

How to use the Scrum project management method with IBM Rational Team Concert and the Jazz platform

Jazz combines agile development with an IDE for frictionless planning, work, and management

Skill Level: Intermediate

Millard Ellingsworth (millarde@us.ibm.com)
Software Developer
IBM Corporation

30 Jun 2008

The integration of an Agile planning tool with IBM® Rational® Team Concert and Jazz provides greater visibility, opportunities for collaboration, and traceability as well as process awareness into a single platform for improving development productivity.

According to IBM Fellow Grady Booch, IBM® Rational® Team Concert and the IBM® Rational® Jazz platform are poised to bring to fruition a "frictionless surface for development [by] eliminating or automating many of the daily activities of the team." One of the many friction points alleviated by the release of Rational Team Concert and its Scrum Process template is the integration of agile planning and tracking with development work. Until now, if you were using a popular agile planning tool, you had to bounce between the tool and your IDE (integrated development environment) to locate the work to perform, complete it, and report on its progress. By integrating these and other activities, Rational Team Concert and Jazz provide greater visibility, opportunities for collaboration, and traceability, as well as process awareness, all combined in a single platform for improving development productivity and software delivery.

The term **scrum** originated in the game of rugby as short for *scrummage* or *scrimmage*. In the context of software development, Scrum is a method of project management in an agile development process to keep the focus on delivering the

highest business value to stakeholders in the shortest possible time. The Scrum process helps ensure that the most valuable remaining features are built next, and it emphasizes that work always be completed to the point of being deliverable. Work is structured in two- to four-week **sprints**, or iterations, with working software produced by the end of each sprint.

To learn how to use the Scrum Process template with Rational Team Concert, you'll work through setting up a project and following it through its first iteration. The data for the sample project is from the extended example in Mike Cohn's Agile Estimating and Planning (Prentice Hall, 2005, which describes a Scrum team at Bomb Shelter Studios working on a new computer game called Havannah. The data for the example is in the attached document, "Rational Team Concert Scrum sample data" (see [Download](#)).

This article assumes that you have registered at jazz.net and have downloaded and installed the software according to the instructions there. You will need to reach the point where your Rational Team Concert client can connect to your Jazz repository. After a quick introduction to the Scrum process, we'll pick up with the creation of a Rational Team Concert Scrum-based project.

Overview of the Scrum process

The Scrum framework consists of the following interrelated aspects:

- Roles
 - Product Owner
 - ScrumMaster
 - Team Member
- Ceremonies (gatherings or meetings)
 - Sprint Planning
 - Sprint Review
 - Sprint Retrospective
 - Daily Scrum
- Artifacts
 - Product Backlog
 - Sprint Backlog
 - Burndown Chart

- Impediments List

Requirements in a Scrum project are captured as items in the **Product Backlog**. This is one of the most important artifacts in a Scrum project. It is a prioritized list of all desired work or features for the project. The **Product Owner** is responsible for maintaining this list and revising priorities for items as the project progresses and the business environment changes. Ideally, the items in the Product Backlog are stated in terms that include an indication of their business values to the stakeholders.

The Product Owner is responsible for the success of the product. He or she defines the features and release schedule and is responsible for determining the business value of the various features. The Product Owner constantly refines and reprioritizes the Product Backlog.

The **ScrumMaster** manages the Scrum process, ensuring that the Scrum practices are properly executed and that Scrum values are understood by the team members. He or she ensures that the team is fully productive by eliminating impediments and shielding them from outside interference (at least during the course of a sprint).

A Scrum team should be a cross-functional mix of the right skills to complete the project (for example: developers, testers, interaction designers, writers). **Team members** should be committed to the Scrum team full-time.

The team holds a **Sprint Planning** meeting at the start of each sprint. At this meeting, the Product Owner presents the features most desired for the upcoming sprint, describing them so that the team can grasp what is expected. The Product Owner and team agree on a goal for the sprint -- essentially a vision statement focused in the work for the next two to four weeks. The team then determines how to accomplish the **Sprint Goal** and breaks the features down into the tasks required.

This list of tasks becomes the **Sprint Backlog**. The items in the Sprint Backlog are estimated in hours, allowing the team to determine if the items can be completed. If there is insufficient time, a feature can be taken out of the Sprint Backlog and returned to the Product Backlog. Estimation is done as a team exercise and is not determined by the ScrumMaster or Product Owner. The Sprint Backlog represents the work the team has committed to completing during the iteration.

During the **Daily Scrum** meeting, the team answers three questions:

- What did I do yesterday?
- What will I do today?
- What, if anything is blocking my progress?

This is not a status meeting, but a planning and commitment meeting. The whole thing should take less than 15 minutes. This regular communication about what

everyone is doing and what issues they are facing can eliminate the need for other meetings and help team members work more effectively together.

At the end of each sprint, the team shows off the work that it has accomplished during the **Sprint Review** meeting. This is a brief and informal meeting to which any and all interested parties are invited. Its purpose is to demonstrate the working software (or other valuable artifacts, such as an overview of the underlying product architecture) and to get feedback from participants and stakeholders on the working software.

After the Sprint Review, the team meets by itself to hold the **Sprint Retrospective** (sometimes called a Reflection). During this meeting, they will talk about what is going well -- and what isn't -- with respect to the project and their progress. This meeting should result in some change in how they work for the coming sprint as they work to continually improve their team and practice effectiveness.

As items from the Product Backlog are selected for completion, they become part of a Sprint Backlog. The Sprint Backlog contains specific tasks associated with the features from the Product Backlog. During the sprint, the status of the work items in the Sprint Backlog is updated daily. The updated status makes it possible to generate a **Sprint Burndown** chart. The burndown chart shows the amount of work remaining in the project in a graphical manner. It is also common for it to include an "idea" line that indicates a smooth progression of the work from start to finish of the sprint.

When there are impediments or blocks to progress, the ScrumMaster may maintain an **Impediments List** (or **Blocks List**) as an additional Scrum artifact. All Scrum artifacts should be kept where they are readily available to all team members.

The Jazz Scrum Process template supports all of the important Scrum roles, ceremonies, and artifacts. Rational Team Concert provides easy access to these features, and many others, through both an Eclipse™-based rich client user interface (UI) and a Web UI.

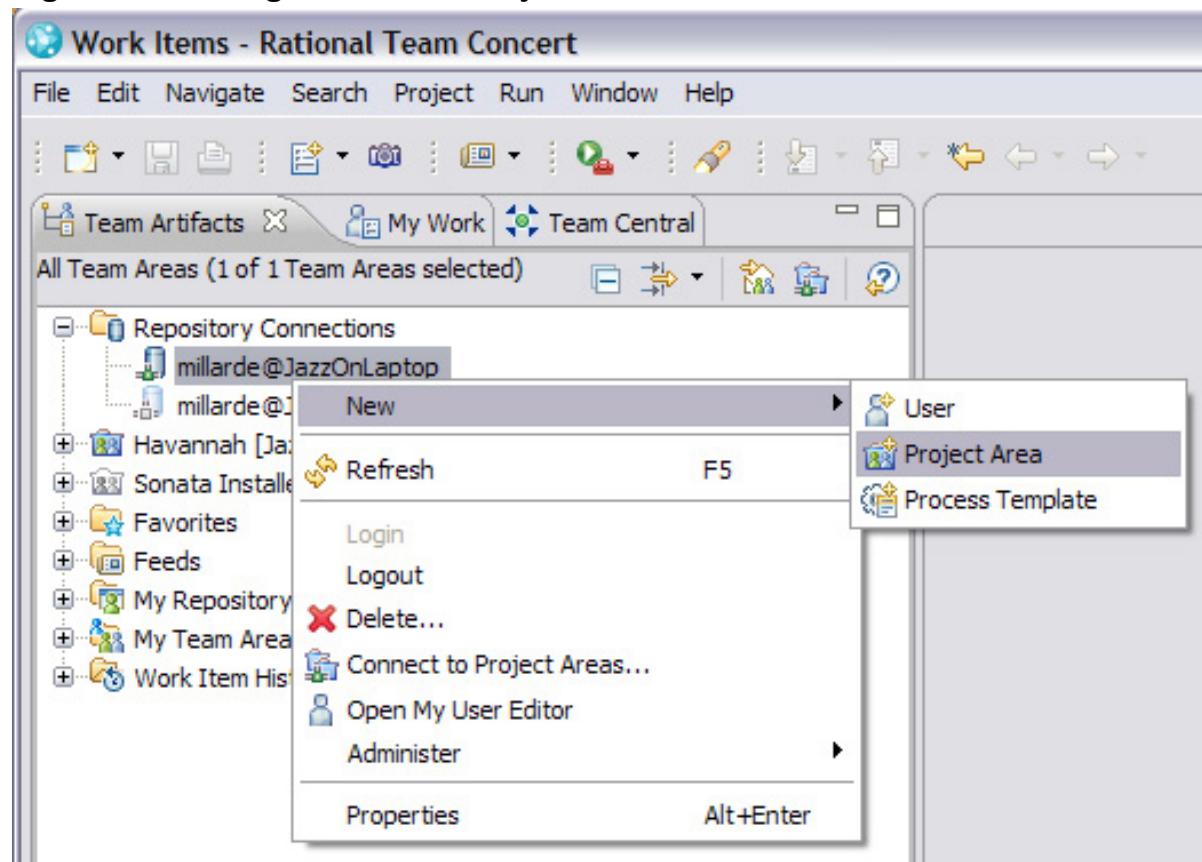
Set up the example project

Before we can look at how Rational Team Concert and the Jazz Scrum Process template support Scrum teams, we need to handle the basic project setup. We used Beta 3 of Rational Team Concert and Jazz plus an early access version of the Scrum Process template to create these screen captures, so they should look very much like the released product.

Create the Project Area

1. Be sure that your Jazz server is started, that your Rational Team Concert client is open, and that you have logged into the Repository Connection.
2. To create a new Project Area, right-click your **Repository Connection** in the Team Artifacts view (Figure 1) and then select **New > Project Area**.

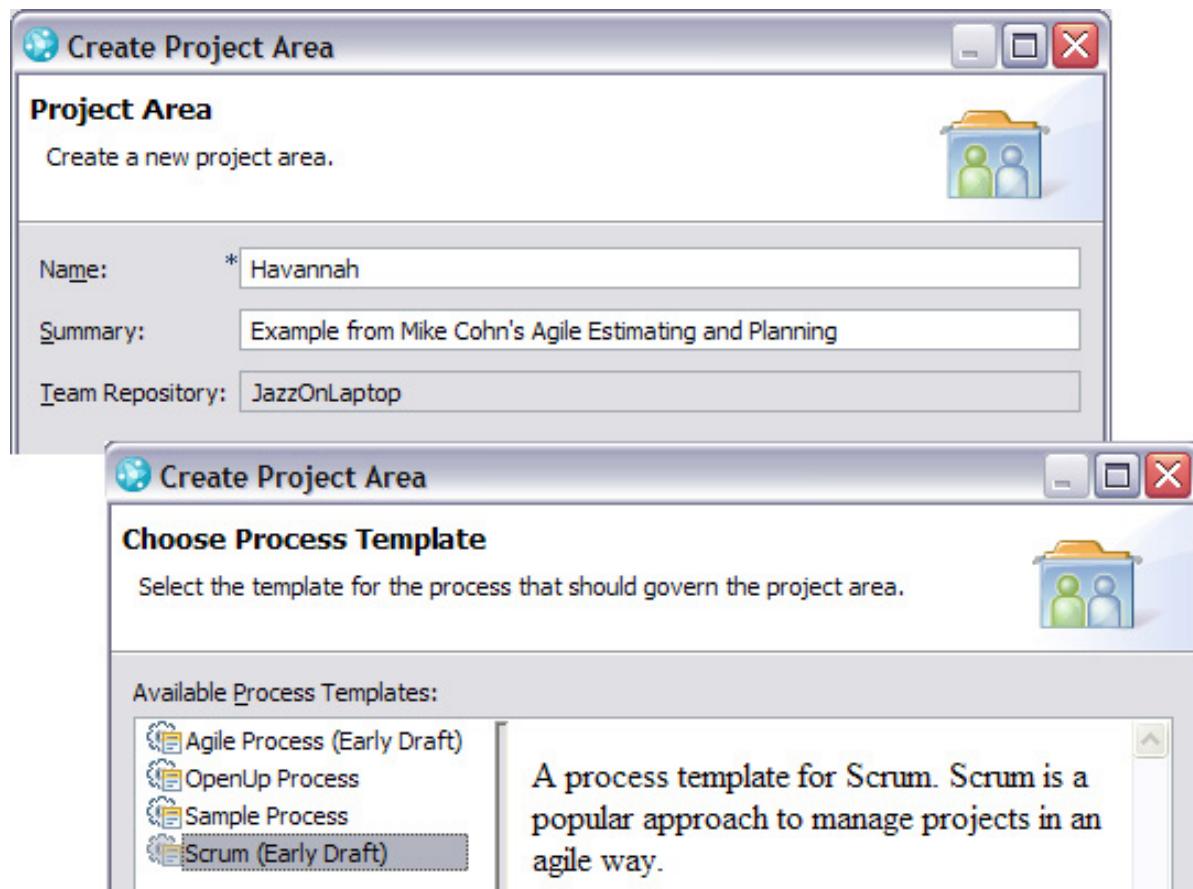
Figure 1. Creating a new Jazz Project Area



The new Project Area wizard has two panels (Figure 2).

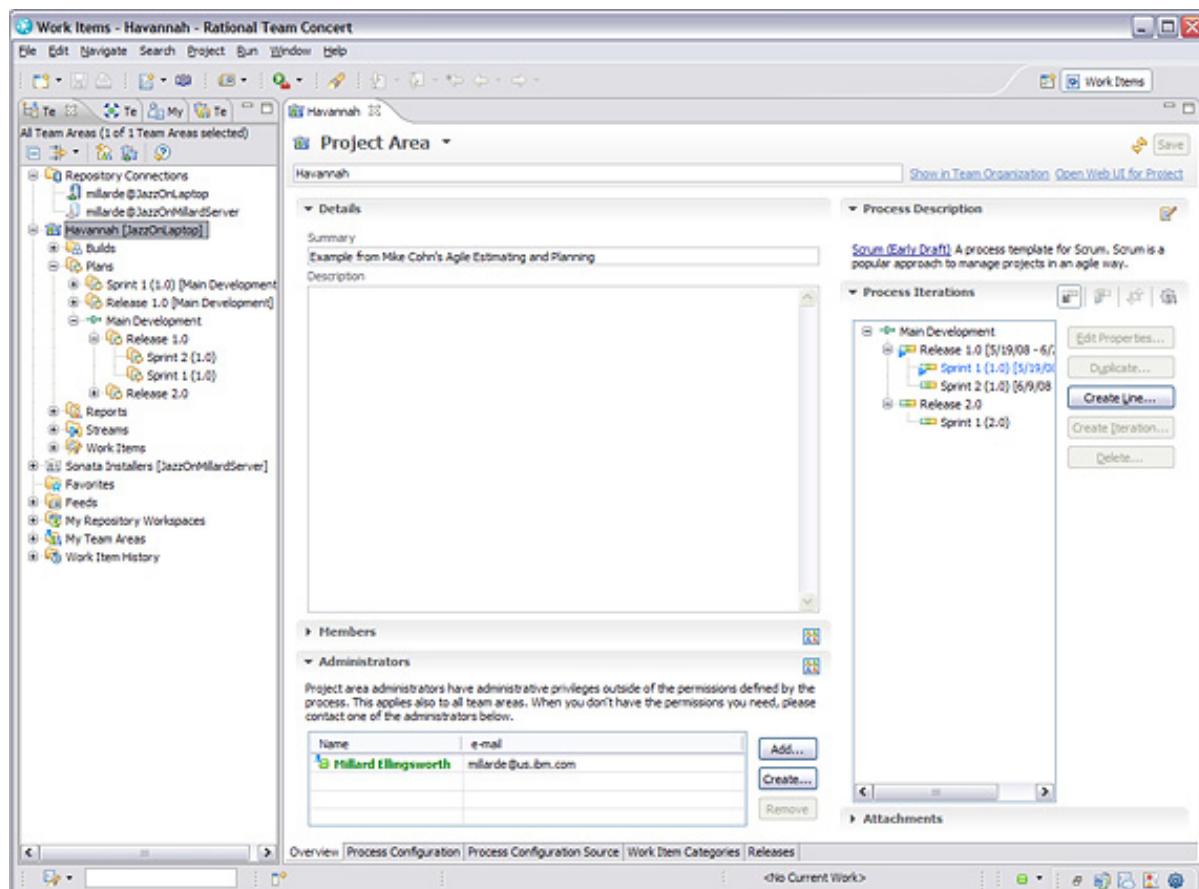
3. On the first, enter a name for the project (`Havannah` in this example), and click **Next**.
4. Under Available Process Templates, choose the **Scrum** template and click **Finish**. (In the released software, the Scrum Process template name will not include the "(Early Draft)" suffix.)

Figure 2. Create Project Area wizard steps



It can take up to several minutes for the Project Area to be initialized on the server. When that is complete, you will see a screen similar to the one in Figure 3.

