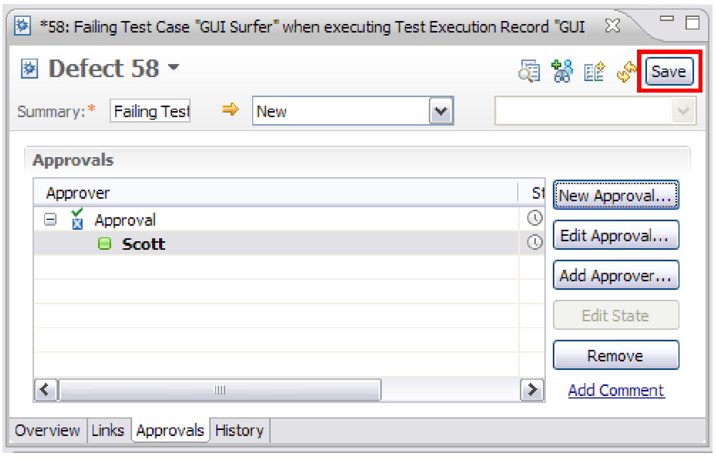
__c. On the new Approval click **Add Approver.**



__d. To find an approver e.g. the team lead, type **s** In the **Open Users** dialog. Press the **Search** button to find matching users. Double click the user **Scott** or press **Select** and **OK**.


Add Approvers
Add Approvers
Select the users to add as approvers
Enter user name:
(Enter a space to search on word boundaries or * for a full wildcard search)
s
Search
Matching users (1 user found):
Scott - Squawk Team
--------- Only showing history. Enter name and click Search to match users ---------
Email: Scott@jazz.squawk.net - ID: Scott
Selected users:
Select
Deselect
OK
Cancel

__e. Make sure to press **Save** to save your new approval work.



__12. While waiting for the approval Deb could **suspend** the change again and work on something else. Deb could also collaborate with Scott to speed it up a bit. Optional 5 shows how instant messaging can help working together.

2.4 Summary

In contrast to other environments Rational Team Concert supports individual planning of work and feeds the planning information back into the general planning component.

Rational Team Concert supports simple or complex co-located or distributed development with advanced collaboration capabilities that allow navigating across tool or domain boundaries using links. In addition Rational Team Concert adds collaboration capabilities such as instant messaging and RSS feeds into the context of development to increase performance.

Advanced software configuration management capabilities closely integrated with work management, build and process enactment increase productivity and help improving the process without getting into the way.

Running a process without tool guidance is very hard and will most likely fail over time. Rational Team Concert provides special capabilities to support processes on the one hand and help users to understand and follow the process on the other hand. This allows bringing new developers in without having them read hundreds of pages of process descriptions and go through long training classes followed by careful ongoing supervision to learn to follow the process.

No process was initially released without flaws. No process is perfect and equally optimized for all uses nor stays it over time. There is no one-size-fits-all process that works in every context and for every team. Rational Team Concert takes this into consideration combining customizable process templates for quick setup with easy to use and flexible process configuration capabilities. Rational Team Concert process configuration supports flexibility as needed allowing different teams in projects to run slightly different process configurations if desired. For example a documentation group has slightly different process requirements compared to teams doing new development or maintenance.

No vendor is able to support all needs, identified by customers over time, out of the box. Rational Team Concert allows extending the Eclipse Client and the Eclipse based Server component using the Eclipse extension mechanism. This technology has proven to be successful, scalable and reliable and is well known to many developers worldwide. Many partner solutions prove the potential and usefulness of this approach.

Lab 3 Verifying and Approving a Change

3.1 The Problem

Most environments today are built on several commercial or open source tools to support different roles in different activities. A typical open source example stack addressing change management, software versioning/configuration management, continuous build test and integration would be Bugzilla, CVS or Subversion plus Maven and several builders.

However, the more complex and distributed projects get, the better the integration between the components should be to gain more value. The information that a defect 123 has been fixed is valuable. Having the information that it has been fixed changing x,y and z lines of code in the files a.java, b.java and that this code has been included in build 567 is even more valuable. It is even more valuable if workflow automation could be provided based on that data and if users can easily navigate between the artifacts.

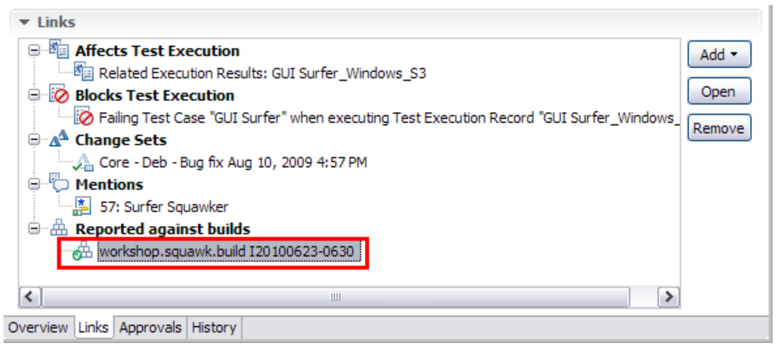
If all tools involved are developed completely independent and have their own data model, workflow and user interface integration of information and navigation becomes an issue. Inconsistent UI's and required background knowledge to use the integrations decrease efficiency and increase training costs. Obvious pattern such as client version dependencies also makes it hard to run these platforms.

3.2 The Solution

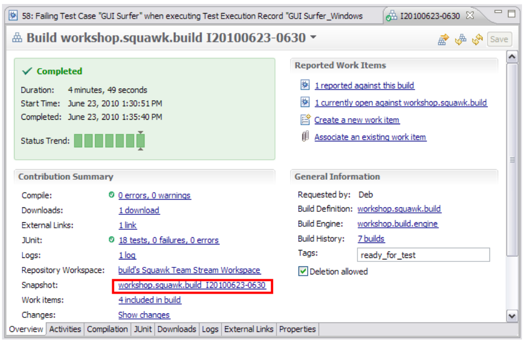
The Jazz Platform and Rational Team Concert provide a consistent data and link model. Platform and tools have been designed for cross activity integration. Emerging server integration standards such as OSLC (see http://open-services.net) allow integrating other compatible platforms while reducing the need and cost of client integrations.

Rational Team Concert provides work items for change management, software configuration management as well as continuous build and test support out of the box. All of these are tightly integrated which makes it easy to provide advanced automation making sure source code changes can be tracked to work items as easy as possible helping the users instead of getting into the way. The build system is integrated into the data model and a first class citizen in the IDE. It provides automatic linking code changes and work items to builds. The integration between work items, source code changes and builds helps to locate issues and related artifacts very easily.

On top of that with RTC in conjunction with tightly integrated Jazz based Requirement and Test Management solutions all roles involved with development and usage of applications in the lifecycle can be tied together. Testers can file and access defects while developers have access to test management data. This enables seamless collaboration between the roles and reduces the time required to locate information essential to work with.



Links
- Affects Test Execution
  - Related Execution Results: GUI Surfer_Windows_S3
- Blocks Test Execution
  - Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_
- Change Sets
  - Core - Deb - Bug fix Aug 10, 2009 4:57 PM
- Mentions
  - 57: Surfer Squawker
- Reported against builds
  - workshop.squawk.build I20100623-0630



58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows | I20100623-0630

Build workshop.squawk.build I20100623-0630

Completed

Duration:	4 minutes, 49 seconds
Start Time:	June 23, 2010 1:30:51 PM
Completed:	June 23, 2010 1:35:40 PM
Status Trend:	

Reported Work Items
- 1 reported against this build
- 1 currently open against workshop.squawk.build
- Create a new work item
- Associate an existing work item

Contribution Summary

Compile:	0 errors, 0 warnings
Downloads:	1 download
External Links:	1 link
JUnit:	18 tests, 0 failures, 0 errors
Logs:	1 log
Repository Workspace:	build's Squawk Team Stream Workspace
Snapshot:	workshop.squawk.build_I20100623-0630
Work items:	4 included in build
Changes:	Show changes

General Information

Requested by:	Deb
Build Definition:	workshop.squawk.build
Build Engine:	workshop.build.engine
Build History:	7 builds
Tags:	ready_for_test

✓ Deletion allowed

3.3 Lab

In this section you will see how RTC helps the team with easy access to several related artifacts simplifying work otherwise impossible to do.

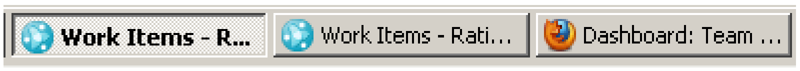
Deb has fixed a defect. Due to process rules the fix needs to be reviewed by a senior team member.

Team role

You are performing the role of Scott, the Scrum Master. In this section, you will use the RTC Eclipse IDE to review load and test Deb’s changes for the final approval.

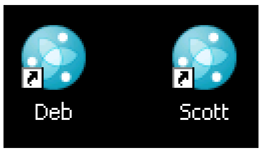
__1. Open the Rational Team Concert client using Scott’s workspace.

__a. The Lab instructor should have started the clients already. If this is the case you should be able to locate them in the taskbar:

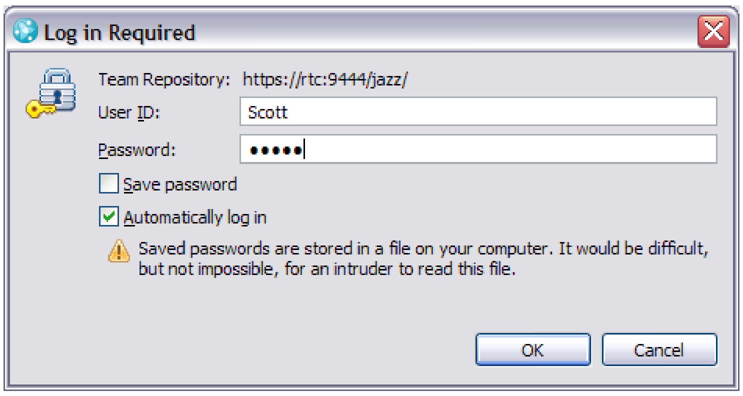


__b. If they can be located restore them until you see Deb’s log-in screen and skip to step __3. Otherwise continue with step __2

__2. Double-click on the provided link for **Scott** and wait for the log-in screen to appear.



__3. In the log-in Screen enter `Scott` for the **Password**, and then press **OK**. The connection process will take some time.



Log in Required

Team Repository: https://rtc:9444/jazz/

User ID: Scott

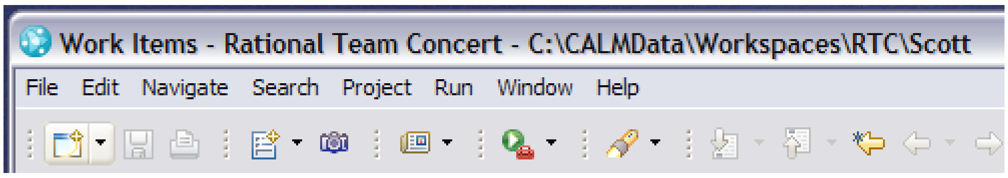
Password: •••••

☐ Save password

☑ Automatically log in

Saved passwords are stored in a file on your computer. It would be difficult, but not impossible, for an intruder to read this file.

__4. Verify the headline shows **Work items – ….. – C:\CALMData\Workspaces\RTC\Scott**.

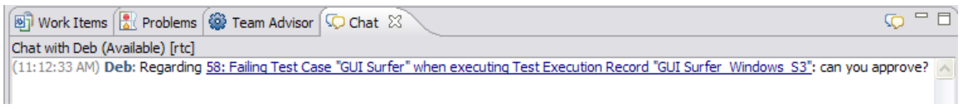

Work Items - Rational Team Concert - C:\CALMData\Workspaces\RTC\Scott
File
Edit
Navigate
Search
Project
Run
Window
Help

__5. Now Scott has to approve the change Deb made to fix the defect. This has several challenges. First Scott needs to know that he has to approve and what. He needs to access the details of the approval and ideally be able to test if the pending work is correct.

With RTC Scott is notified in various ways. Depending on the context RTC provides many easy ways to display and access referenced artifacts.

If he configured his personal preferences he will find an approval request in his mail inbox.

Scott could configure his Dashboard to show a query with **Pending Approvals**. RTC provides an instant messaging integration that allows sending artifact links in chats like below

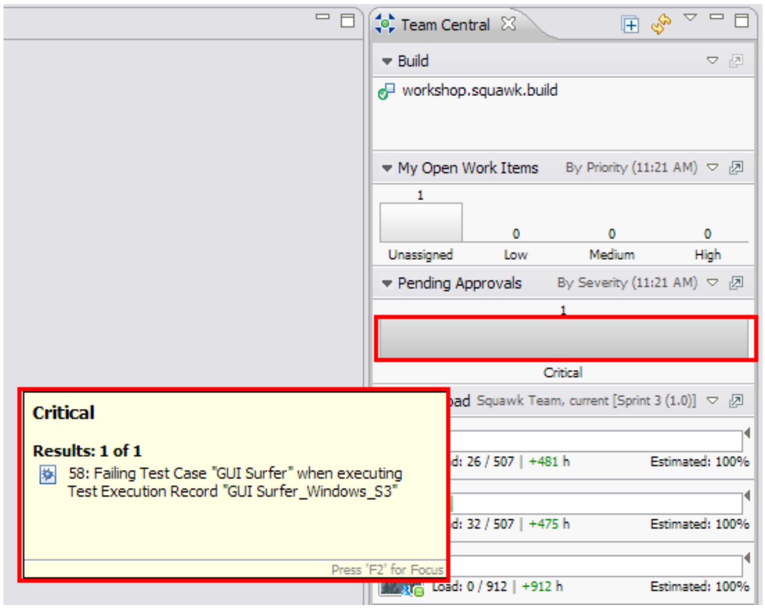


Chat with Deb (Available) [rtc]

(11:12:33 AM) Deb: Regarding 58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3": can you approve?

__a. Switch to the **Team Central** view. Find the show **Pending Approvals** section. Click the **Refresh** icon at the top of the Team Central view in case the section is empty.

__i. There is one pending approval for a critical work item pending. To open the request from here hover over the bar representing the critical approval request.



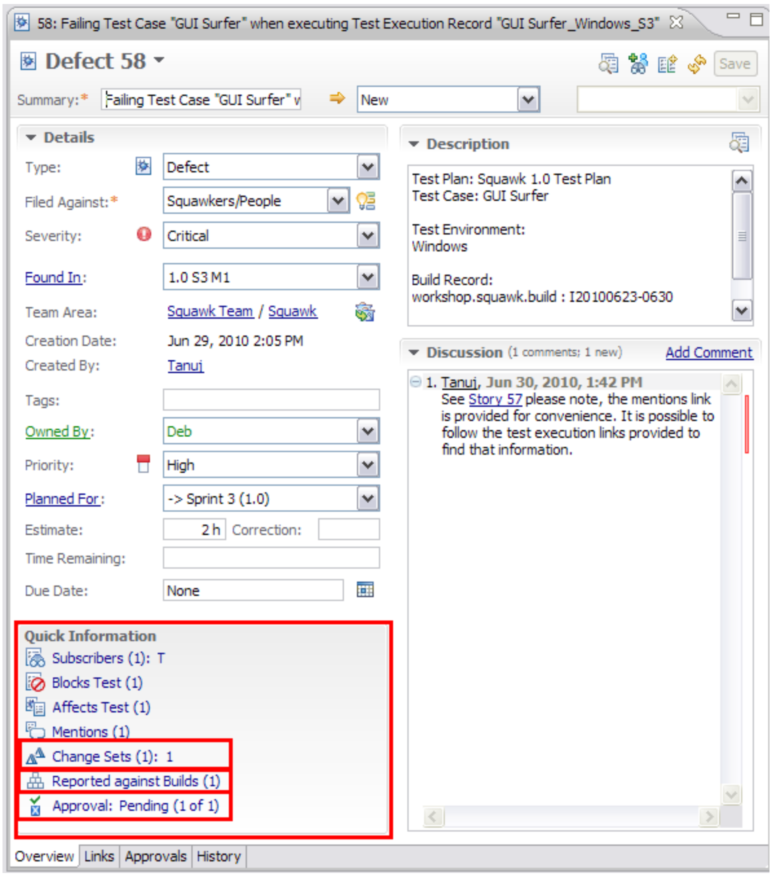
Critical

Results: 1 of 1

58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"

__ii. A tool tip shows a defect. Press **F2** to focus on the tool tip preview window. Move the mouse to the Link to "**58: Failing Test…**" and **click** to open it.

__6. RTC makes accessing the required information as easy as possible. After opening the defect most important information can be access from the overview tab. This includes the source code (**Change Sets** link), the **Build** used for the test, the test **result** and **failure information** and the **Approval** to approve or reject.



58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3"

Defect 58

Save

Summary:* Failing Test Case "GUI Surfer" v | New

Details

Type:	Defect
Filed Against:*	Squawkers/People
Severity:	Critical
Found In:	1.0 S3 M1
Team Area:	Squawk Team / Squawk
Creation Date:	Jun 29, 2010 2:05 PM
Created By:	Tanuj
Tags:	
Owned By:	Deb
Priority:	High
Planned For:	-> Sprint 3 (1.0)
Estimate:	2 h Correction:
Time Remaining:	
Due Date:	None

Quick Information
Subscribers (1): T
Blocks Test (1)
Affects Test (1)
Mentions (1)
Change Sets (1): 1
Reported against Builds (1)
Approval: Pending (1 of 1)

Description

Test Plan: Squawk 1.0 Test Plan
Test Case: GUI Surfer

Test Environment:
Windows

Build Record:
workshop.squawk.build : I20100623-0630

Discussion (1 comments; 1 new) Add Comment

1. Tanuj, Jun 30, 2010, 1:42 PM
See Story 57 please note, the mentions link is provided for convenience. It is possible to follow the test execution links provided to find that information.