Figure 3. Havannah project now shows in All Team Areas



In the left column in the Team Artifacts view, you can see the new project area with folders for Builds, Reports, Streams (source control), Work Items (queries), and Plans (the last is the area that this article focuses on). In the Project Area editor, there is an area for providing a description of the project. In the lower-right corner, you can see a section for adding attachments. This is intended to support the Scrum Process template and is **not** a good place to put your project attachments.

Add members and specify their roles

As the creator of the Project Area, you are the initial Administrator (Admin). However, most of the permissions for modifying the project structure within the Scrum Process template belong to the ScrumMaster or Product Owner. Therefore, one of the first things that you'll need to do is to add members to the project and set their roles (at least for the ScrumMaster and Product Owner). Typically, Project Area membership is for members with management responsibilities. Other team members will be added to a team area later.

1. Click the arrow to the left of Members to expand that panel (Figure 4).

If you configured Jazz with an external user repository (such as your corporate

LDAP server), you can import user entries without having to create them. However, let's assume that you used the default Tomcat application server and that you need to create users within the project area.

2. Click **Create** (Figure 4) to start the New Member wizard.

Figure 4. Members view

The screenshot shows a software interface titled "Members". On the left is a table with two columns: "Name" and "Process Roles". To the right of the table are four buttons: "Add...", "Create...", "Remove", and "Process Roles...". Above the table, there is a descriptive text block: "Roles determine a user's permissions as well as any preconditions and follow-up actions that are run for project and team operations. The roles assignments below are also valid in all the project's team areas. Unless configured otherwise, all users in the repository play the 'default' role."

3. Select **Create a new user** (Figure 5) and then click **Next**.

Figure 5. Create a new user

The screenshot shows a dialog box titled "Create or Import Users". It contains the text: "The server is configured with a directory service that allows new users to be created. Do you want to create a new user or import existing external users?". There are two radio buttons: one selected for "Create a new user" and one for "Import existing users".

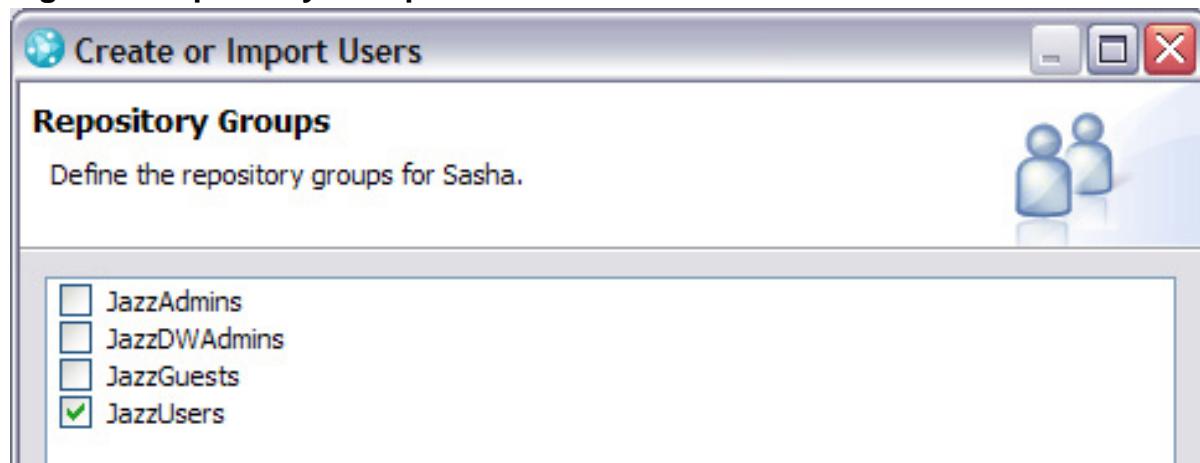
4. Enter the user's information, and add a photo if one is available (see Figure 6). The e-mail address is a required field because Jazz can send e-mail notification of many of the team's activities.

Figure 6. User Information view



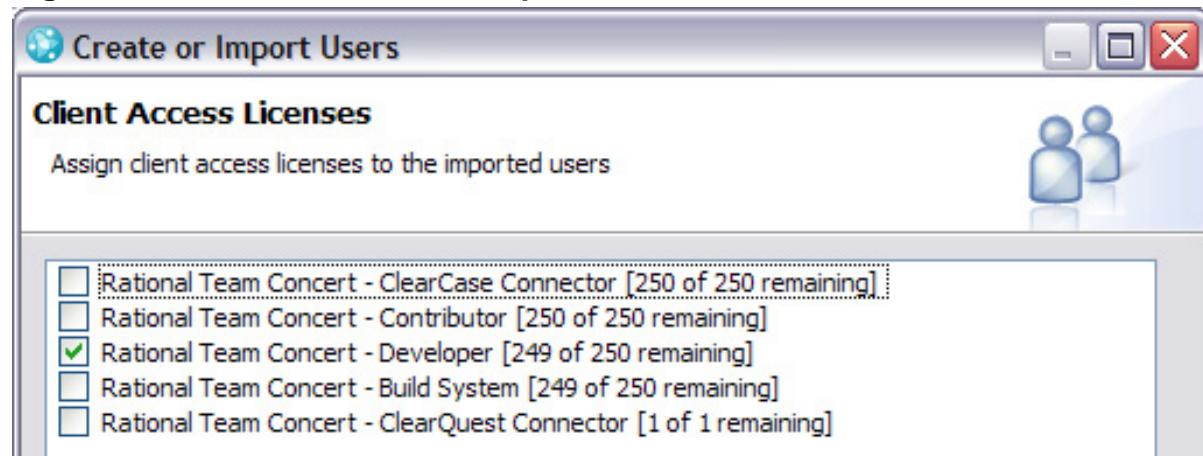
5. Click **Next**.
6. Choose the Jazz security groups appropriate for the user (see Figure 7).
 - **Jazz Users** have the default permissions in a Project Area. This typically allows them to create and add to Work items, to view reports, and to use other features defined by the project's security settings.
 - **Jazz Admins** can access the various settings related to the Jazz server and the project, such as creating additional Project Areas.

Figure 7. Repository Groups view



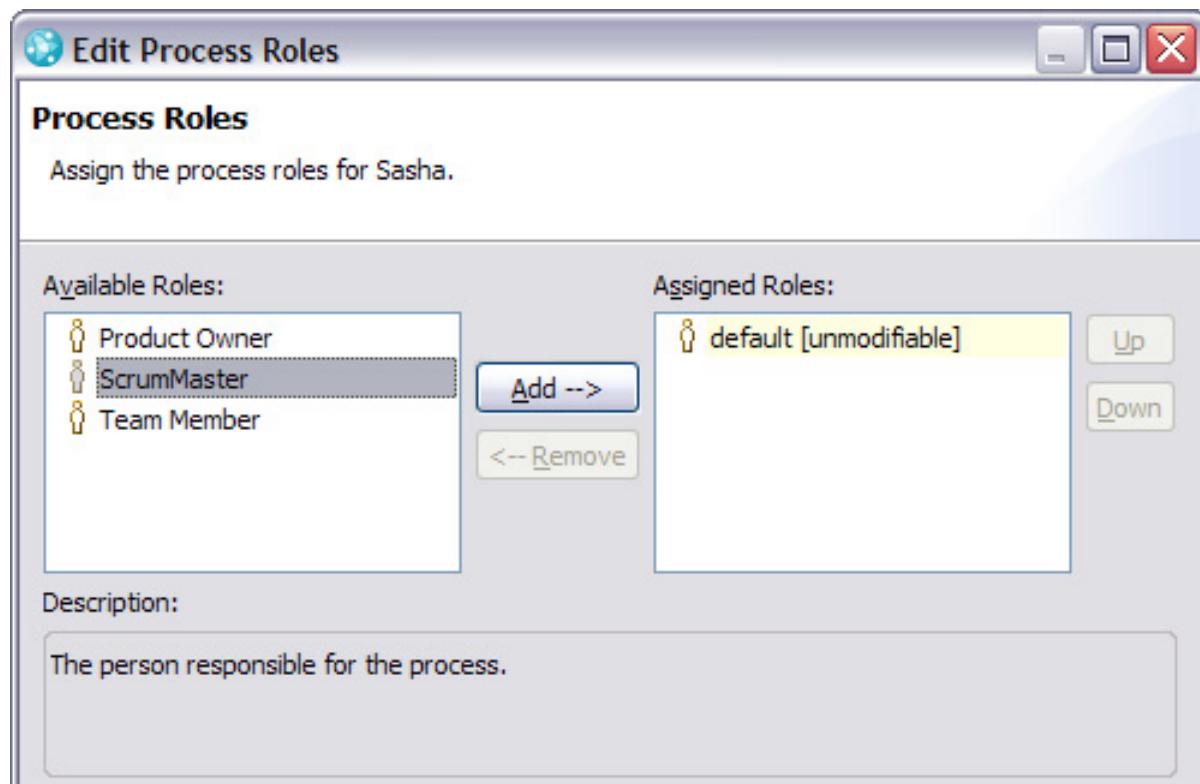
7. Click **Next**.
8. Select the appropriate type of license for the user (Figure 8).
 - **Developer** licenses are appropriate for team members who will be contributing code and running builds.
 - For example, **Build** and **ClearCase Connector** licenses are typically only assigned to administrative users.
 - **Contributor** licenses are a good choice for all other users.

Figure 8. Client Access License options



9. Click **Finish**.
10. Now that there is a member in the Members area, the Process Roles button (visible but previously disabled) is available when you select **Sasha** from the Members list.
11. Click Process Roles (Figure 9), select the ScrumMaster role, add it to the Assigned Roles list and click **Finish**.

Figure 9. Select and add roles



12. Add Frank as the Product Owner.

Figure 10 shows the completed Members list for the sample project, as well as the ScrumMaster in case the permissions diverge in the future.

Note:

ScrumMasters are not necessarily team members.

Figure 10. Completed Team Members list

▼ Members

Name	Process Roles
Frank	Product Owner
Sasha	ScrumMaster

Add...
Create...
Remove
Process Roles...

- To save the changes, click **Save** in the upper-right corner of the Havannah Project Area editor.

After saving your changes, you will be presented with a list of the new team members and asked if you want to send them an invitation by e-mail. If you have configured e-mail support (when you set up your server), this will send the project members a welcome e-mail message, along with links and information for connecting to the project.

- Because these users are fictitious, close this dialog window.

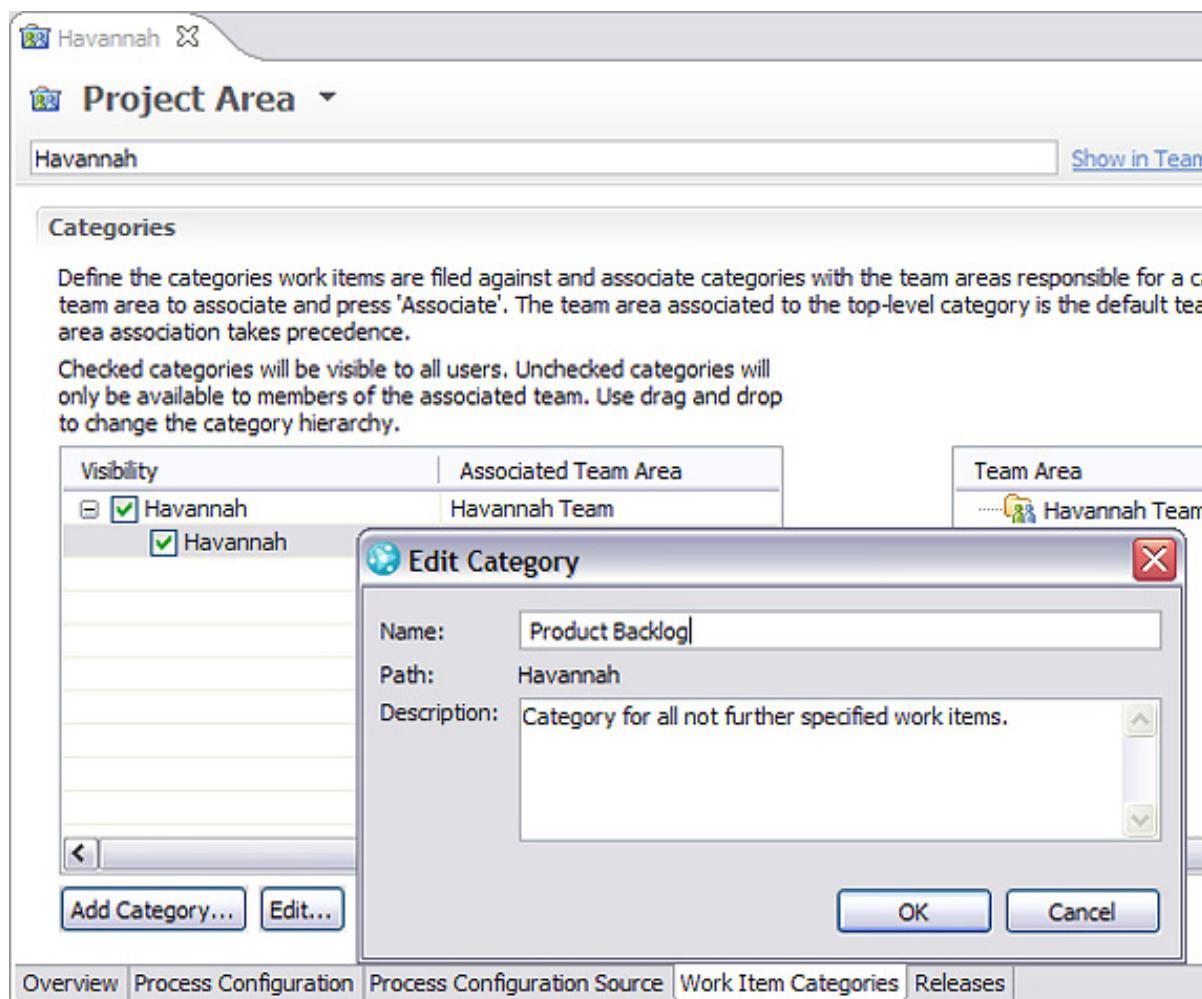
Change the default category name

There is one other setup item that is recommended for using the Scrum Process template, which you can do as an Administrator who is not a ScrumMaster or Product Owner.

- Change the name of the default category name for work items to **Product Backlog** (Figure 11) to be consistent with how Scrum manages features to be added to the product. (This step should not be necessary for the released product, however.)

Categories can also be used to organize features, tasks, defects, and so forth. For example, there could be categories for various UI components, different services, a different category for the Web UI -- whatever makes sense for the product organization. But for now, Product Backlog will do.

Figure 11. Change the category name to Product Backlog



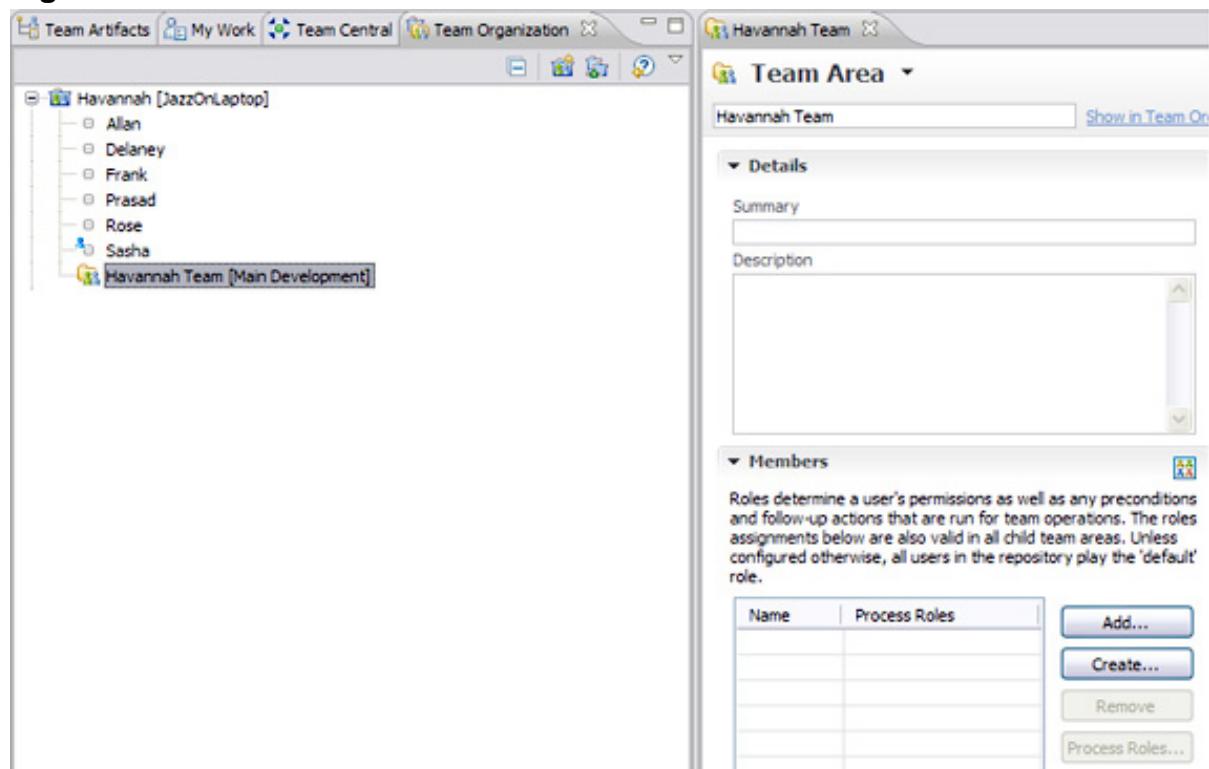
2. Using the tabs at the bottom of the Project Area editor, select **Work Item Categories** and then select the first category (only that one now) under the project name and edit it by selecting **Edit** at the bottom of the drop-down menu.
3. Change the name to Product Backlog, click **OK**, then save the Project Area changes.