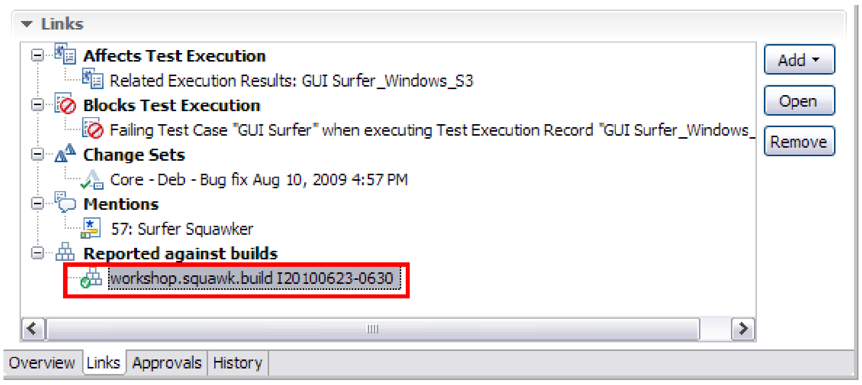
Overview | Links | Approvals | History

__7. Scott intents to test the original code with the changes. He needs to make sure that the fix does not affect the rest of the code and especially the integrity of the build. Therefore he wants to integrate the fix exactly into the code where the error has been detected. In most environments today Scott would be unable to produce that code without help of at least one person because the information is simply not available or not easy to access. Rational Team Concert however supports Scott to get it in some few easy steps.

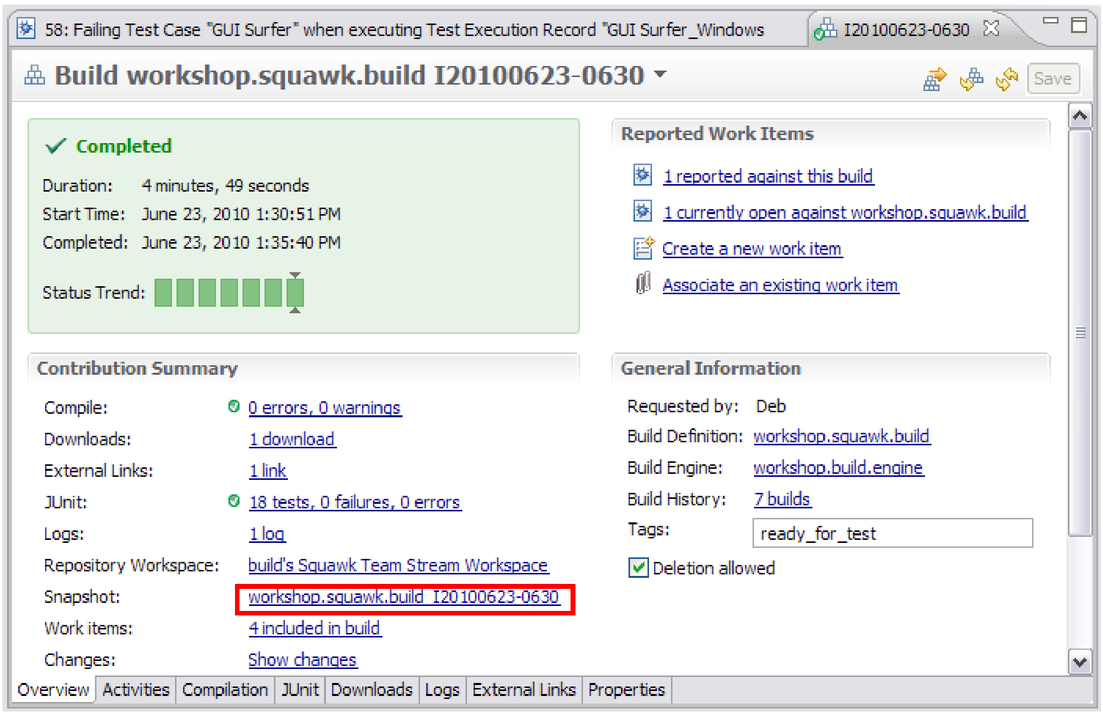
Of course Scott could call Deb or start a chat and ask for the Repository Workspace containing the code Deb used. But what if Deb is unavailable?

__a. To get the code that was used for the build, click the **Quick Information Reported against Builds (1)** to open the **Links** tab of the defect.

__b. To access the build result the defect is reported against double click the **Link** to the **Build Result**.


Links
Affects Test Execution
Related Execution Results: GUI Surfer_Windows_S3
Blocks Test Execution
Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_
Change Sets
Core - Deb - Bug fix Aug 10, 2009 4:57 PM
Mentions
57: Surfer Squawker
Reported against builds
workshop.squawk.build I20100623-0630
Add
Open
Remove
Overview
Links
Approvals
History

__c. RTC opens the build result and makes all relevant information available. The Build result contains a link to a so called **Snapshot** which allows recreating all code that contributed to the build result.


58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows | I20100623-0630

Build workshop.squawk.build I20100623-0630

Completed

Duration:	4 minutes, 49 seconds
Start Time:	June 23, 2010 1:30:51 PM
Completed:	June 23, 2010 1:35:40 PM
Status Trend:	

Reported Work Items
- 1 reported against this build
- 1 currently open against workshop.squawk.build
- Create a new work item
- Associate an existing work item

Contribution Summary

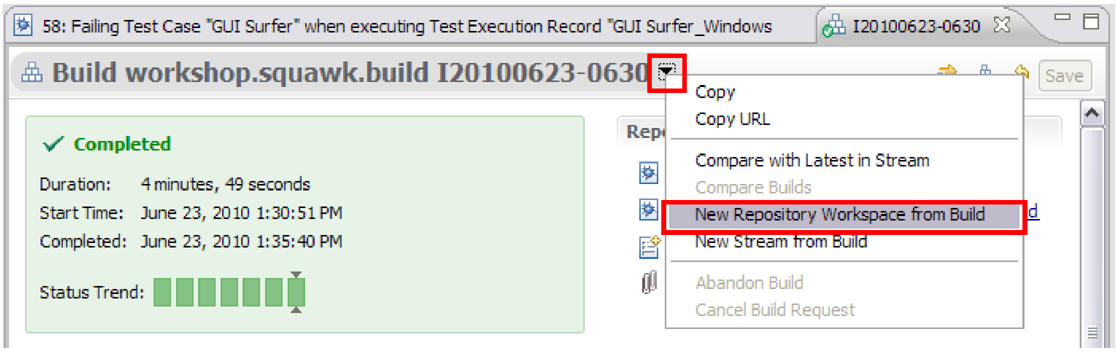
Compile:	0 errors, 0 warnings
Downloads:	1 download
External Links:	1 link
JUnit:	18 tests, 0 failures, 0 errors
Logs:	1 log
Repository Workspace:	build's Squawk Team Stream Workspace
Snapshot:	workshop.squawk.build_I20100623-0630
Work items:	4 included in build
Changes:	Show changes

General Information

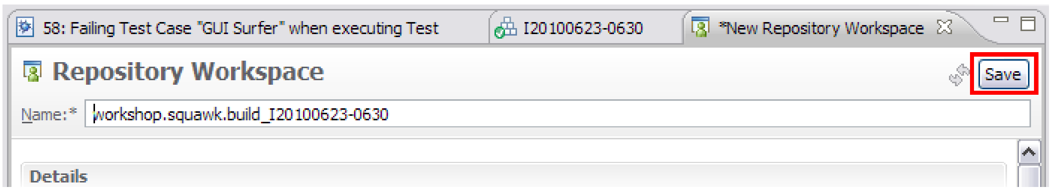
Requested by:	Deb
Build Definition:	workshop.squawk.build
Build Engine:	workshop.build.engine
Build History:	7 builds
Tags:	ready_for_test

✓ Deletion allowed

__d. Use the menu icon to access the menu for the build result. Select **New RepositoryWorkspace from Build** to get access to the source code for the build.


58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows
I20100623-0630
Build workshop.squawk.build I20100623-0630
Save
Completed
Duration: 4 minutes, 49 seconds
Start Time: June 23, 2010 1:30:51 PM
Completed: June 23, 2010 1:35:40 PM
Status Trend:
Copy
Copy URL
Compare with Latest in Stream
Compare Builds
New Repository Workspace from Build
New Stream from Build
Abandon Build
Cancel Build Request

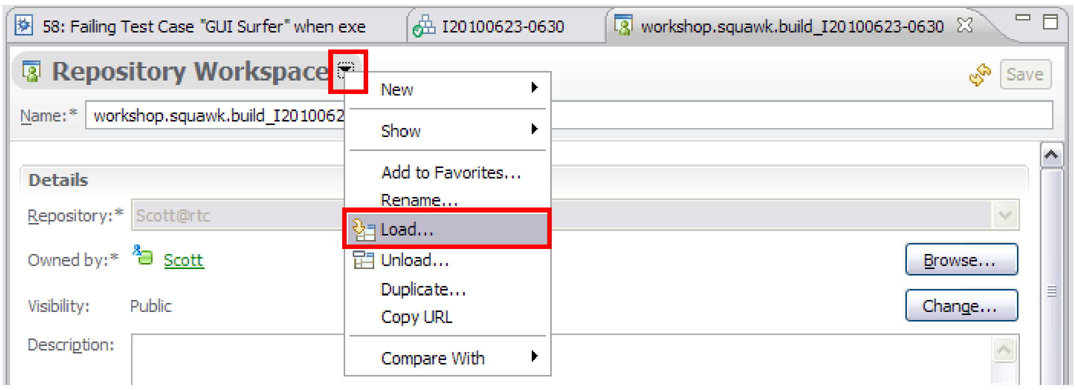
__e. Save the new Repository Workspace with the default name.


58: Failing Test Case "GUI Surfer" when executing Test
I20100623-0630
*New Repository Workspace
Repository Workspace
Save
Details

Name:* workshop.squawk.build_I20100623-0630

__f. You now have a private repository workspace that references the source code used by the build. This is on the server; to make it available in Eclipse Scott needs to load it into his local Eclipse workspace. RTC makes this is easy to achieve.

Use the icon to open the menu of the **Repository Workspace** and select **Load** to load the source to your local drive. If you discover any issues try to **refresh** the repository workspace and try the step again.



Name:* workshop.squawk.build_I2010062

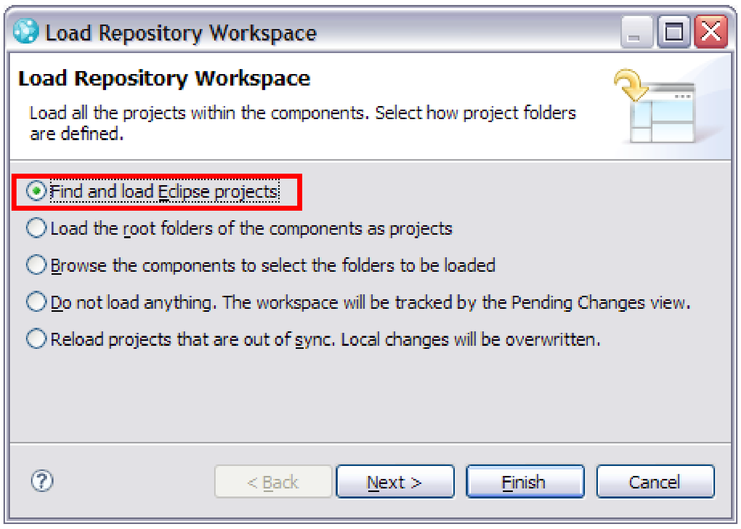
Repository:* Scott@rtc

Owned by:* Scott

Visibility: Public

Description:

__g. In the Load Wizard select **Find and load Eclipse projects** and press **Finish** to run the load process.



Load Repository Workspace

Load all the projects within the components. Select how project folders are defined.

[✓ Find and load Eclipse projects]

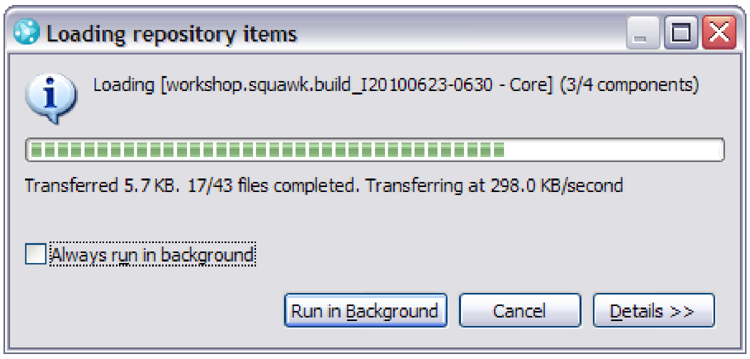
Load the root folders of the components as projects

Browse the components to select the folders to be loaded

Do not load anything. The workspace will be tracked by the Pending Changes view.

Reload projects that are out of sync. Local changes will be overwritten.

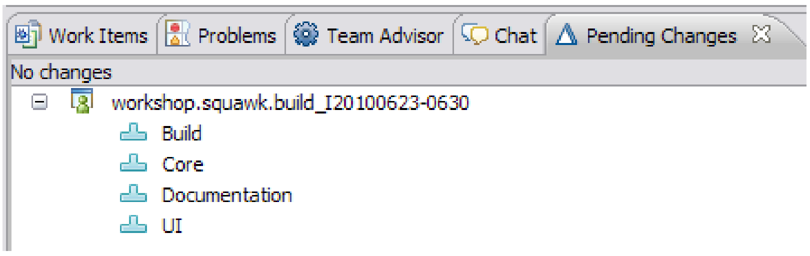
__h. The load process takes some time. Wait until the load process is finished.



Loading [workshop.squawk.build_I20100623-0630 - Core] (3/4 components)

Transferred 5.7 KB. 17/43 files completed. Transferring at 298.0 KB/second

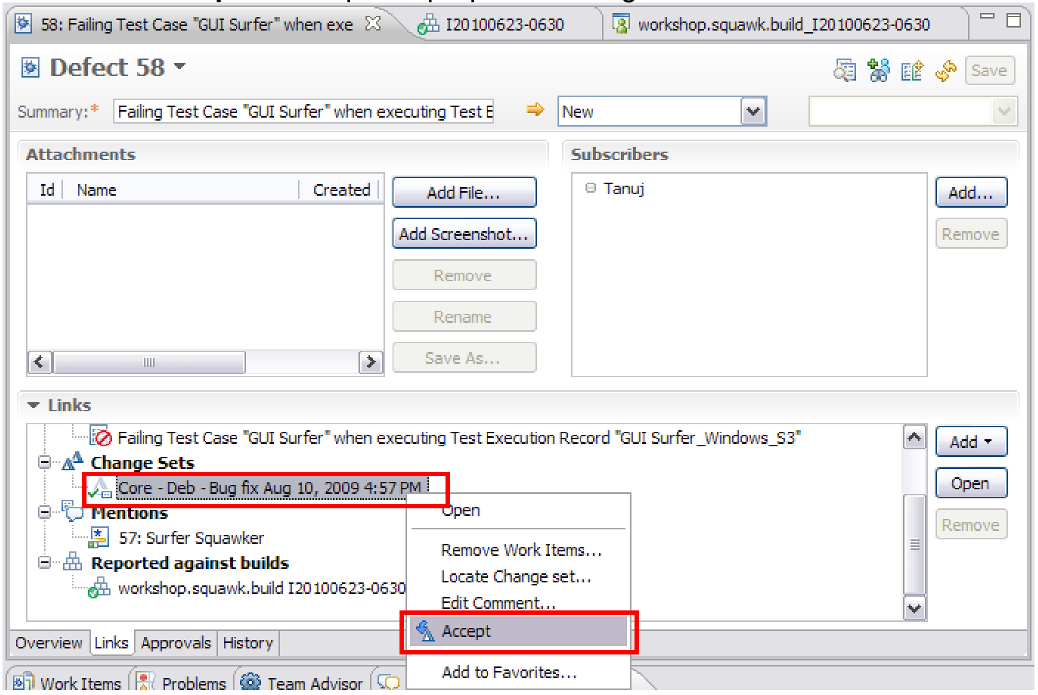
Once the load process has finished you see the components loaded in the pending changes view.



__8. Now that the source code is so easily available it is possible to bring in Deb's changes. Again RTC allows doing this from the most logical place: the defect Scott needs to approve.

__a. **Navigate** back to the **defect** for example use the link in the build result.

On the **Links** tab locate the **Change sets** node. **Right click** on Deb's **change set** and select **Accept** to accept the proposed changes into the code.



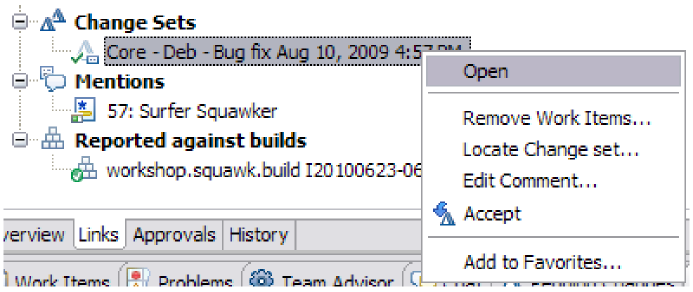
Summary:* Failing Test Case "GUI Surfer" when executing Test E

New

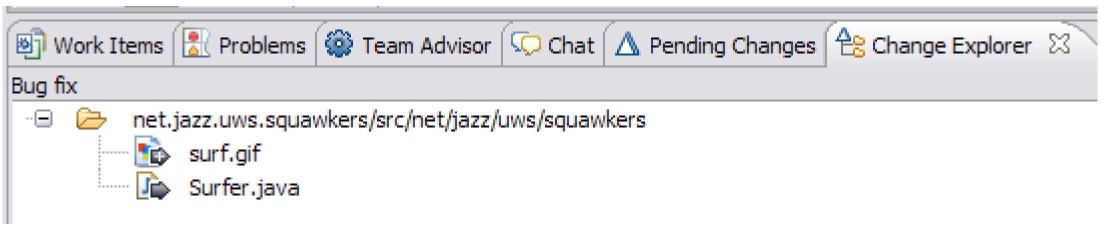
After the operation finished your local source code is up to date with the changes.

__9. Optional: You can easily review the changes you just accepted.

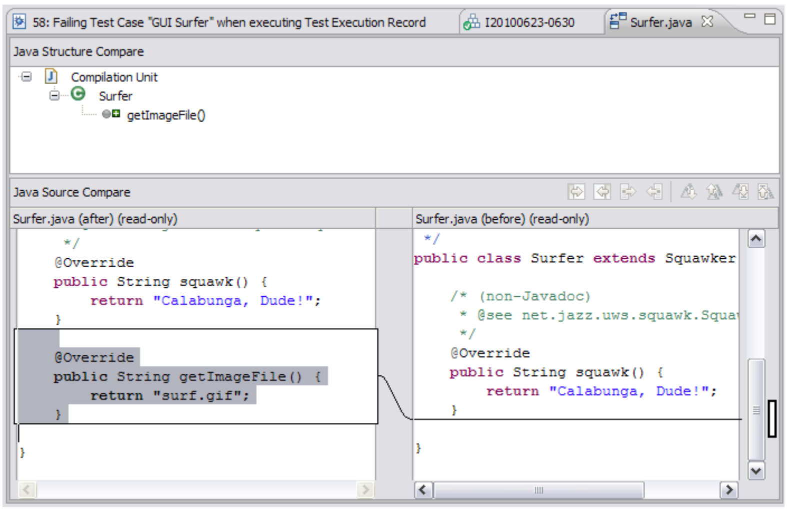
__a. On the defect's **links** tab, right click **Deb's** change set and select open.