Add members and their roles to the Team Area

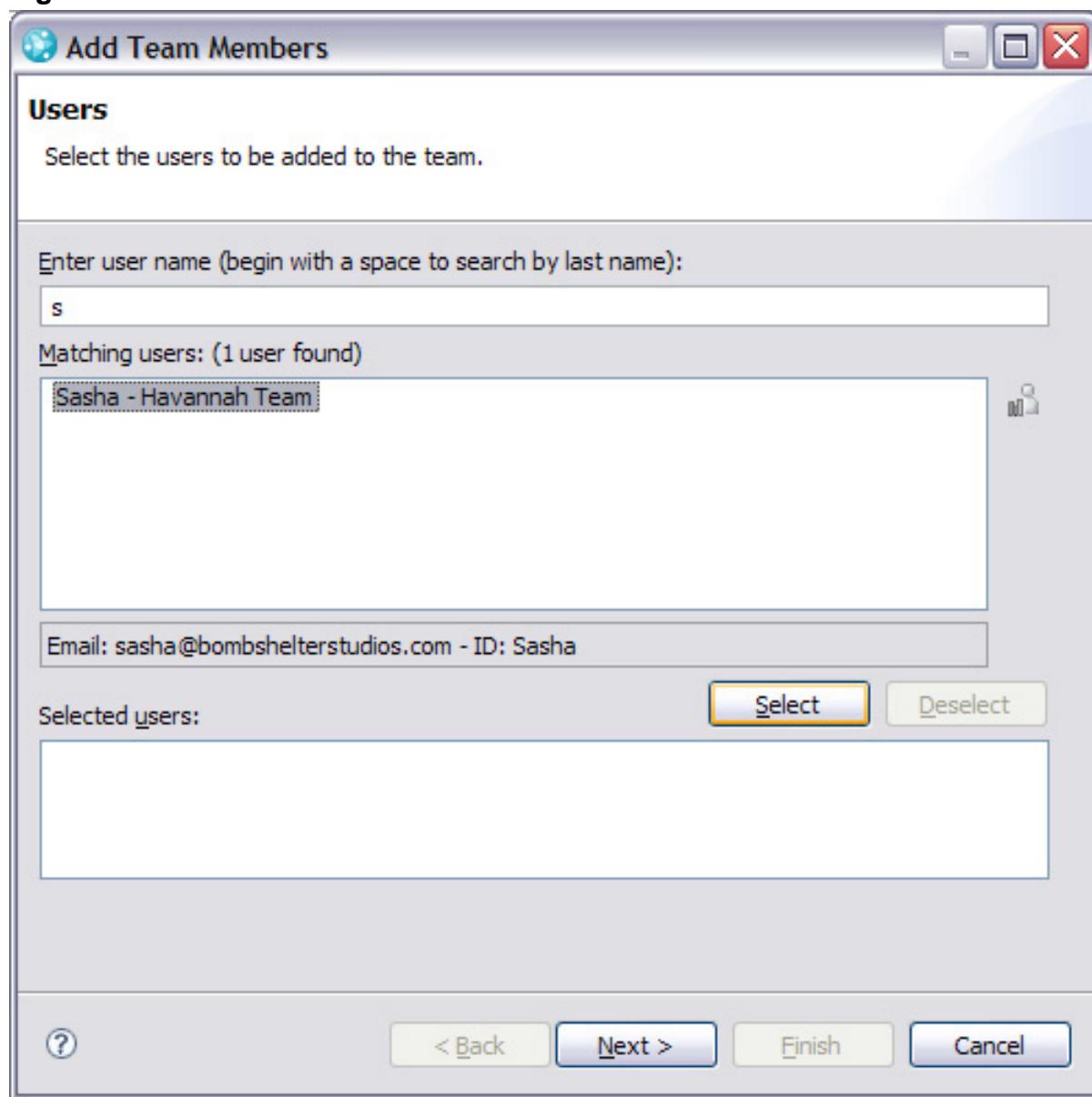
The Jazz model can support reasonably large teams with multiple development lines and subteams (Figure 12). Within Project Areas, there are Team Areas to organize the team members. Only Team Area members are eligible to be assigned work related to that team. Even though the example has just one team, it still needs to be populated with members.

1. Click the **Team Organization** tab in the left pane (Figure 12). If it isn't open, use **Window > Show View > Team Organization** to open it.
2. Double-click the **Hannah Team** folder (which was automatically created when the Project Area was created) to open the Team Area editor.
3. In the editor pane, click **Add** to launch the wizard for selecting Project Area members to add to this team.

Figure 12. Editor view for Hannah Team



4. Add Sasha as a Team Member and ScrumMaster:
 - A. Type an **s** (letter **S**) in the user name text area to search for matching names (Figure 13). Sasha will appear in the Matching Users list.
 - B. Click **Select** to move him to the Selected Users list.
 - C. Click **Next** to assign the ScrumMaster and Team Owner roles to him.
 - D. Click **Finish**.

Figure 13. Add team members view

5. The other users do not exist in the system yet. Use the **Create** button as you did with Frank and Sasha earlier to add the other team members as Team Members.

The final Members list should look like the one in Figure 14.

Figure 14. Updated Members list

▼ Members

四

Roles determine a user's permissions as well as any preconditions and follow-up actions that are run for project and team operations. The roles assignments below are also valid in all the project's team areas. Unless configured otherwise, all users in the repository play the 'default' role.

Name	Process Roles
Allan	Team Member
Delaney	Team Member
Prasad	Team Member
Rose	Team Member
Sasha	Team Member, ScrumMaster

Add...

[Create...](#)

[Remove](#)

Process Roles...

▼ Administrators



Project area administrators have administrative privileges outside of the permissions defined by the process. This applies also to all team areas. When you don't have the permissions you need, please contact one of the administrators below.

Name	e-mail
 Millard Ellingsworth	millarde@us.ibm.com
<input type="radio"/> Sasha	sasha@bombshelterstudios.com

Add...

[Create...](#)

[Remove](#)

- Also, open the **Administrators** section and add Sasha as a Project Area administrator. Later, if appropriate, the team's Project Area administrator can remove you as an administrator.

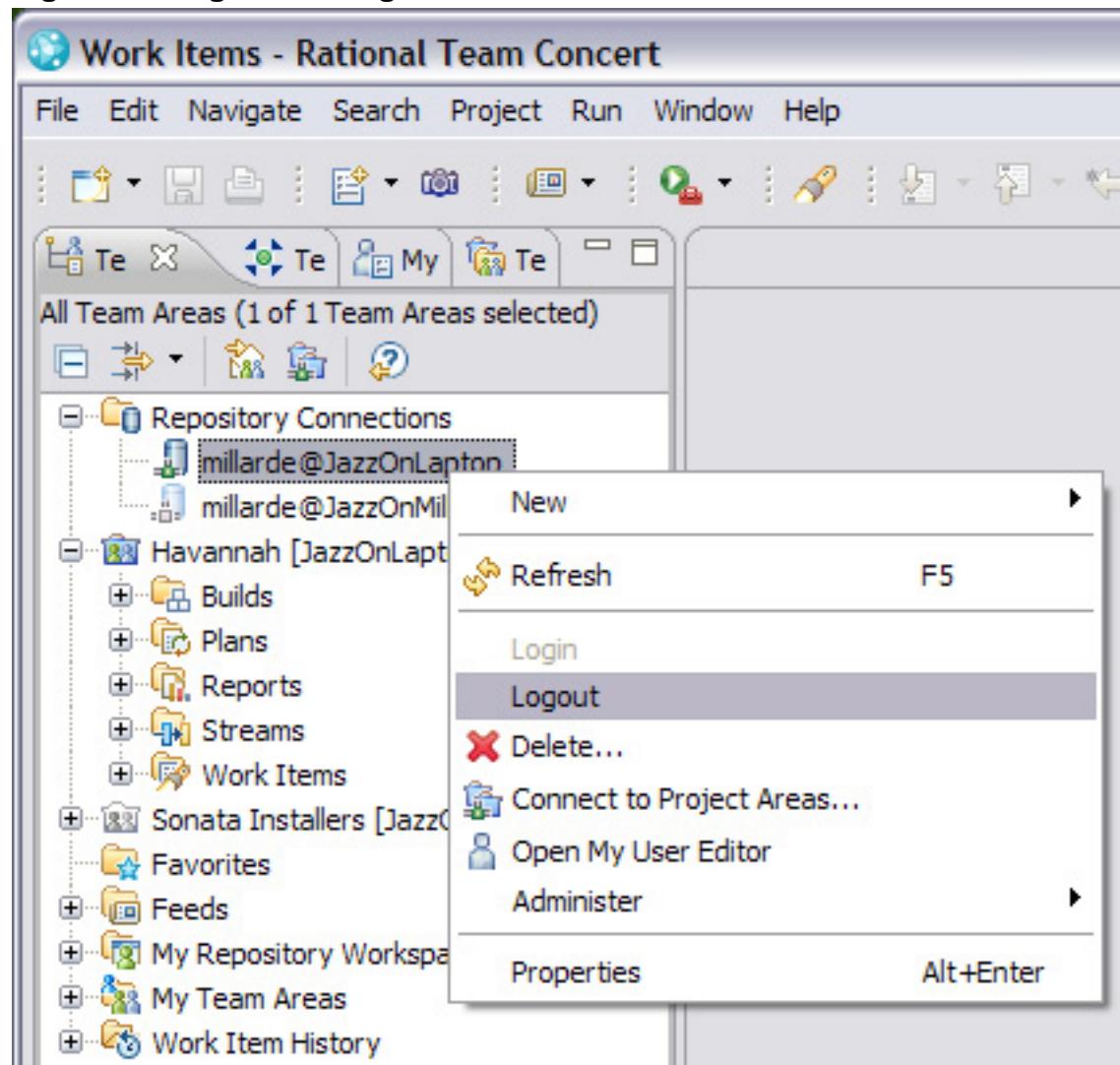
Establish iterations, called "sprints"

To start planning the actual project, you need to be logged in as the ScrumMaster or Product Owner, because these are the only roles that are assigned the privilege of changing plans. If you don't do this before attempting to modify the Release and Iteration plans, you will get a permission-denied error when attempting to save the changes. When setting up your own project (not this example one), if you are the ScrumMaster or Product Owner, it won't be necessary to follow the steps for

logging out and back in.

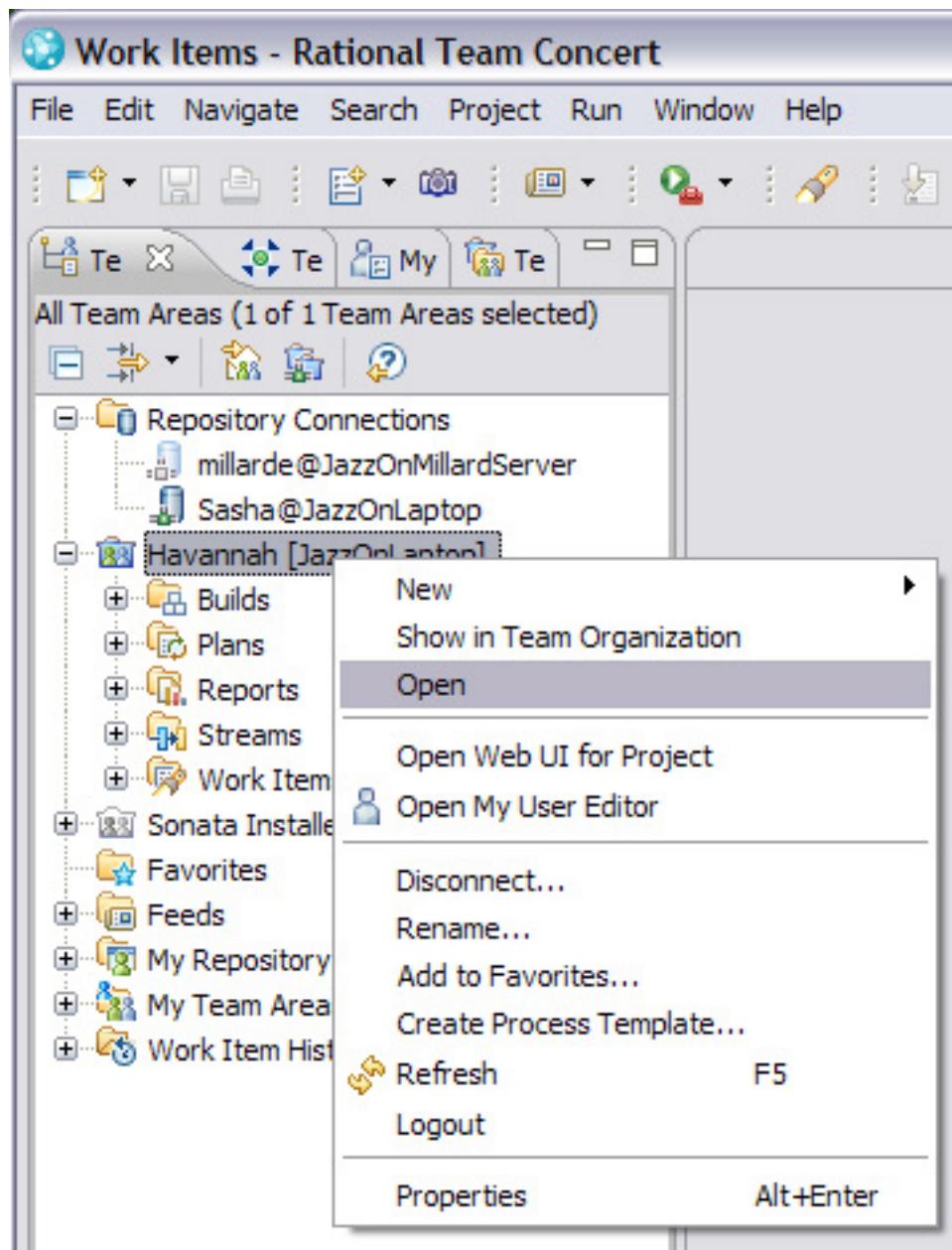
The drop-down menu from the Repository Connection (shown in Figure 15) contains the logout and login actions.

Figure 15. Logout and login as different user



1. Using the same drop-down menu that you used to log out, log in as Sasha (passwords default to the same value as the user ID, thus the password is also Sasha until it is changed). Because the Project Area is a top-level node, you'll need to use its drop-down menu (Figure 16) and select **Open** (double-clicking will only expand or contract it).