__b. The change explorer view opens and shows the changed and added files.



__c. To review the details of the changes for example double click **Surfer.java** and RTC brings up a compare dialog highlighting the change.

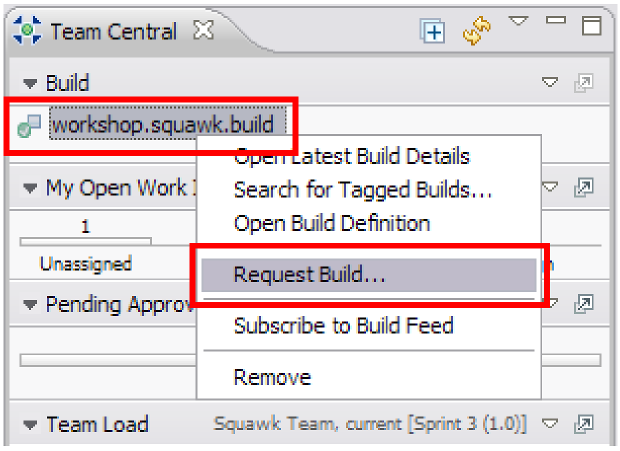

58: Failing Test Case "GUI Surfer" when executing Test Execution Record
I20100623-0630
Surfer.java
Java Structure Compare
Compilation Unit
Surfer
getImageFile()
Java Source Compare
Surfer.java (after) (read-only)
Surfer.java (before) (read-only)

```
     */
    @Override
    public String squawk() {
        return "Calabunga, Dude!";
    }

    @Override
    public String getImageFile() {
        return "surf.gif";
    }

}
```

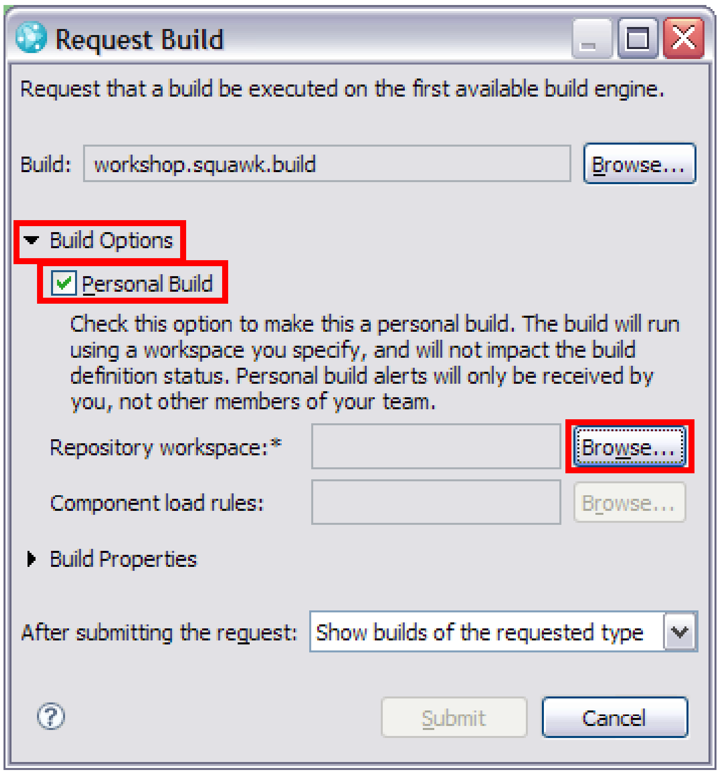
```
 */
public class Surfer extends Squawker

    /* (non-Javadoc)
     * @see net.jazz.uws.squawk.Squa
     */
    @Override
    public String squawk() {
        return "Calabunga, Dude!";
    }

}
```

__10. Running a build can make sure we did not introduce errors to running components. But in most SCM systems local changes like above are only available on the local computer. A build can be a long running operation and running a build locally would prevent working on other things. To be able to build on a server with many other SCM systems would require the code that needs to be built being check in. A temporary private branch would be an option. To set all this up is a major effort.
RTC keeps repository workspaces on the server as private system for code versioning. The changes above are therefore already available on the server and it is possible to run a build with the "official" build system on the private files.

__a. In the **Team Central** view in the section **Build** right click the **workshop.squawk.build**. Use **Request Build** in the context menu to run a build.



__b. In the following **Request Build** dialog configure the build. First **expand** the **Build Options**, select **Personal Build** and **Browse** for a Repository Workspace


Request Build

Request that a build be executed on the first available build engine.

Build: workshop.squawk.build Browse...

Build Options

Personal Build

Check this option to make this a personal build. The build will run using a workspace you specify, and will not impact the build definition status. Personal build alerts will only be received by you, not other members of your team.

Repository workspace:* Browse...

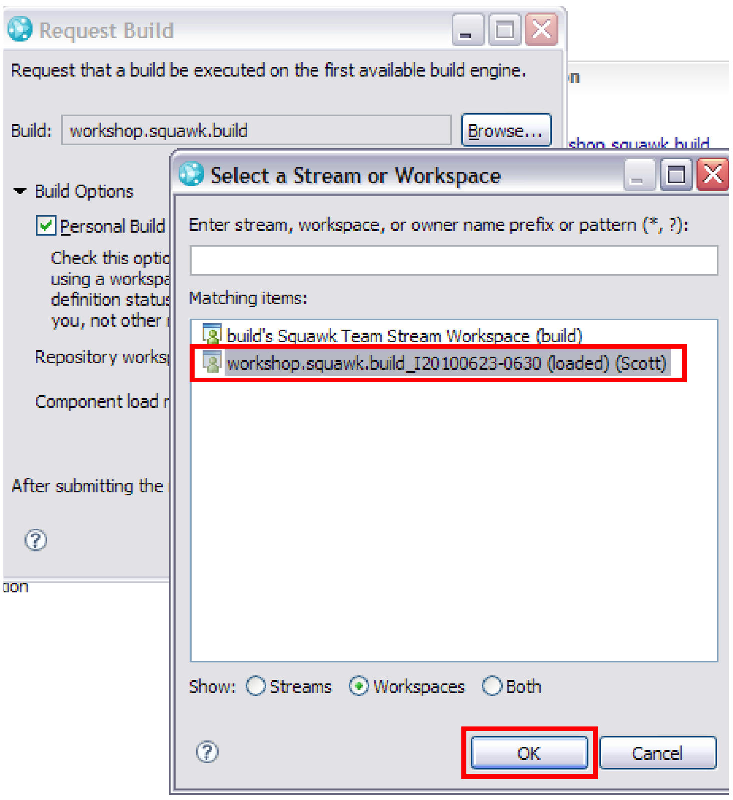
Component load rules: Browse...

Build Properties

After submitting the request: Show builds of the requested type

Submit Cancel

__c. In the Workspace Selection dialog select the repository workspace that is marked as **(loaded) (Scott)** and click **OK**.


Request Build

Request that a build be executed on the first available build engine.

Build: workshop.squawk.build Browse...

Build Options

☑ Personal Build

Check this optic using a workspa definition status you, not other

Repository worksp

Component load r

After submitting the

Select a Stream or Workspace

Enter stream, workspace, or owner name prefix or pattern (*, ?):

Matching items:

build's Squawk Team Stream Workspace (build)

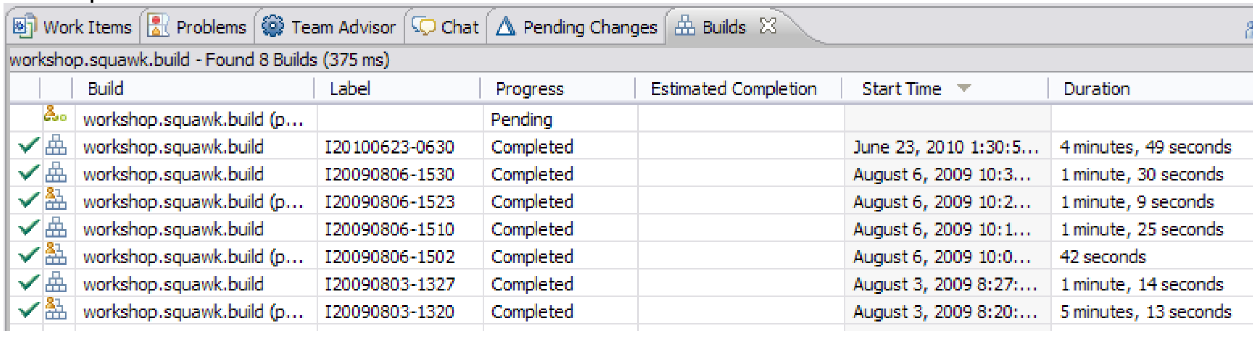
workshop.squawk.build_I20100623-0630 (loaded) (Scott)