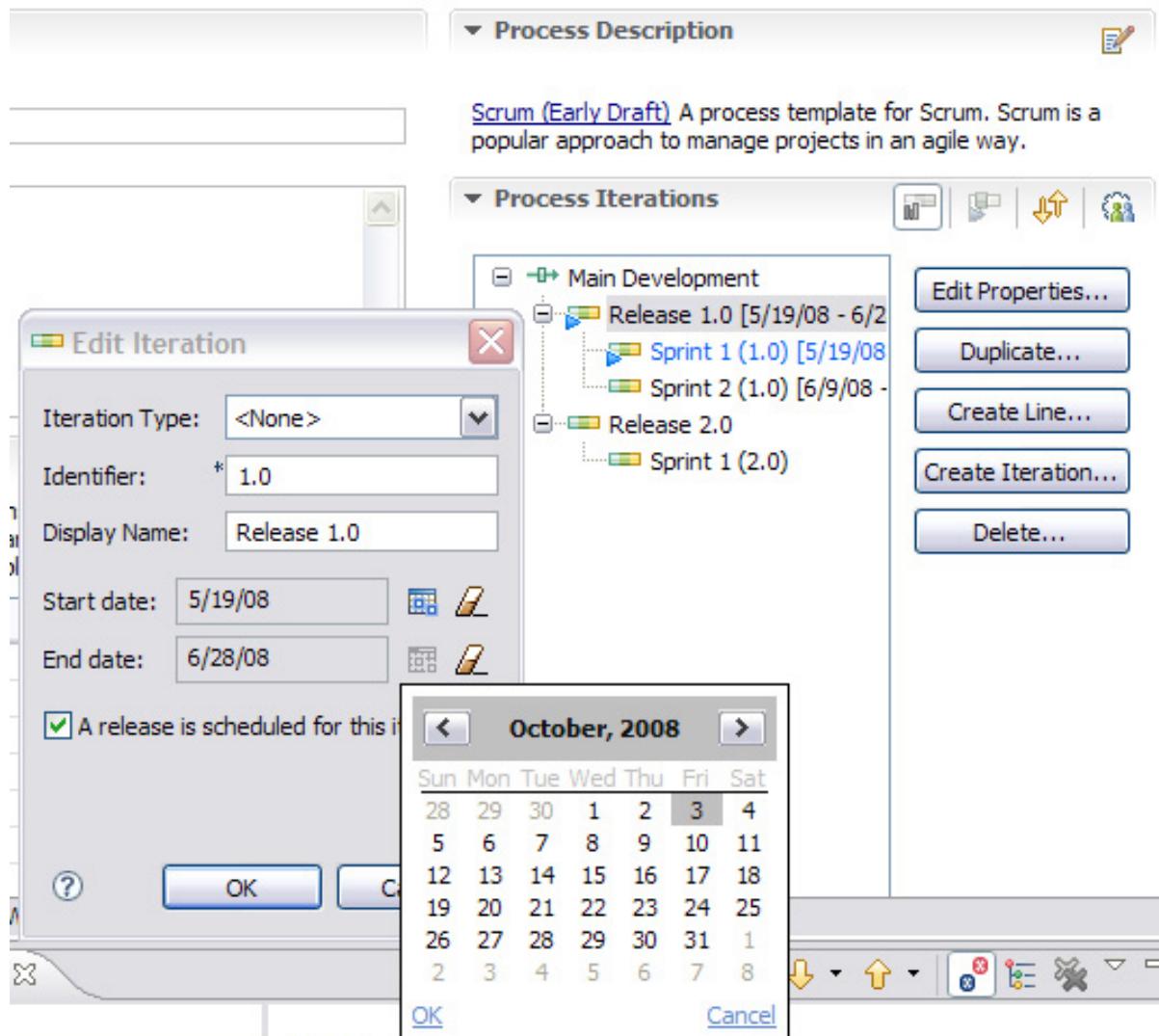
Figure 16. Opening the project area



When the Project Area editor is open again, you can see the default placeholder iterations that the process template established when you created the Project Area (Figure 17).

2. Select **Release 1.0** under Process Iterations, and click **Edit Properties**.

Figure 17. Editing release properties



3. The identifier must be unique, but you can change it to match your product.
4. Set the starting and ending dates for the release (see Note), make sure that the check box for "A release is scheduled for this iteration" is checked, and click **OK**.
5. You can edit the properties for the existing iterations similarly

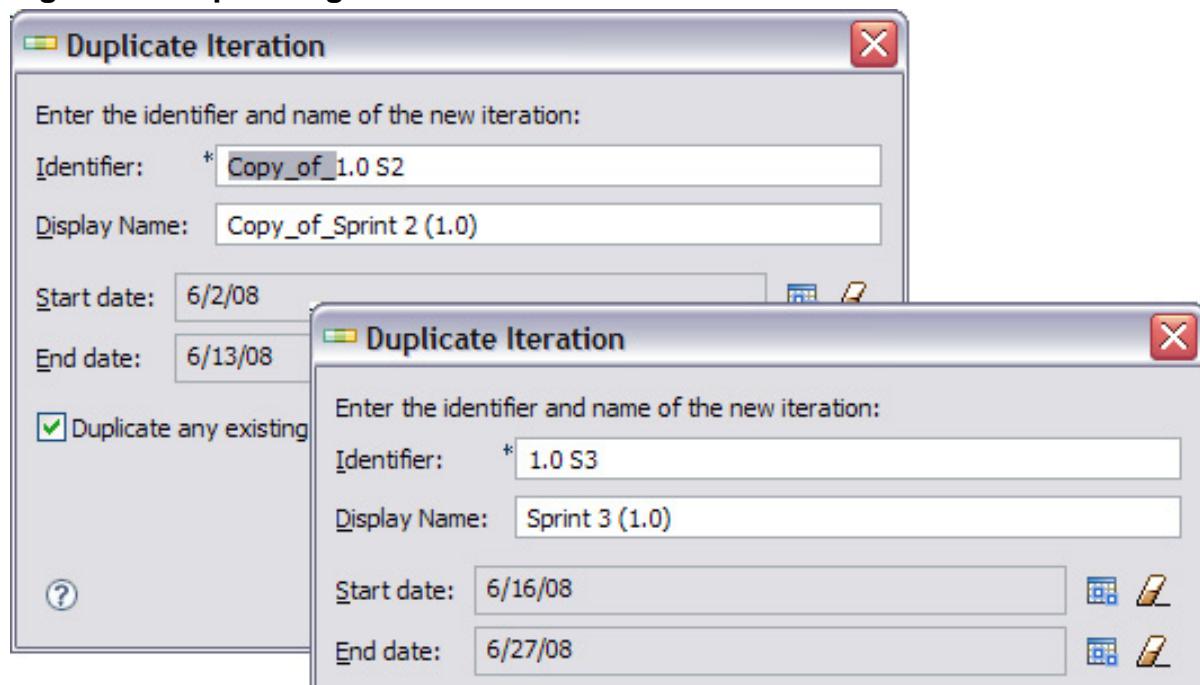
Note:

Because work tracking is time-sensitive, the dates are based on when you are working on this example. Start the release and first iteration today or some day in the near future. Set the release end date for at least six weeks later. Each iteration should last two weeks (typically, entered Monday to the following Friday, leaving out the non-work days on the ends).

The small blue triangle decorators on the Release 1.0 icon and Sprint 1 icon indicate that these are the current Release and Iteration.

6. To add an iteration, either click **Create Iteration** to add a new one, or choose an existing iteration and click **Duplicate** (Figure 18). To keep the identifiers and display names consistent, you can use the duplicate approach.
7. In this example, create a third iteration for the product: click on **Sprint 2** and then select **Duplicate**.

Figure 18. Duplicating an iteration



8. Remove the "Copy_of_" text and change the appropriate sections to name the new iteration.
9. Adjust the dates and click **OK**.
10. Save the Project Area changes.

Create the Product Backlog record

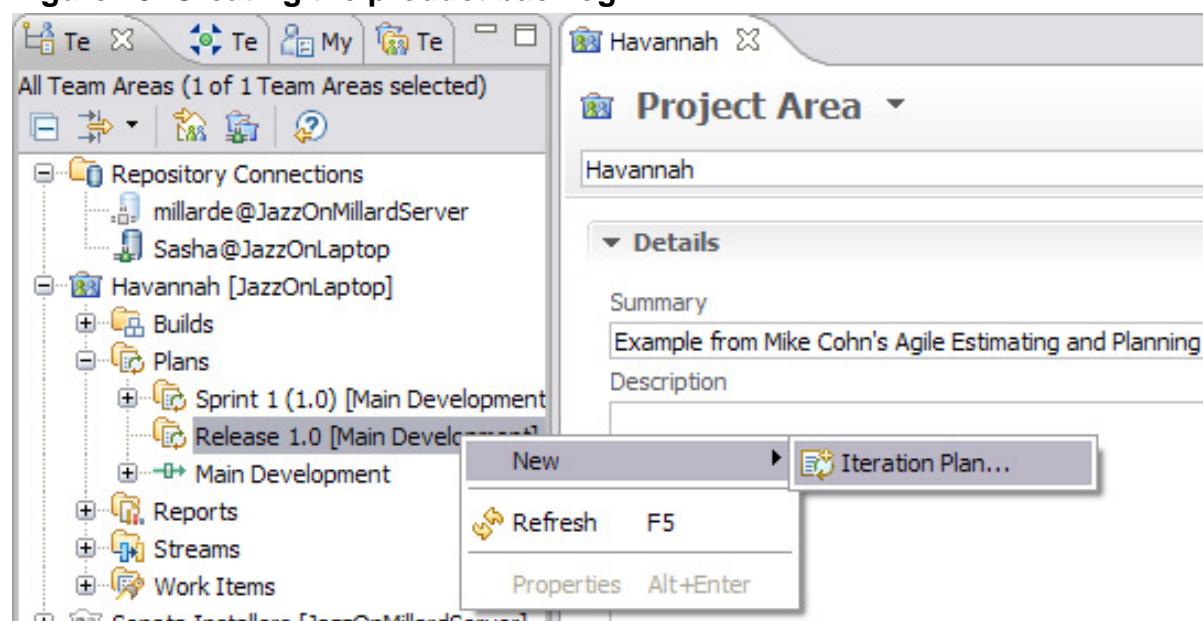
One of the most important artifacts in the Scrum method is the Product Backlog.

- To build it for this example, go to the **Team Artifacts** view (Figure 19) and add an iteration plan to the release by selecting **Release 1.0 > New > Iteration Plan**.

Tip:

You may have to open one or more tree nodes to see the Release 1.0 iteration. Additional release and sprint plans (if available) are under the development line (in this case, Main Development), but the current release and sprint are visible at the Plans level for easier access.

Figure 19. Creating the product backlog



- In the New Iteration Plan wizard that opens, enter **Product Backlog** for the name and click **Finish**.

A multi-page editor will open for the Product Backlog iteration plan (Figure 20).

- On the Overview page (see the tab at the bottom), you can enter information about your product by using wiki-style formatting. This will display as part of your project Web page in the Web UI.
- The Attachments area at the bottom of the editor view is a great place to add any requirements or other general documentation pertinent to this release so that they are readily available to all team members.

Figure 20. Editing the iteration plan

The screenshot shows the 'Product Backlog' view for the 'Havannah' team area in 'Release 1.0'. The toolbar includes standard icons for RSS, sharing, and saving. A message bar at the top indicates the team area and iteration, and provides instructions for using wiki-style syntax. On the left, a sidebar lists various heading styles from 'Bold (***)' to 'Heading 6 (=====)'. The main content area displays a large 'Heading 1' in bold black text on a yellow background. A status bar at the bottom of the editor window says 'Press 'Ctrl+Space' to show Attachment Proposals'.

Stories and Story Points

According to Mike Cohn, *stories*, also called *User Stories*, "describe functionality that will be valuable to either a user or purchaser of a system or software. *Story Points* indicate the size and complexity of a given User Story relative to other stories that are part of the project "

- Every Story should include a description that provides additional details about what is expected (a description can also be added when you add the Story to the plan). In this example, it references (imaginary) documentation that is attached to the Project Area.
- Each Story should be assigned a size in Story Points. See Mike Cohn's books, *User Stories Applied* or *Agile Estimating and Planning* (both cited in [Resources](#)), because a discussion of estimating and Story Points is

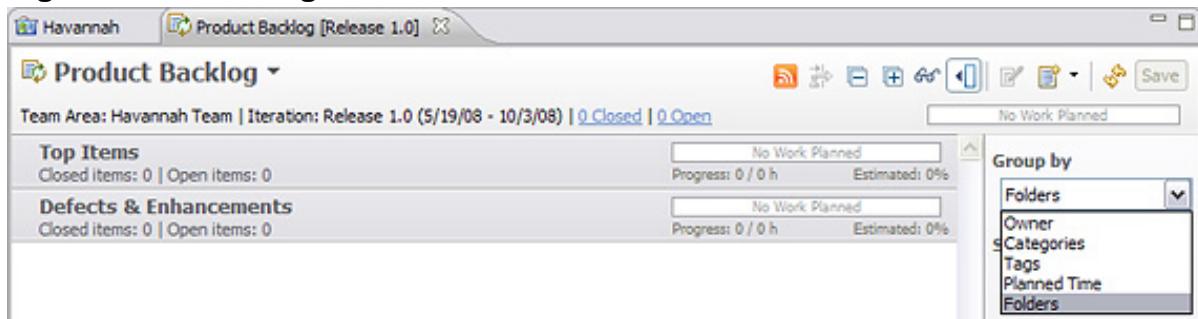
beyond the scope of this article.

- Every Story should also have acceptance tests identified by the team and Product Owner as proof that the Story was successfully completed and meets stakeholder expectations.

Each Story should eventually have all of these details completed. High-priority stories that are likely to be worked on soon should have these details completed as soon as possible and well before they reach the Sprint Planning meeting stage.

3. Click the **Planned Items** tab to begin adding backlog items.
4. Use the **Group by** drop-down menu on the right-side panel (Figure 21) to switch to the **Folders** view (if it is not the view already selected). This is the easiest view to work with when adding many new stories.

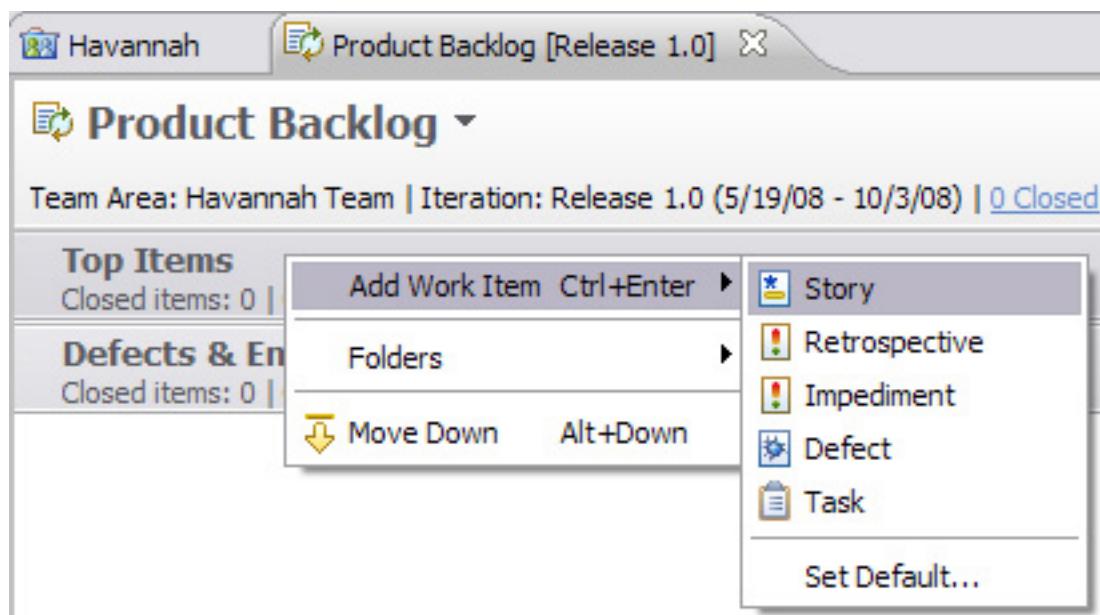
Figure 21. Switching to the Folder view



You can add folders (and rename the default ones) if you like. This is a simple mechanism for organizing the work items. For this example, you will add all of the Product Backlog items as stories to the **Top Items** folder.

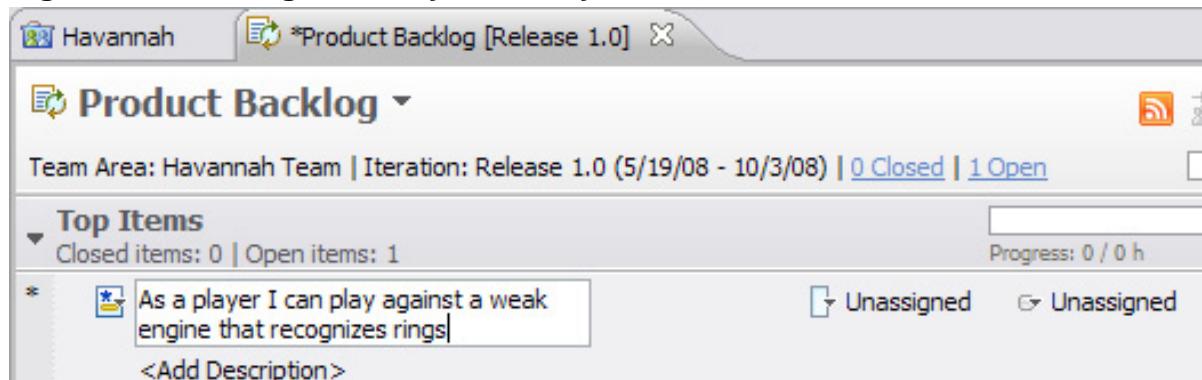
5. Right-click the **Top Items** folder, choose **Add Work Item** (Figure 22) and then **Story** to add the first item.

Figure 22. Adding product items as Stories



A new item is added to the folder and an in-place text area opens to enter the Story Summary (Figure 23).

Figure 23. Entering the Story summary



This is an easy way to enter new stories.

6. Use the drop-down menu to add additional stories (or you can use Ctrl+Enter as a keyboard shortcut to bring up the Work Item context and then a down arrow to move to the first item, Story, and select Enter).
7. Click **Save** on the upper-right of the Iteration Plan tab at any time to save your work.

After the stories are added, the next step for the Product Owner is to set priorities for them. Currently, only a 1-10 priority scale is available. The system remembers user ordering done with drag-and-drop (or use of the Alt+Cursor Up and Down shortcut). Therefore, if you arrange them within Priority that way, whenever you select the

User defined order from the **Group by** drop-down menu, you will see them the way that you arranged them. You can assign priorities easily by using the drop-down menu embedded in the list items, as Figure 24 shows.

Figure 24. Prioritizing Stories

The screenshot shows a product backlog in IBM Rational Team Concert. A context menu is open over the second item in the list, displaying a prioritization scale from 1 High to 10 Low. The item being prioritized is currently set to '6 Low'. The menu also includes options for 'Unassigned' and 'Medium' priority levels.

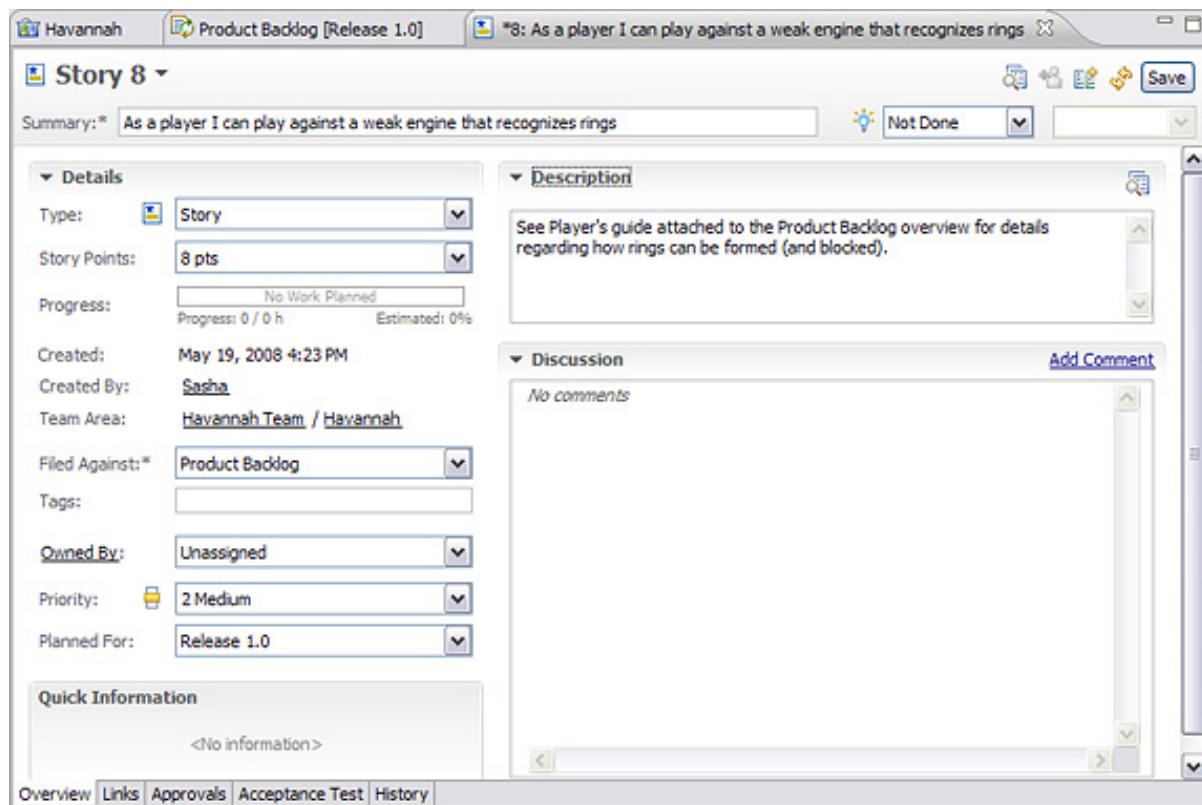
Priority Level	Description
1 High	Top priority
2 Medium	Medium priority
3 Medium	Medium priority
4 Medium	Medium priority
5 Medium	Medium priority
6 Low	Current selection
7 Low	Low priority
8 Low	Low priority
9 Low	Low priority
10 Low	Lowest priority
Unassigned	No priority assigned

Below the prioritization menu, the backlog header shows 'Defects & Enhancements' and 'Closed items: 0 | Open items: 0'. At the bottom, there is a navigation bar with tabs for 'Overview', '*Planned Items', and 'Charts'.

Of course, it wouldn't be much of a Product Backlog if all we had were Story Summaries.

8. Double-click a Story to open its editor (in this example Story that says "As a player, I can play against a weak engine that recognizes rings" shown in Figure 25).

Figure 25. Editing a Story

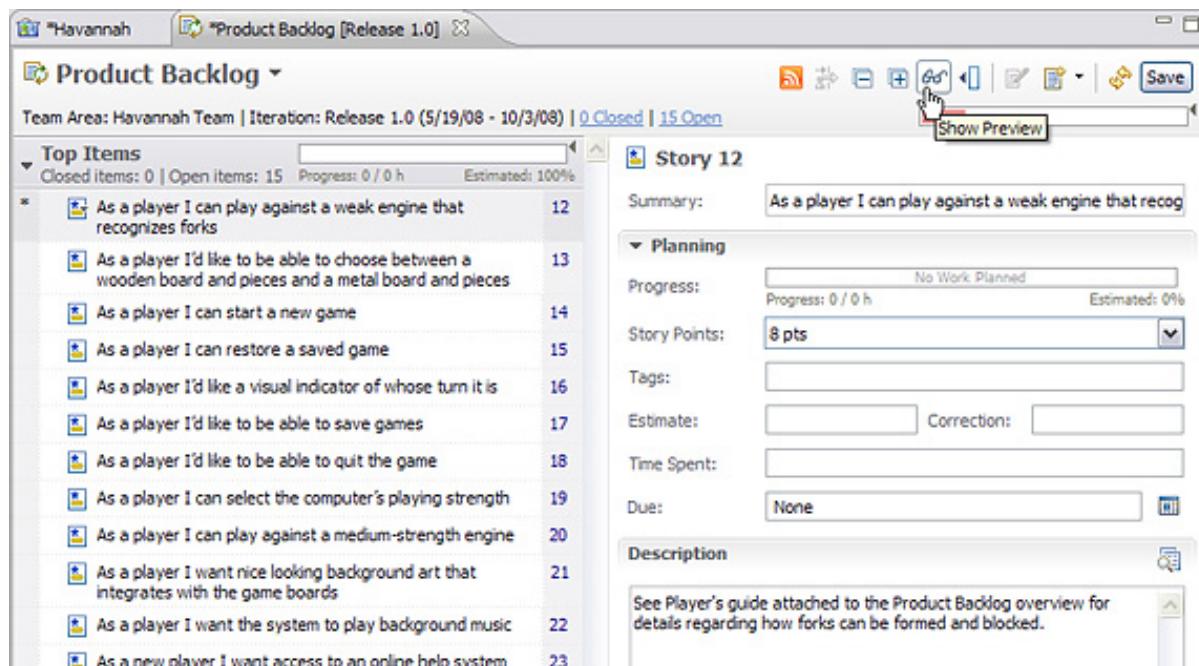


9. Click on the **Acceptance Test** tab at the bottom and add the stories there.

Tip:

Another good way to deal with adding details while also building the list is to turn on the Preview mode in the iteration planning editor. Click the eye glasses icon in the toolbar (Figure 26), and the editor will split to show the Iteration plan and the currently selected item side-by-side.

Figure 26. Previewing Stories



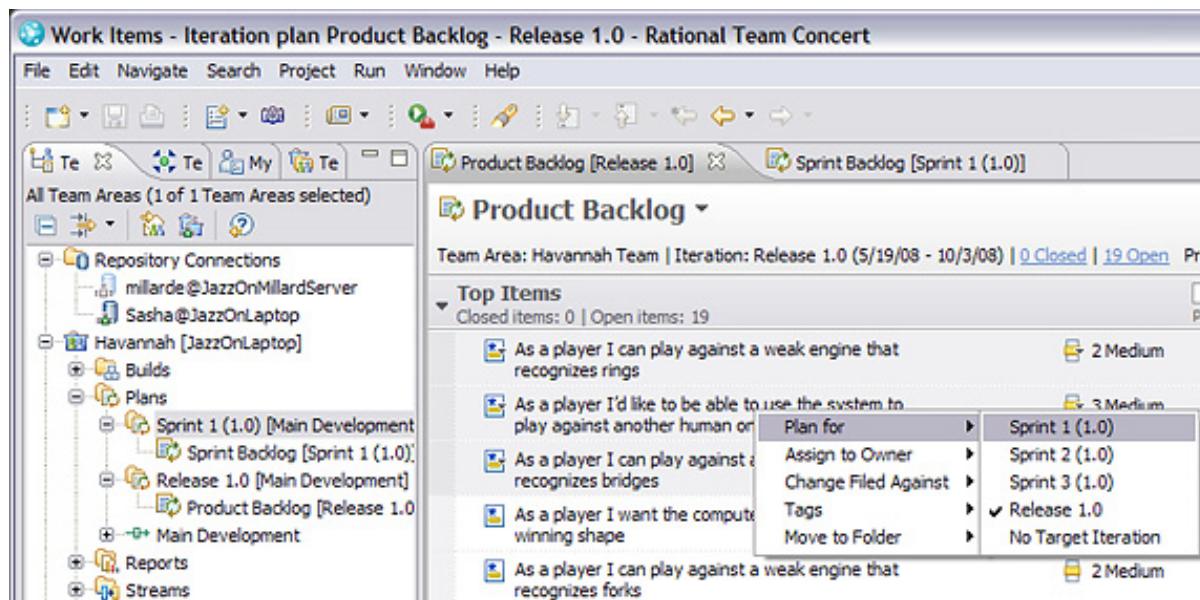
Plan the iteration

As discussed in the [Overview of the Scrum process](#) section previously, each iteration, or *sprint*, starts by pulling items from the Product Backlog into the iteration and developing a Sprint Backlog of tasks that the team needs to complete.

Create a Sprint Backlog iteration plan

1. First, create a Sprint Backlog iteration plan.
 - A. Similar to the previous instructions for creating the Product Backlog iteration plan, right-click on **Sprint 1 (1.0)** in the Plans node of the Team Artifacts view, select **New** and then **Iteration Plan**.
 - B. In the wizard, enter **Sprint Backlog** for the name.
 - C. With the Product Backlog window open, select the desired stories, right-click, and select **Plan For** and then **Sprint 1.0**.

Figure 27. Creating an iteration plan



This will move the stories to the Sprint Backlog tab. Initially, they will show small arrows in the left margin, indicating that they are awaiting a move.

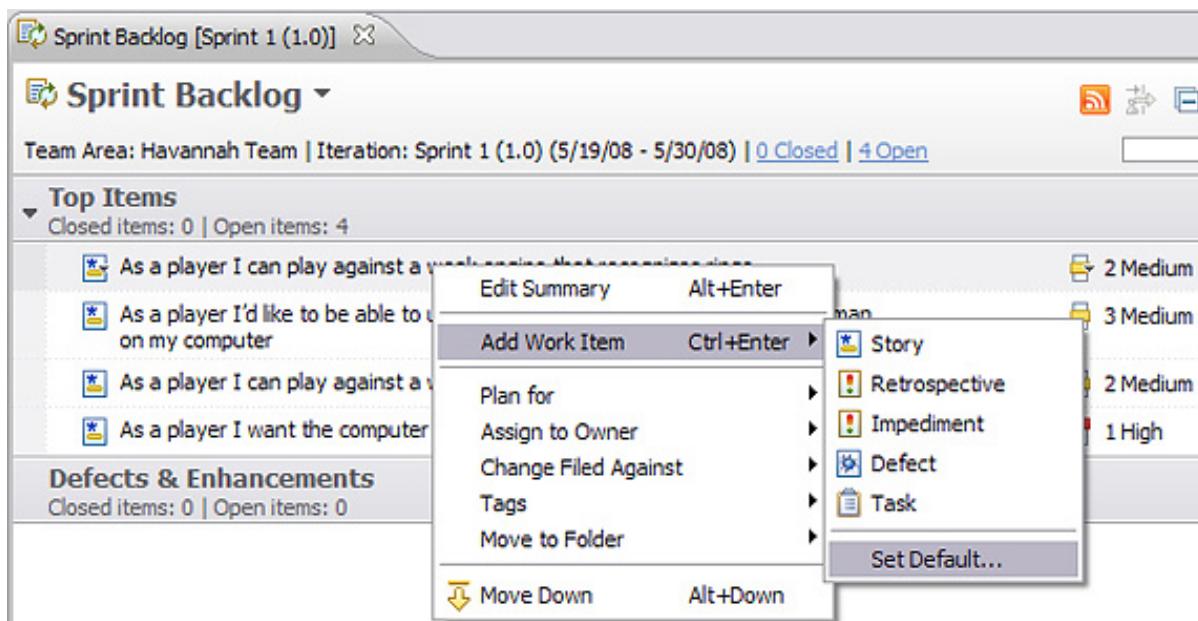
2. Click **Save** to complete the move.
3. Click the **Sprint Backlog** tab to bring it forward.

Add tasks

Now add tasks to the stories.

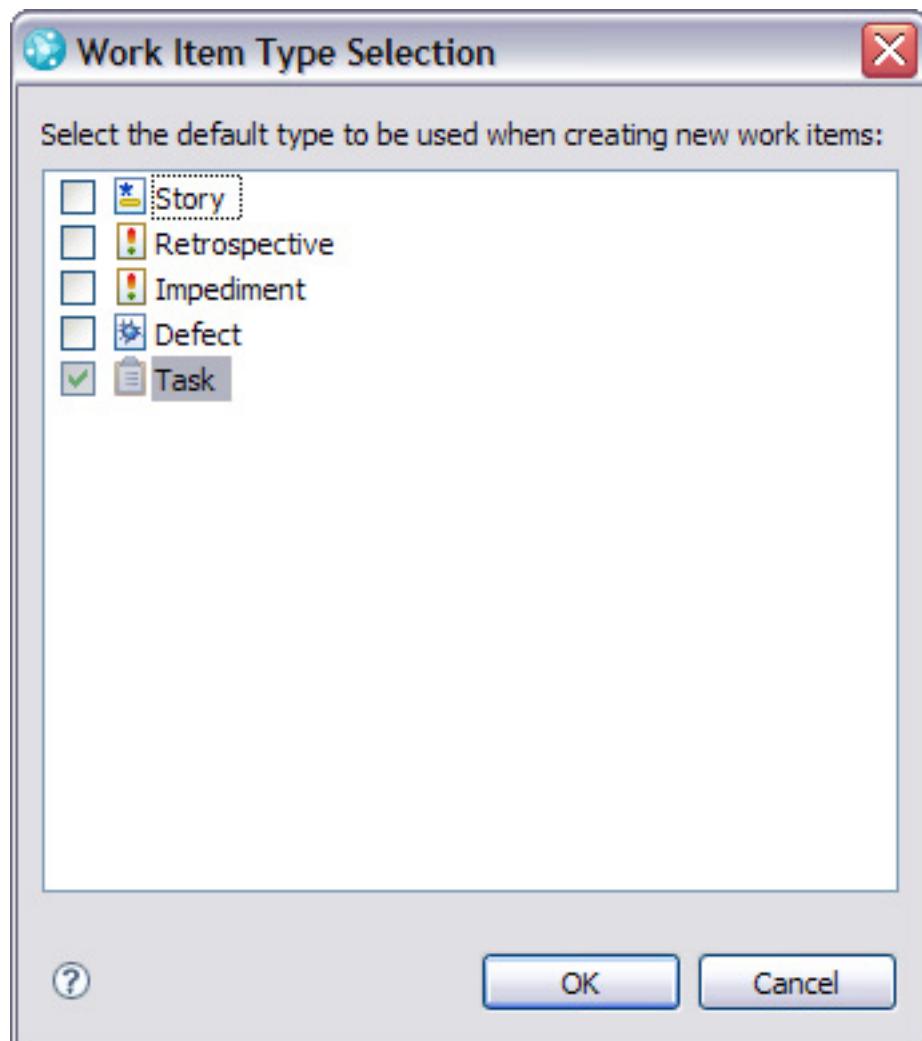
1. First, set up a shortcut to make this easier: Right-click on the first Story to add a Work Item (much in the way that you added stories), but select the **Set Default** option (Figure 28) from the drop-down menu, instead.

Figure 28. Adding tasks



You'll get a dialog like the one shown in Figure 29 so that whenever you press Ctrl+Enter, it will automatically add an item of the default type. You can use this dialog later to remove or reassign the default Work Item type.

Figure 29. Setting the default Work Item type



Now click **OK**.

2. To start entering tasks, select the first Story and press **Ctrl+Enter**.

A task will be created beneath the selected Story. However, notice that it is at the same indentation level as the Story (Figure 30).

3. Because this task "belongs" to the Story, press **Tab** once to move it *under* the Story.

Note:

Rational Team Concert will flag the task as not having the required Summary field entered yet, but the warning will go away after the summary is added. You can enter the Summary first to avoid the warning if you prefer.

Figure 30. Indenting a task

The screenshot shows the 'Sprint Backlog' view for 'Sprint 1 (1.0)'. The 'Top Items' section displays two tasks: 'As a player I can play against a weak engine that recognizes rings' and 'Code state management classes'. The second task is indented under the first, indicating it is a sub-story. A tooltip 'Tasks added to this Story will automatically be considered children, or sub-stories.' is visible below the second task.

Tasks added to this Story will automatically be considered children, or sub-stories.

In a typical Sprint Planning meeting, team members will call out tasks that need to be done. Just getting them all recorded will be sufficient for now. Time estimates and assignments can come later in the meeting.