Show: ○ Streams ● Workspaces ○ Both

OK Cancel

__d. In the **Request Build** dialog select **Submit** to submit the build.

__e. RTC opens a view to follow the Build

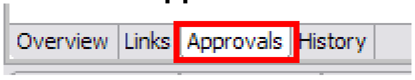


workshop.squawk.build - Found 8 Builds (375 ms)

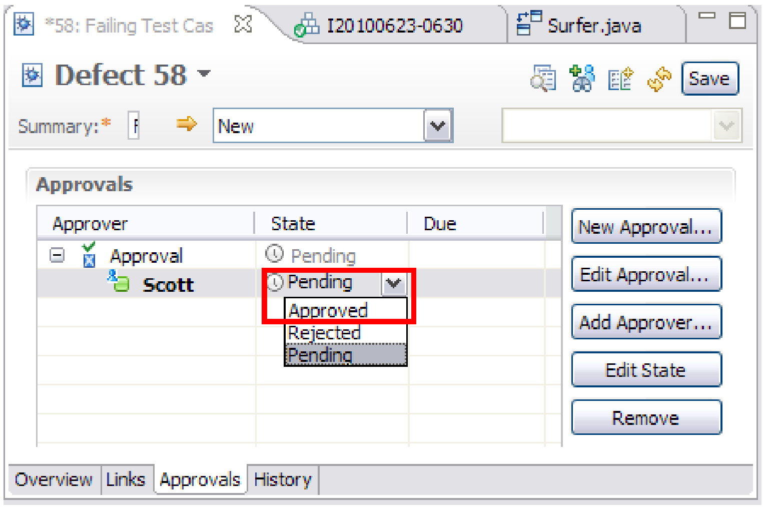
	Build	Label	Progress	Estimated Completion	Start Time	Duration
	workshop.squawk.build (p...		Pending			
✓	workshop.squawk.build	I20100623-0630	Completed		June 23, 2010 1:30:5...	4 minutes, 49 seconds
✓	workshop.squawk.build	I20090806-1530	Completed		August 6, 2009 10:3...	1 minute, 30 seconds
✓	workshop.squawk.build (p...	I20090806-1523	Completed		August 6, 2009 10:2...	1 minute, 9 seconds
✓	workshop.squawk.build	I20090806-1510	Completed		August 6, 2009 10:1...	1 minute, 25 seconds
✓	workshop.squawk.build (p...	I20090806-1502	Completed		August 6, 2009 10:0...	42 seconds
✓	workshop.squawk.build	I20090803-1327	Completed		August 3, 2009 8:27:...	1 minute, 14 seconds
✓	workshop.squawk.build (p...	I20090803-1320	Completed		August 3, 2009 8:20:...	5 minutes, 13 seconds

__11. While the build is running on a server, approve the change, to save some time. Navigate back to the Defect.

__a. Select the **Approvals** tab of the defect.



__b. Approve the approval request. Select the **Pending** state and select **Approved**. Save the defect.

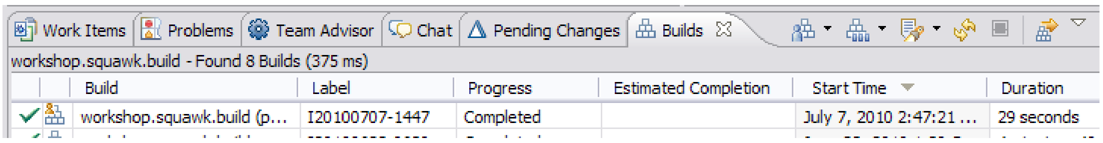


Approver	State	Due
Approval	Pending	
Scott	Pending	

Deb now gets several indicators that the change is approved and can continue and deliver the change.

__12. Meanwhile the private Build should have finished successfully. Click the **Refresh** icon at the top of the view to update. Again this happens automatically but could take a minute or two.

__a. Check the status in the **Builds** view in the lower area of the client.



workshop.squawk.build - Found 8 Builds (375 ms)

	Build	Label	Progress	Estimated Completion	Start Time	Duration
✓	workshop.squawk.build (p...	I20100707-1447	Completed		July 7, 2010 2:47:21 ...	29 seconds

__b. Open the Build result to browse the details.

3.4 Summary

Rational Team Concert supports simple or complex co-located or distributed development with a tightly integrated development environment. It supports the most common development activities planning, software configuration management, change and work management, continuous integration, build and unit test in one tool. All the artifacts of these activities are logically linked together, automatically wherever possible, using links. The user interface is optimized for typical development tasks and allows, in combination with the artifact linking very easy, efficient access to data required for those tasks. The web accessibility also adds other collaboration capabilities such as accessing artifacts in chats and e-mails.

Rational Team Concert closes the gap between development activities and artifacts and supports the users to concentrate on the real challenges in development. This makes users more efficient and productive. The tight integration allows support for process automation – out of the box or customized – that support the user in following the process and not getting into their way.

Table of contents

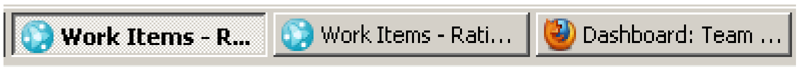
Appendix A. Optional Labs

Lab 4 Optional: Analyze Test results in Quality Manager

__1. Locate and prepare Eclipse Clients.

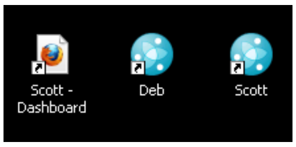
To work with Code Rational Team Concert provides a development client. Rational Team Concert's development client can run based on Eclipse or Microsoft Visual Studio.

__a. To save some time with the limited resources of the lab machines, running servers and clients the Eclipse clients should be started in the beginning. The Lab instructor should have started the clients already. If this is the case you should be able to locate them in the taskbar:



__b. If they can be located skip to step __3 otherwise continue with step __2

__2. Start the Eclipse Client for **Deb** using the provided Link

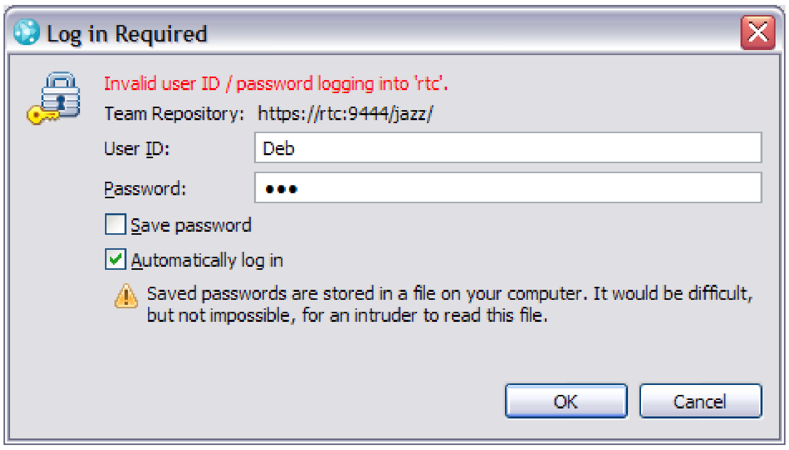
__a. Locate the quick-start links 

on the desktop.

__3. Double-click on the provided link for **Deb** and wait for the log-in screen to appear.

__4. In the log-in Screen for Deb's workspace

__a. Enter `Deb` for the **Password**, and then press **OK**. The connection process will take some time.


Log in Required

Invalid user ID / password logging into 'rtc'.

Team Repository: https://rtc:9444/jazz/

User ID: Deb

Password: •••

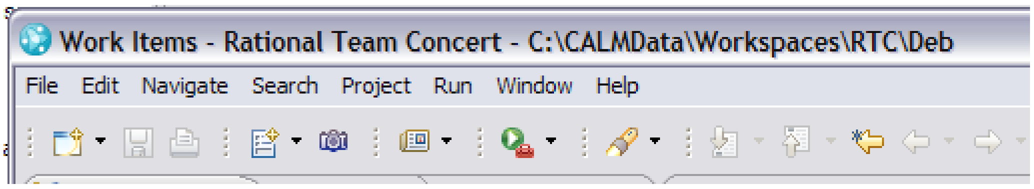
☐ Save password

☑ Automatically log in

Saved passwords are stored in a file on your computer. It would be difficult, but not impossible, for an intruder to read this file.

OK Cancel

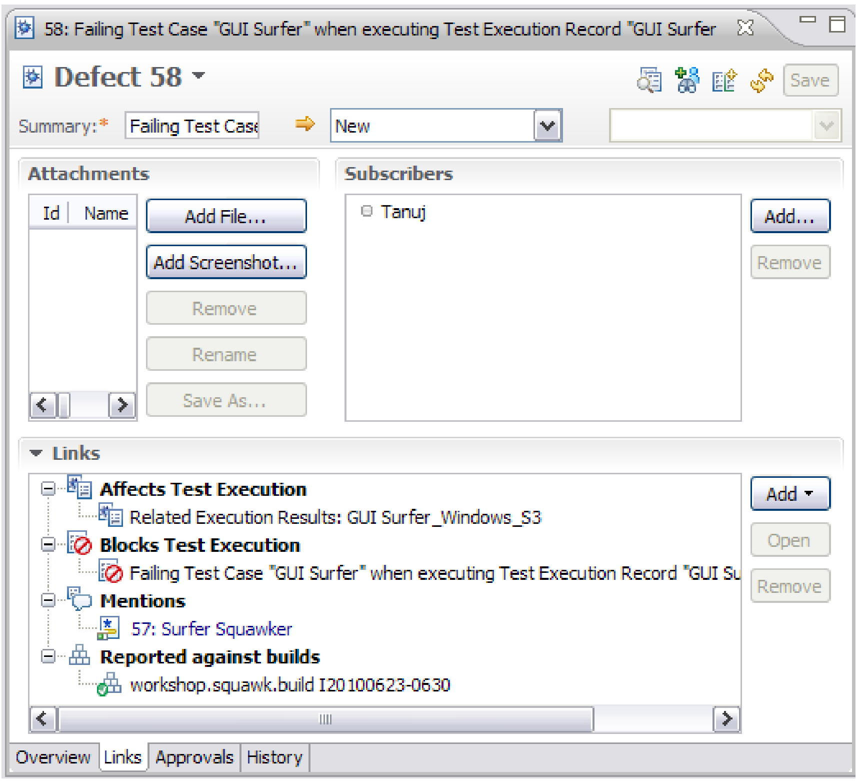
__5. Open Deb's RTC Client. Verify the headline shows **Work items – ….. – C:\CALMData\Workspaces\RTC\Deb**.