4. Continue adding tasks for this and the other stories.

Tip:

Do this one Story at a time, because, as estimates are created, it will be necessary to track the workload on the team members. There is no point in planning more work than there is time to do.

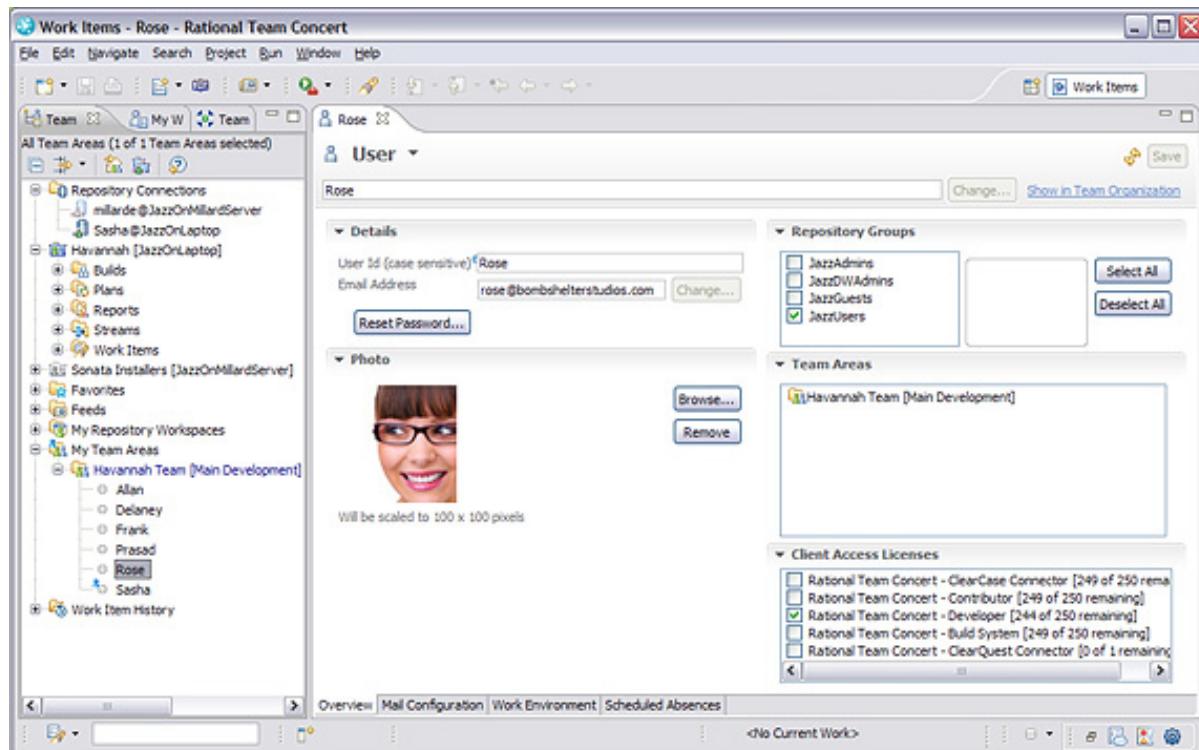
Add time estimates

After the tasks for the first Story are determined, it is time for the team to add estimates.

For team workload calculations to be accurate, adjust team members' availability. Perhaps a team member has responsibilities outside of the team that limit participation or has a vacation planned. These are adjusted on the user's page.

1. In the Team Artifacts view, open the **My Team Areas** node and then the **Havannah Team** folder.
2. Double-click on the name **Rose** to open the details for her (Figure 31).

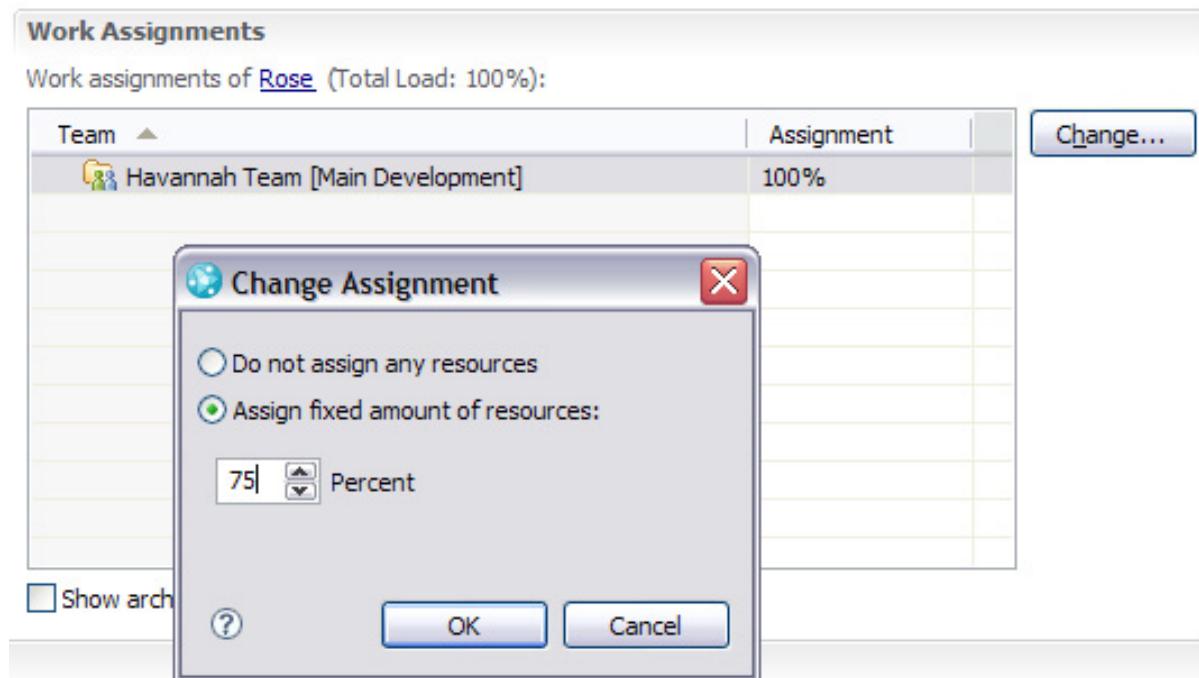
Figure 31. Adjusting team load



Rose is available to the team only 75% of the time.

3. Click the **Work Environment** tab at the bottom and then, in the Work Assignment section, click the **Havannah Team** line and the **Change** button (Figure 32).

Figure 32. Changing assignment level

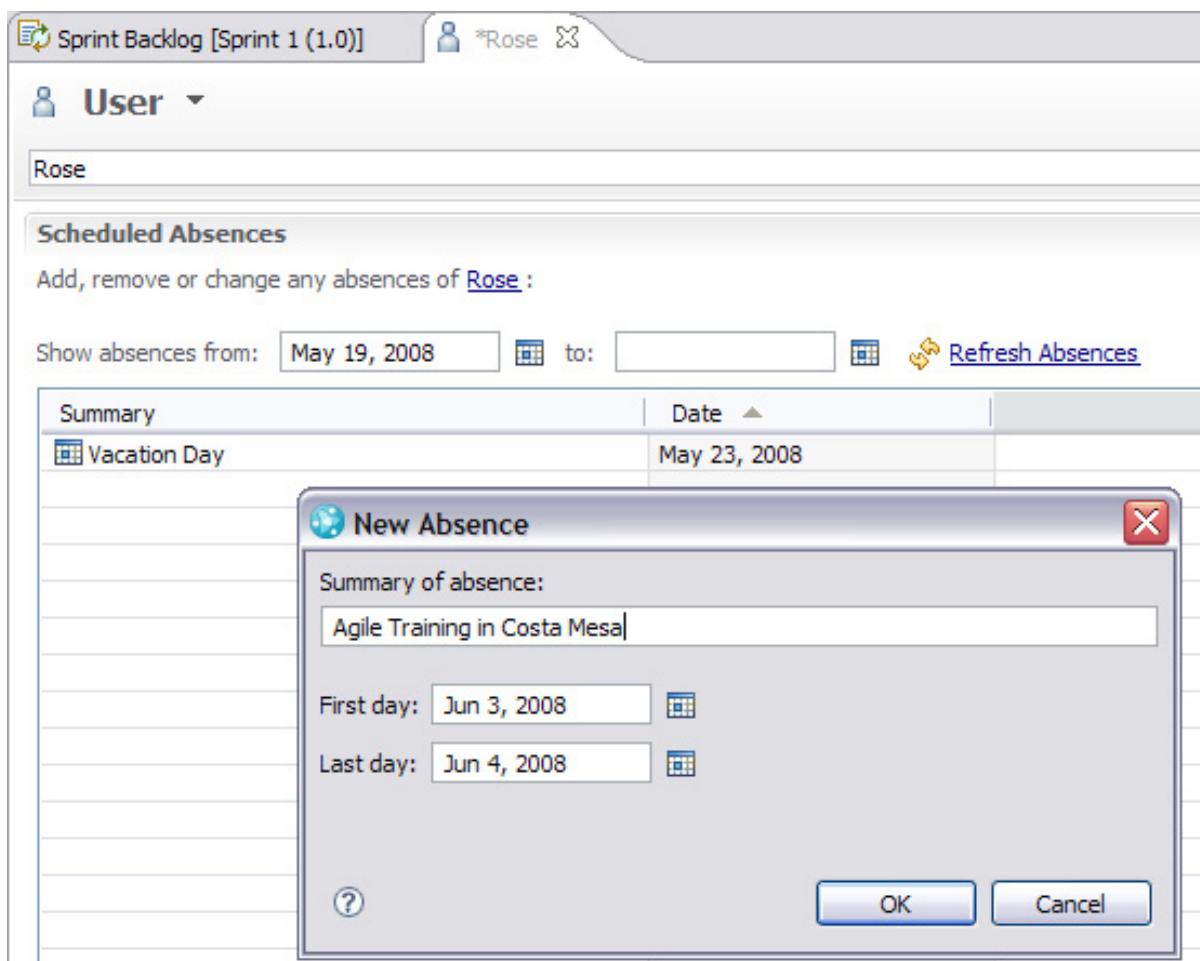


4. Lower her assignment to 75%.
5. Click **OK**.

This will adjust the number of hours that Rose is available for work assignments. Notice that she also has a vacation day coming up.

6. Switch to the **Scheduled Absences** tab, click **New** on the right side of the list, and enter the date for her vacation time (Figure 33).
7. Click **OK** and **Save** to update Rose's details.

Figure 33. Adding absences



Rose is the only team member who will be away during this iteration, so you can start estimating tasks now.

8. Return to the **Sprint Backlog** tab (open the Plans node and double-click on the Sprint Backlog if necessary).
9. As the team decides what the estimates should be, they can easily be specified by using the drop-down menu triggered by clicking the small **clock icon** shown in Figure 34. (You could double-click on each task and open it, instead, but this is easier). If your time estimate is not in the drop-down menu, choose **More**, and a small input dialog will pop up where you can enter your estimate. Enter the estimates in terms of days and hours, for example: 10 h or 2 d.

Figure 34. Entering time estimates

The screenshot shows the 'Sprint Backlog [Sprint 1 (1.0)]' window. In the center, there's a list of tasks under the heading 'Top Items'. To the right of each task are columns for 'Closed' (0% Closed), 'Priority' (2 Medium), and 'Estimate' (Unassigned). A context menu is open over the last task in the list, showing a dropdown menu for 'Estimate' with options like '30 minutes', '1 hour', '2 hours', etc., down to 'No Estimate' and 'More...'. The 'More...' option is currently selected.

Task Description	Closed	Priority	Estimate
As a player I can play against a weak engine that recognizes rings	0% Closed	2 Medium	Unassigned
Code state management classes	2 days	Unassigned	Unassigned
Write automated tests for state management classes	Unassigned	Unassigned	Unassigned
Have move engine pursue an unblocked ring	Unassigned	Unassigned	Unassigned
Write automated tests for unblocked rings	Unassigned	Unassigned	Unassigned
Have move engine pursue a ring even if the human player tries to block it	Unassigned	Unassigned	Unassigned
Identify test cases for trying to make a blocked ring	Unassigned	Unassigned	Unassigned
Automate test cases for trying to make a blocked ring	Unassigned	Unassigned	Unassigned
Have the move engine try to block a human player who is trying to make a ring	Unassigned	Unassigned	Unassigned
Identify test cases for blocking a human player making a ring	Unassigned	Unassigned	Unassigned
Automate test cases for blocking a human player making a ring	Unassigned	Unassigned	Unassigned

- For the team workload to make sense, you need to assign the tasks, as well.

Tip:

Typically, Scrum team members sign up for tasks, and the assignments can be made at the same time. Because different team members may take different amounts of time to complete a task, it hardly makes sense to finalize a time estimate without an assignment. Thus, a similar drop-down menu is available that lists the members assigned to the Havannah team (Figure 35).

Figure 35. Assigning users to tasks

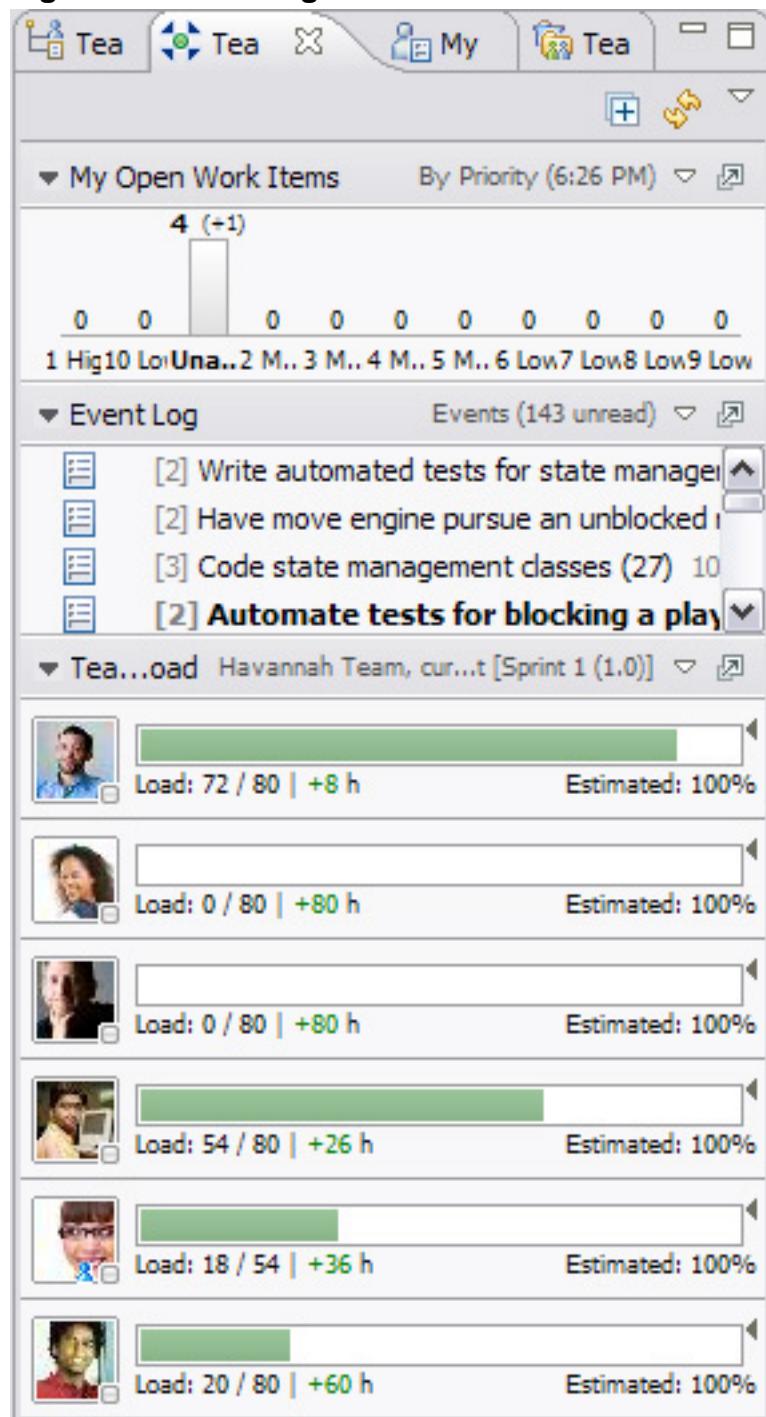
This screenshot is similar to Figure 35 but focuses on the 'Assignee' column. A context menu is open over the last task, showing a dropdown menu for 'Assignee' with names like Allan, Delaney, Frank, Prasad, Rose, and Sasha. The name 'Sasha' is currently selected.

Task Description	Closed	Priority	Assignee
As a player I can play against a weak engine that recognizes rings	0% Closed	2 Medium	Unassigned
Code state management classes	2 days	Unassigned	Unassigned
Write automated tests for state management classes	1 day 2 hours	Unassigned	Unassigned
Have move engine pursue an unblocked ring	1 day 4 hours	Unassigned	Unassigned
Write automated tests for unblocked rings	1 day 4 hours	Unassigned	Unassigned
Have move engine pursue a ring even if the human player tries to block it	1 day	Unassigned	Unassigned
Identify test cases for trying to make a blocked ring	4 hours	Unassigned	Unassigned
Automate test cases for trying to make a blocked ring	4 hours	Unassigned	Unassigned

Team workload can be monitored as tasks are estimated. The easiest way is to make the Team Organization view visible (Figure 36), and it will update as estimates

are made. This view can be kept open along with the Sprint Backlog view.

Figure 36. Team Organization view



Notice that Rose has only 54 available work hours because of her limited availability and vacation day. You would continue estimating tasks until team members no longer have time available. For Scrum teams to be successful, remember to leave

some slack in everyone's workload to adjust for estimates that are inadequate or for interruptions.

Note:

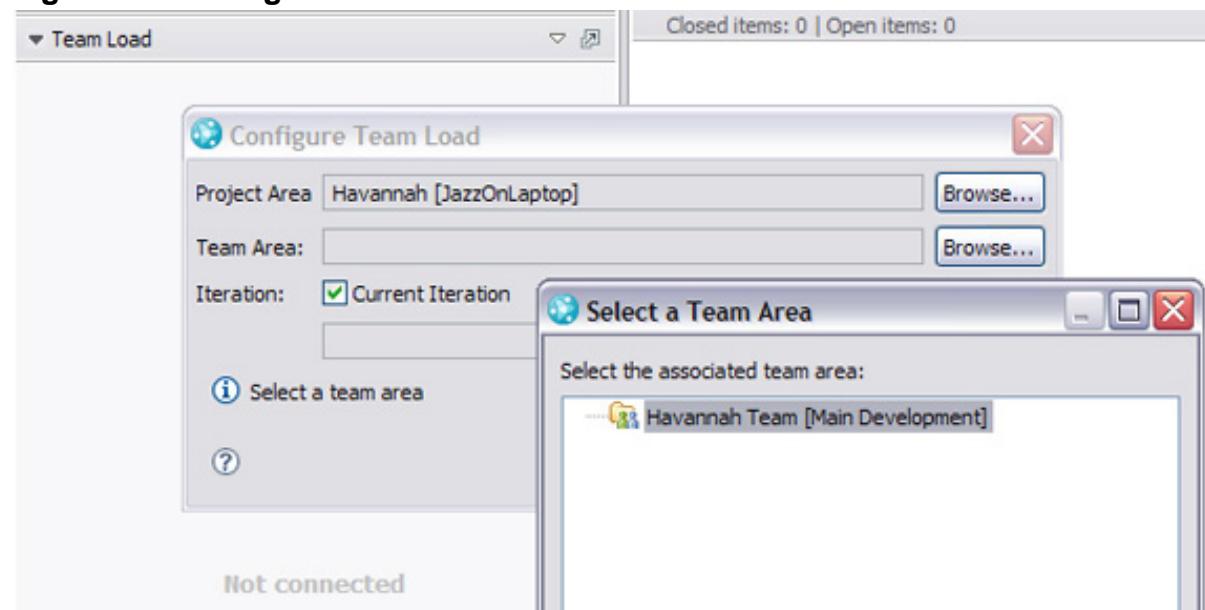
If the Team Load view is empty or indicates "Not Connected, you need to configure it: Click the downward white triangle to the right of the Team Load view to use its drop-down menu (Figure 37), and select **Configure**.

Figure 37. Configuring the Team Load view



11. In the subsequent dialog, select the **Project Area** (Havannah should already be shown) and select the **Team Area (Havannah Team)**.
12. Click **OK** twice and the workload for team members should be visible as in Figure 38.

Figure 38. Adding team members into the Load view



Do the work

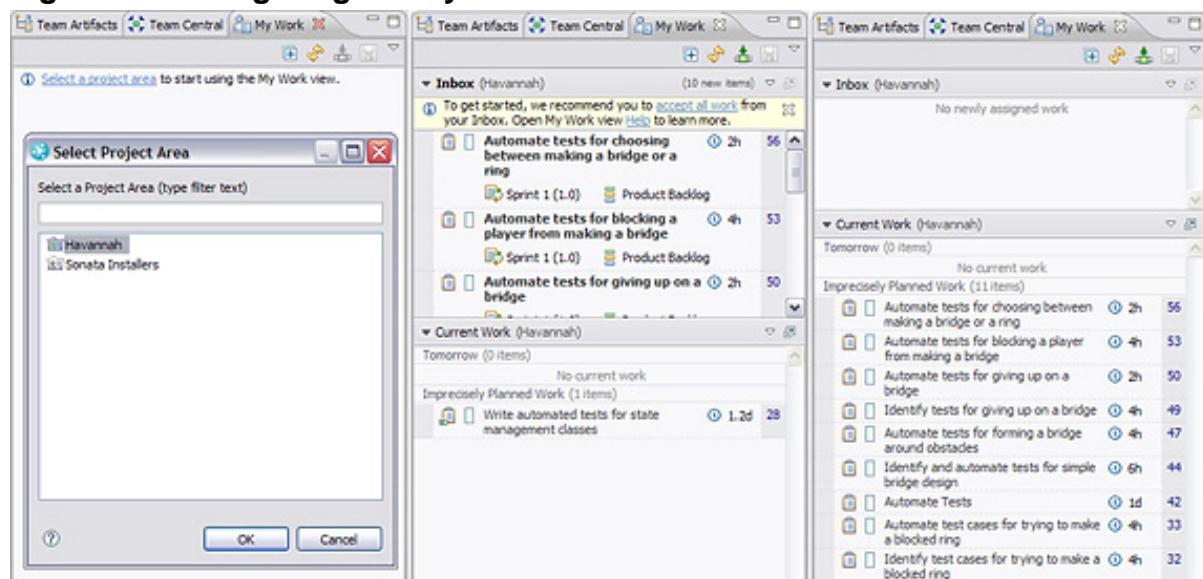
The easiest way for team members to track their work is with the **My Work** view, which opens in the left pane by default. The first time that you use it, it may need to

be configured. If so, assuming that you logged in as Prasad, the three screen captures that follow (Figures 39, 40, and 41) illustrate the steps required to populate the view.

1. In the first area, shown in Figure 39, click the link to select a **Project Area** to configure the view. The dialog shows all available Project Areas.
2. Select **Havannah** and click **OK**.
3. Because this is the first time that Prasad has used this view, all of the work is in his Inbox. Follow the instructions and click the link to accept the work.

The work is then moved into the Current Work panel.

Figure 39. Configuring the MyWork view



Notice that the current work panel (Figure 40) indicates that the work is "Imprecisely Planned," yet it is estimated in hours and assigned to someone.

Figure 40. Accepting assigned work

Current Work (Havannah)			
Tomorrow (0 items)			
Imprecisely Planned Work (11 items)			
<input type="checkbox"/>	Automate tests for choosing between making a bridge or a ring	(1) 2h	56
<input type="checkbox"/>	Automate tests for blocking a player from making a bridge	(1) 4h	53
<input type="checkbox"/>	Automate tests for giving up on a bridge	(1) 2h	50
<input type="checkbox"/>	Identify tests for giving up on a bridge	(1) 4h	49
<input type="checkbox"/>	Automate tests for forming a bridge around obstacles	(1) 4h	47
<input type="checkbox"/>	Identify and automate tests for simple bridge design	(1) 6h	44
<input type="checkbox"/>	Automate Tests	(1) 1d	42
<input type="checkbox"/>	Automate test cases for trying to make a blocked ring	(1) 4h	33
<input type="checkbox"/>	Identify test cases for trying to make a blocked ring	(1) 4h	32
<input type="checkbox"/>	Write automated tests for unblocked rings	(1) 1.5d	30
<input checked="" type="checkbox"/>	Write automated tests for state management classes	(1) 1.2d	28

4. Within this view, you can use your mouse cursor to drag items (or use Alt + cursor up or down) to set an order for the work items (which Jazz will remember).

That way, rather than a random list of tasks, the My Work view will show the items in the order that the team member plans to complete them. Also, when the work is viewed in the Iteration Plan's Group by Planned Time view, this timeline for the pending work will be visible to all users.

Figure 41 shows two other ways to see the work that is assigned:

- Under the Work Items folder, open the **Shared Queries** and **Predefined** folders and double-click the **Open assigned to me** query. This will open the query view shown in Figure 41.
- Also, in the Sprint Backlog view, you can change the Group By drop-down

menu in the right panel to **Group by Owner**. This will show the stories that each user is associated with and, when a Story is expanded, it will show that user's tasks that are related to that Story.

Figure 41. Two views of assignments assigned to me

I	Status	P	S	Summary	Owned By
56	New			Automate tests for choosing between making a bridge or a ring	Prasad
53	New			Automate tests for blocking a player from making a bridge	Prasad
50	New			Automate tests for giving up on a bridge	Prasad
49	New			Identify tests for giving up on a bridge	Prasad
47	New			Automate tests for forming a bridge around obstacles	Prasad
44	New			Identify and automate tests for simple bridge design	Prasad
42	New			Automate Tests	Prasad
33	New			Automate test cases for trying to make a blocked ring	Prasad
32	New			Identify test cases for trying to make a blocked ring	Prasad
30	New			Write automated tests for unblocked rings	Prasad
28	In Progr...			Write automated tests for state management classes	Prasad

From either the Query view or the My Work view, the drop-down menu enables you to start working on an item. You can also open a task to change its state (see Figure 42).

Figure 42. Working on a Task

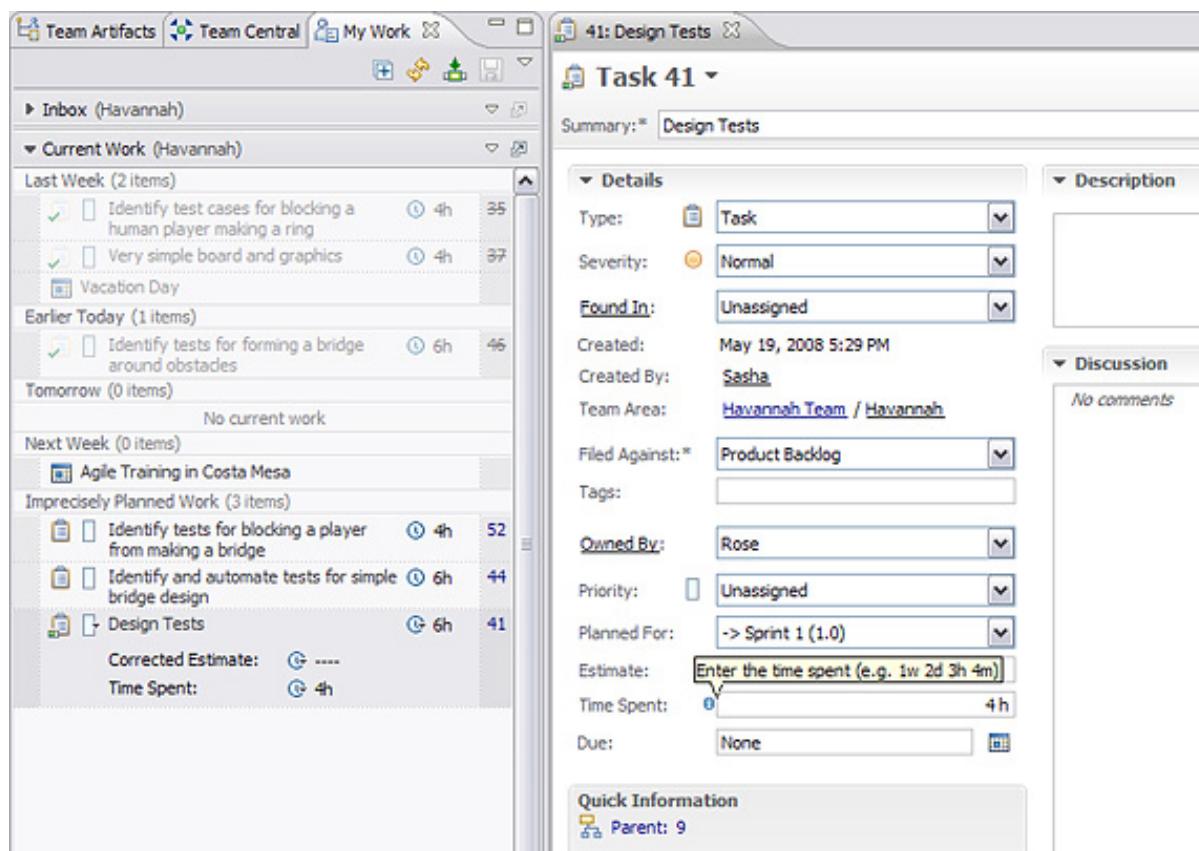
The screenshot shows a list of work items in a grid format. The columns are labeled I, Status, P, S, Summary, and Owned By. The 'Status' column contains entries like 'New', 'In Prog...', and 'Open'. The 'Summary' column contains detailed descriptions of tasks. A context menu is open over the last row, listing options such as 'Copy Web URL', 'Copy ID and Summary', 'Link to Work Item...', 'Move/Copy to Project Area...', 'Mark Read', 'Find Potential Duplicates', 'Duplicate', 'Start Working' (which is highlighted in grey), and 'Stop Working'.

I	Status	P	S	Summary	Owned By
56	New			Automate tests for choosing between making a bridge or a ring for blocking a player from making a bridge	Prasad
53	New			Automate tests	Prasad
50	New			Automate test	Prasad
49	New			Identify tests f	Prasad
47	New			Automate test	Prasad
44	New			Identify and au	Prasad
42	New			Automate Test	Prasad
33	New			Automate test	Prasad
32	New			Identify test ca	Prasad
30	New			Write automati	Prasad
28	In Prog...			Write automati	Prasad

Track and report progress

Team members should start working on tasks, report time spent on them, and resolve them. Given that the Scrum method values work completed, not work started, it is preferable to start and complete a work item before moving on to the next item. This enables the team to complete stories more quickly and move them into the "Ready for Sprint Review" state sooner. To help team members track their history and success in estimating tasks, Time Spent should be recorded against the tasks that people worked on each day (see Figure 43). Time Spent is a simple text field; therefore, if a task is worked on for multiple days, each team member needs to update the Time Spent field each day to include the previous value plus any additional time.

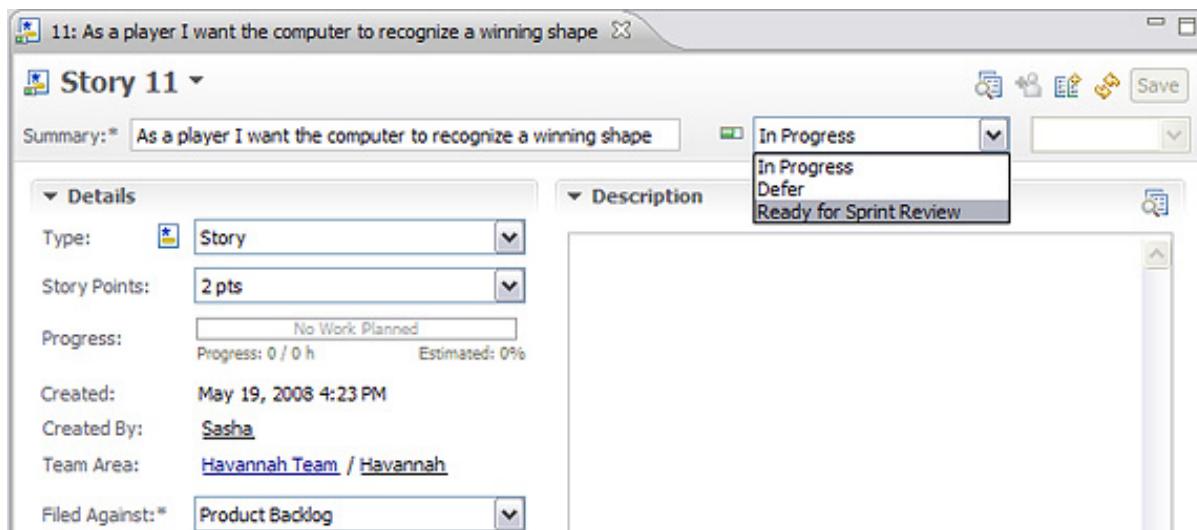
Figure 43. Recording time spent



Use the Discussion feature to record progress or to capture information learned about the task. Any team member can update information in this field, and it is an excellent way to capture the history of the work item.

Story progress should be tracked by starting work on it, as well. Often, no particular team member is made responsible for the Story as a whole, thus, as the tasks under the Story are completed, the ScrumMaster can set the Story as "Ready for Sprint Review" during the Daily Scrum meeting (see Figure 44). This Story status is reported on the [Project Area Dashboard](#).

Figure 44. Readying Story for Sprint Review



One way to easily prepare for the Daily Scrum meeting is to use the Recently Modified query to identify tasks that have been updated in the last day or so (Figure 45). The number of hours that qualify as "recently" is configurable as a query argument, with the default being 12 hours.