Work Items - Rational Team Concert - C:\CALMData\Workspaces\RTC\Deb

__6. Deb needs to understand what happened and can get the details stored in **Rational Quality Manager**.

__a. In the **My Work** view, double-click the defect to open the work item editor.

__b. Read the defect description and then click the **Links** tab. In the links section RTC provides all other artifacts this defect is related to. In the links section you can see that the defect is related to a test execution result that failed. Because of this failure the defect blocks further tests of the test case linked below. The test execution record and the test case actually belong to a related test project in **Rational Quality Manager**. The data is stored on a server and can be directly accessed from here.


58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer

Defect 58

Summary:* Failing Test Case New

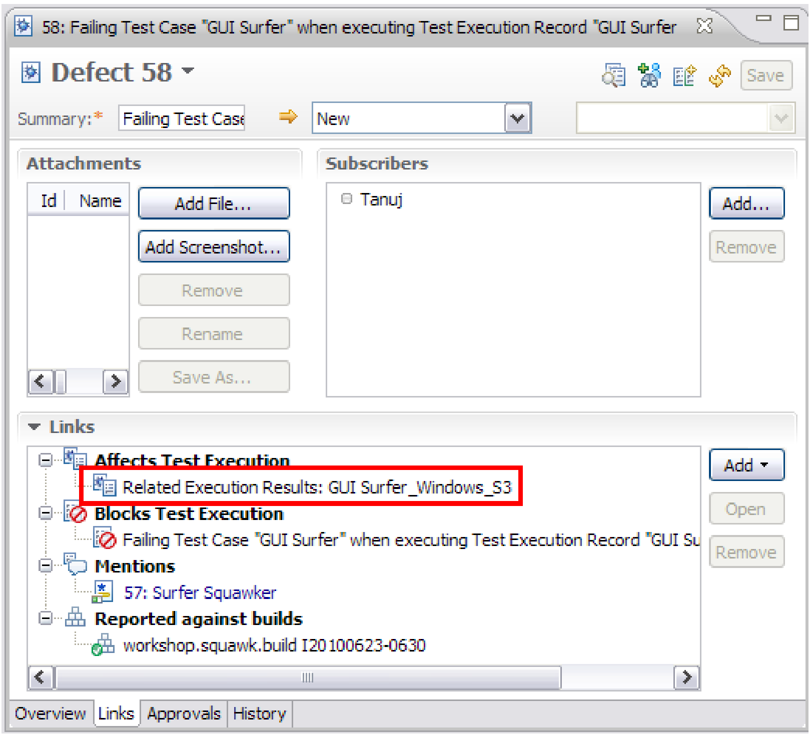
Attachments: Id | Name

Subscribers: Tanuj

Links

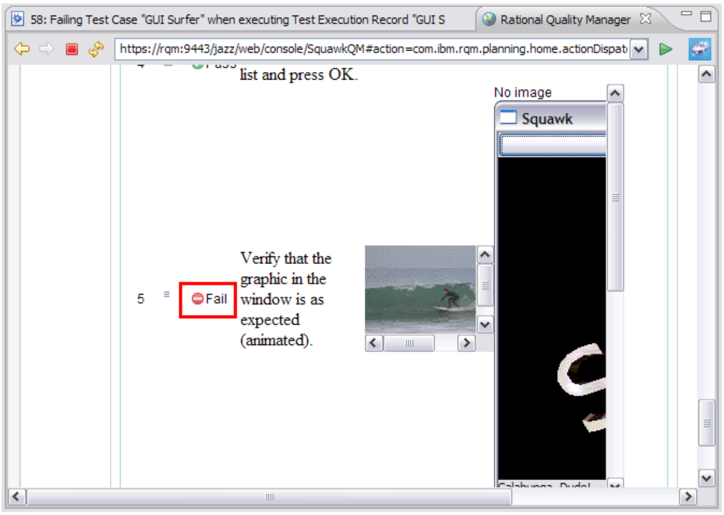
- Affects Test Execution
  - Related Execution Results: GUI Surfer_Windows_S3
- Blocks Test Execution
  - Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Su
- Mentions
  - 57: Surfer Squawker
- Reported against builds
  - workshop.squawk.build I20100623-0630

__c. Double-click the link to the affected test execution results record (**Related Execution Results: GUI Surfer_Windows**). The result will be displayed in a new Rational Team Concert tab showing the Rational Quality Manager Web UI. (Select **Yes** if you receive a Security Alert and log-in).


58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer
Defect 58
Save
Summary:* Failing Test Case
New
Attachments
Id Name
Add File...
Add Screenshot...
Remove
Rename
Save As...
Subscribers
Tanuj
Add...
Remove
Links
Affects Test Execution
Related Execution Results: GUI Surfer_Windows_S3
Blocks Test Execution
Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Su
Mentions
57: Surfer Squawker
Reported against builds
workshop.squawk.build I20100623-0630
Add
Open
Remove
Overview Links Approvals History

__d. If prompted to **Log in**, enter `Deb` for both the **User ID** and **Password**.

__e. The Execution Result will appear in Deb's workspace. Note that it may take several seconds to load, depending on the hardware resources available. In the Execution Result scroll down to step 5 to see the bad actual result.


58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI S
Rational Quality Manager
https://rqm:9443/jazz/web/console/SquawkQM#action=com.ibm.rqm.planning.home.actionDispat
list and press OK.
No image
Squawk
5
Fail
Verify that the graphic in the window is as expected (animated).

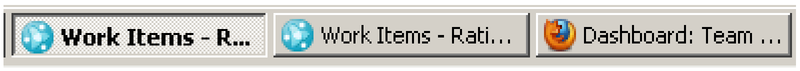
__7. This concludes the optional lab.

Lab 5 Optional: Pass information in instant messaging

__1. Locate and prepare Eclipse Clients.

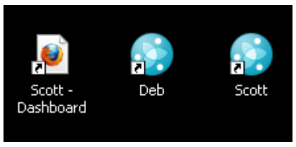
To work with Code Rational Team Concert provides a development client. Rational Team Concert's development client can run based on Eclipse or Microsoft Visual Studio.

__a. To save some time with the limited resources of the lab machines, running servers and clients the Eclipse clients should be started in the beginning. The Lab instructor should have started the clients already. If this is the case you should be able to locate them in the taskbar:

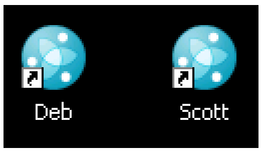


__b. If they can be located skip to step __3 otherwise continue with step __2

__2. Start the Eclipse Client for **Deb** and **Scott** using the provided Links

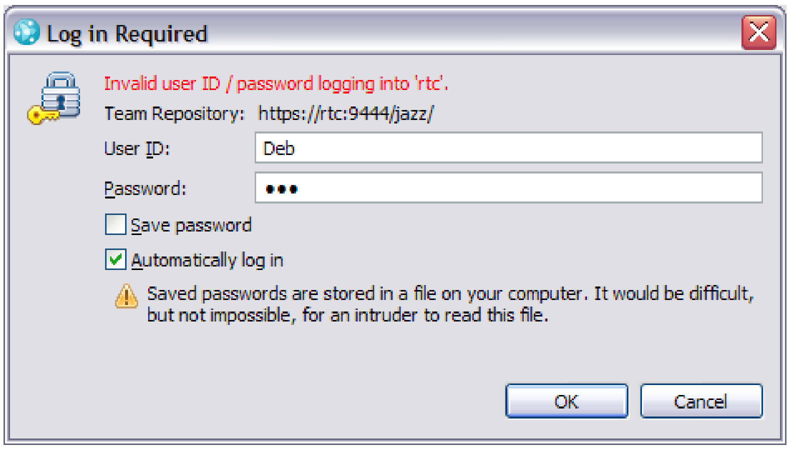
__a. Locate the quick-start links  on the desktop.

__b. Double-click on the provided links **Deb** and **Scott**



__3. In the Log in Screen for Deb's workspace

__a. Enter `Deb` for the **Password**, and then press **OK**. The connection process will take some time.



Log in Required

Invalid user ID / password logging into 'rtc'.

Team Repository: https://rtc:9444/jazz/

User ID: Deb

Password: •••

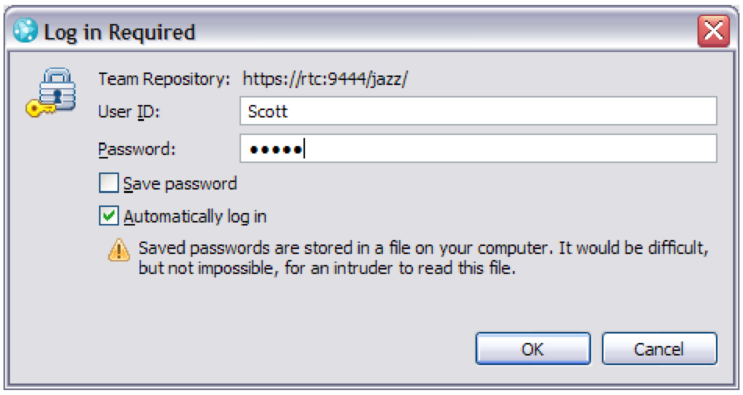
☐ Save password

☑ Automatically log in

Saved passwords are stored in a file on your computer. It would be difficult, but not impossible, for an intruder to read this file.

__4. In the Log in Screen for Scotts workspace

__a. Enter `Scott` for the **Password**, and then press **OK**. The connection process will take some time.

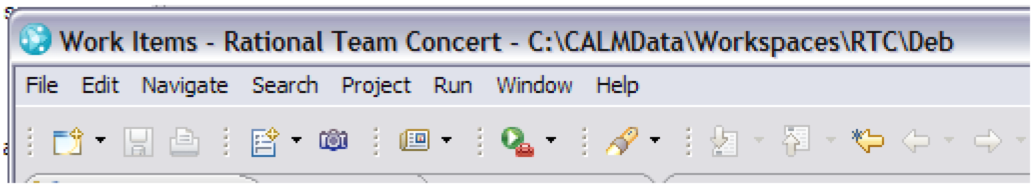


Log in Required

Team Repository: https://rtc:9444/jazz/
User ID: Scott
Password: •••••
[ ] Save password
[✓] Automatically log in
Saved passwords are stored in a file on your computer. It would be difficult, but not impossible, for an intruder to read this file.

OK Cancel

__5. Open Deb's RTC Client. Verify the headline shows **Work items – ….. – C:\CALMData\Workspaces\RTC\Deb**.

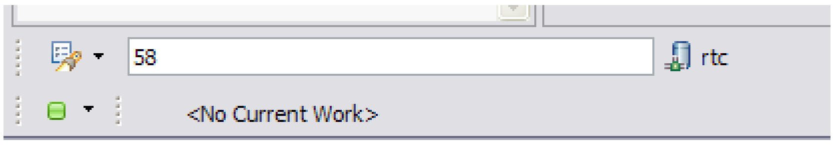


Work Items - Rational Team Concert - C:\CALMData\Workspaces\RTC\Deb

__6. Open the new Defect **58**

__a. In the **My Work** view, double-click the defect to open the work item editor.

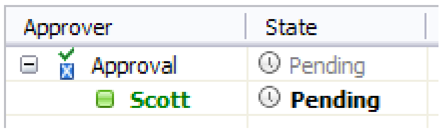
__b. You can search for the defect in the quick search on the lower left corner of your RTC Client. Enter **58** and press return. The work item editor opens up automatically.



__7. Read the defect description and then click the **Approvals** tab. There should be an approval request you created. If not create one and add Scott as approver.

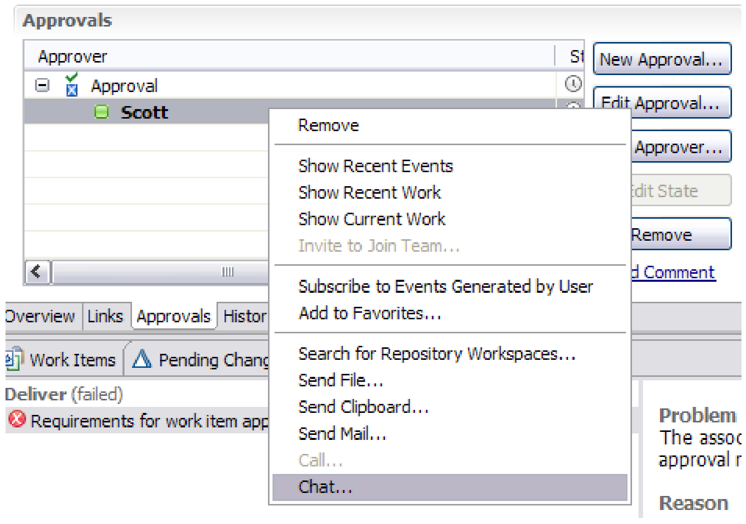
__a. We have to wait for the approval to deliver. We can suspend the change and continue working on something else. Let's try if we can speed up the process a bit.

__b. Verify that the online status of Scott is green, showing 🟩 Scott .

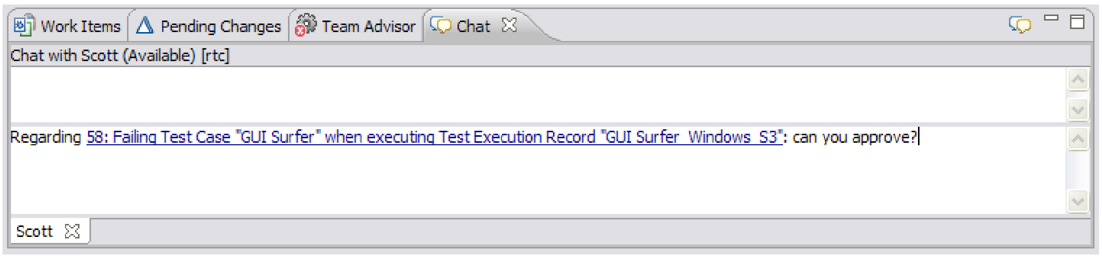


Approver	State
⊟ Approval	Pending
🟩 Scott	**Pending**

__c. Right click **Scott** at the approval and select Chat.



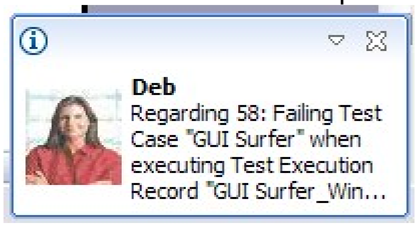
__8. In the chat window type **"can you approve?"** behind the automatically create link to the work item 58 and press **Enter**.



Chat with Scott (Available) [rtc]

Regarding 58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3": can you approve?

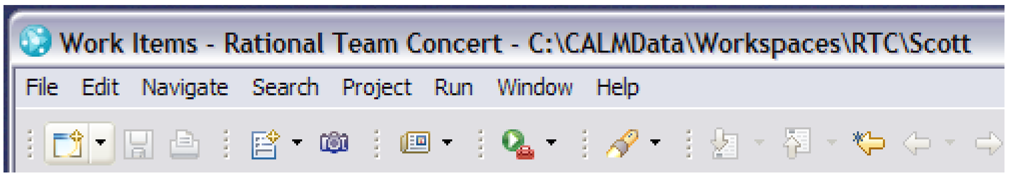
__9. If Scott is logged into his RTC Eclipse Client and Deb chats he sees the fly-out message below and can click it to open the chat in his Eclipse client.



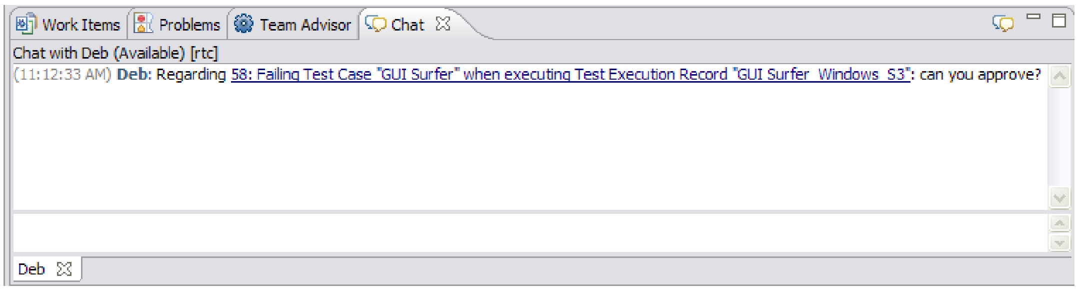
Deb
Regarding 58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Win...

__10. If the client did not open automatically open Scotts RTC Client.

__11. Verify the headline shows **Work items – ….. – C:\CALMData\Workspaces\RTC\Scott**.



Work Items - Rational Team Concert - C:\CALMData\Workspaces\RTC\Scott __a. If Scott opens the chat window he can click the link automatically provided in the chat to open the work item.



Chat with Deb (Available) [rtc]

(11:12:33 AM) **Deb:** Regarding 58: Failing Test Case "GUI Surfer" when executing Test Execution Record "GUI Surfer_Windows_S3": can you approve?

__b. Depending on the context RTC provides easy ways to display and access to referenced artifacts.

__12. While waiting for the approval Deb could suspend the change again and work on something else.

__13. You can chat from any place a team member name is presented for instance from Teams, from work item owners etc. Dependent on the context the Team Concert client creates an artifact link automatically. You can also drag and drop artifacts on chats to create links. For work items you can type the type and the number, for example **defect 58**, in a chat and in other places and a link will automatically be created.

__14. This concludes the lab.