Figure 45. Running the Recently Modified query

Work Items						Tag Cloud	Problems	Team Advisor				
Found 8 work items - Recently modified												
	ID	Status	P	S	Summary			Owned By				
								Allan	Allan	Allan		
	43	In Prog...			Engine can find path from one corner to another (that is, form a bridge)							
	34	In Prog...			Have the move engine try to block a human player who is trying to make a ring							
	31	Resolved			Have move engine pursue a ring even if the human player tries to block it							
	33	In Prog...			Automate test cases for trying to make a blocked ring							
	32	Resolved			Identify test cases for trying to make a blocked ring							
	30	Resolved			Write automated tests for unblocked rings							
	38	In Prog...			Draw empty board							
	27	In Prog...			Code state management classes							

This list will quickly show who has been doing what. It helps determine who has not reported any progress lately (notice that Rose's name does not appear in the list).

Continually update the Sprint Burndown Report

How the burndown rate is calculated

Many of Jazz's historical reports, such as the Burndown Report, gather data daily through a data warehousing process that runs nightly. Therefore, if team members do not update their data by the end of each day, their work item status will not be properly reflected in the report. It is possible to force collection of the work item data for burndown reporting, but this can be a time-consuming process and can affect server performance if it is done during the work day.

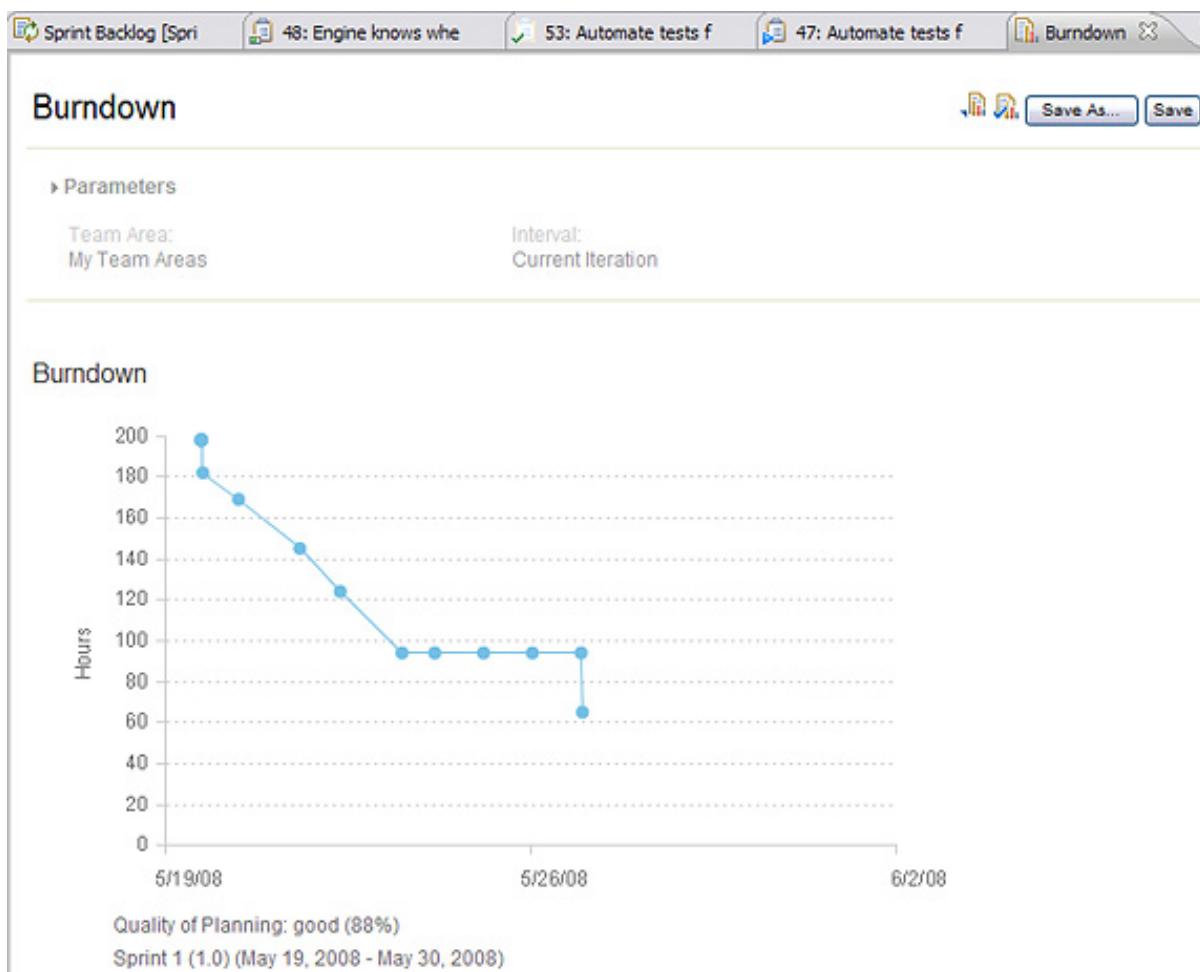
In the Burndown Report sample shown here (Figure 46), the data warehousing process was run twice on the first day and twice on the last day shown. This resulted in a vertical drop between the two points plotted for the same date. Jazz doesn't skip non-work days in

its plotting, thus the flat line partly represents the weekend time. This article was researched and written using Beta 3 of Jazz and Rational Team Concert and an early access version of the Scrum template. You can anticipate that these aspects will be improved as the product matures.

The team uses the Sprint Burndown Report to see how work is progressing. They use it every day, because it provides an instant answer to the question "Are we on track to finish all the work that we committed to do?" As work is completed, the line trends closer to zero (work remaining). The burndown rate should be easily visible to all of the team all of the time.

1. Access the Sprint Burndown Report by opening the **Reports** node in the Team Artifacts view and then opening the **Shared Reports and Work Items** folders.
2. Double-click the Burndown Report. (For the Burndown Report to look similar to the one in Figure 46, you must update and log work completed by various team members for several days.)

Figure 46. Viewing the Burndown report

**Tip:**

You can add reports, such as the Burndown Report, and queries, such as "Open Assigned to Me" and "Recently Modified," to your Favorites folder by right-clicking on them and selecting the **Add to Favorites** drop-down menu item. This will make them much easier to access.

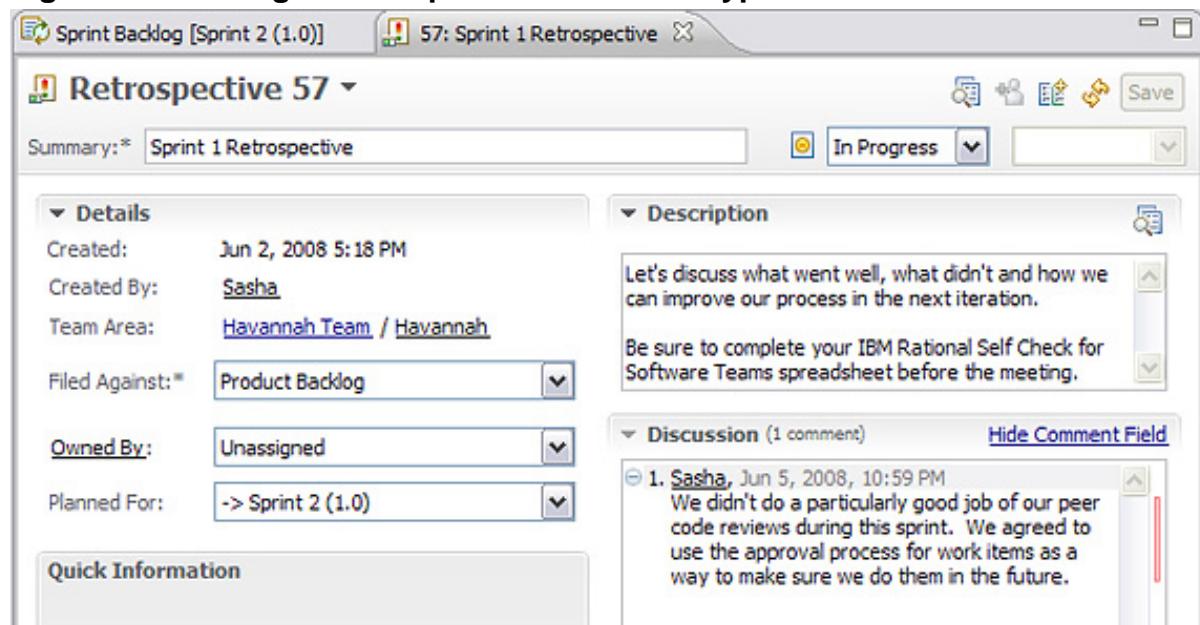
Schedule the Iteration Retrospective meeting

An important part of the Scrum process is the Sprint Review meeting. The first part of this is the demo to the stakeholders. Rational Team Concert may not be part of this, because the point is to show off working software, not the list of tasks. However, feedback and comments from the review meeting should be captured, either in the Iteration's Overview page or as an attachment to that page.

The next part of the Sprint Review meeting is the Iteration Retrospective (sometimes called Reflection). This is a chance for the team to discuss what went well, what didn't, and what they plan to do about it. The Scrum Process template defines a Retrospective work item type (Figure 47) that can be used to make sure that the

reflection is performed and to track the team's comments and plans.

Figure 47. Viewing a Retrospective Work Item type



Do it all over again

The life of a healthy Scrum team is one filled with a rhythm of success. Plan some, work some, deliver, and repeat. When it is time for the next sprint, use the same approach for creating an iteration plan for Sprint 1 to create one for Sprint 2, then start moving Product Backlog items into it for planning.

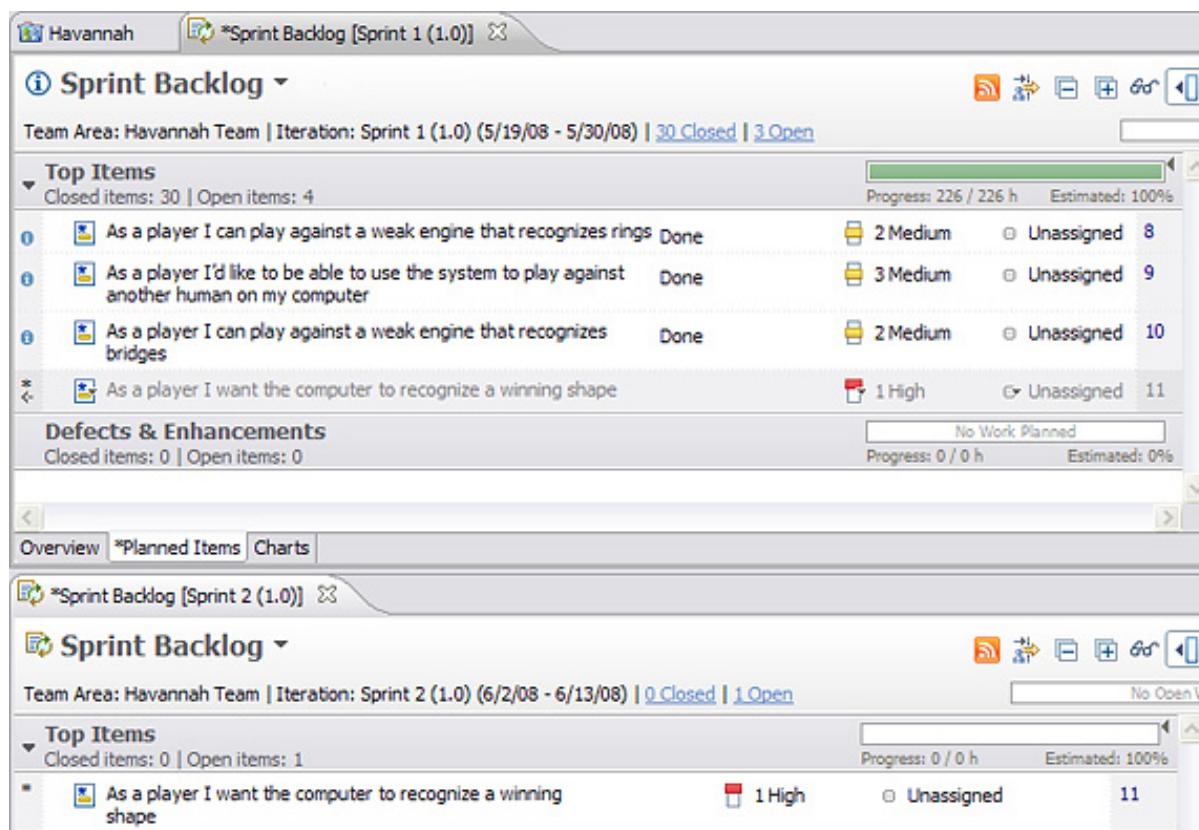
If a Story does not get completed in the current sprint, drag it to the new Sprint (see Note):

1. By clicking the tab for the **Sprint 2 Backlog** and dragging it toward the bottom of the screen, you can make both plans visible at once.
2. Then drag the Story from one to the other.

Note:

You must drag it to the new sprint for its child tasks to move to the new sprint with it, or you can select all of the items at once. Merely reassigning the Story will not move its children, or subtasks.

Figure 48. Planning the next iteration



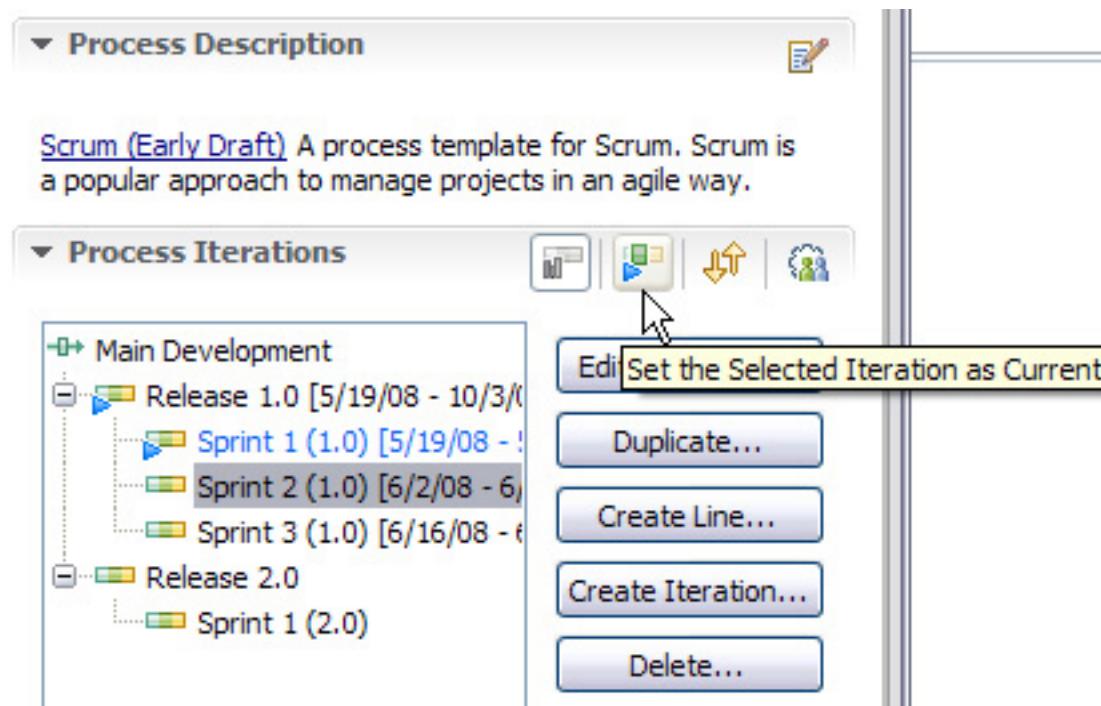
Important:

The asterisk at the left of the Story indicates that the changes have not been saved yet. The gray arrow pointing to the left of the grayed copy in the Sprint 1 window indicates that a move is pending for that item. When you click **Save**, the move will be completed.

Even though you created dates for the sprints, Rational Team Concert does not automatically shift to the next sprint from the current one just because time has marched forward. You must manually adjust which sprint is considered current:

3. Open the Havannah project.
4. In the Process Description section, click on **Sprint 2** (the current Sprint) and then the button with the little blue triangle to move the current designator to the new Sprint.

Figure 49. Setting the current iteration



The Rational Team Concert Web UI

Not every team member either has or wants to install an Eclipse-based client. Many of the Rational Team Concert features are available through the Web UI. There are even some features, such as the Project Area Dashboard, that are unique to the Web interface. Dashboards can be customized by the user, and they come in slightly different configurations, depending on the process template installed. Figure 50 shows the Scrum Dashboard for the Havannah project. (The dashboard for the released version may differ.)

Figure 50. Scrum Dashboard

The screenshot shows the Rational Team Concert interface with the 'Havannah' dashboard selected. The 'General' tab is active. A tooltip is displayed over the 'Progress Report' section, showing the following data:

- Work Hours Done: 154 of 206
- Expected Work Hours Done: 130 (Ahead by 24 hrs)
- Items Estimated: 100%
- Items Completed: 19 out of 34(56%)

The interface employs many Web 2.0-like behaviors that are similar to the progress report bubble that you get if you hover your cursor over the Sprint bar. A similar popup view shows at least partial query results if you hover your cursor over one of the queries at the bottom. Because this is the Scrum Dashboard, the list of open impediments or blocks is in the middle, at the top of the page.

What's missing? The Scrum burndown graph, of course. (As of RC5, there is a Trends tab with the burndown graph on it, along with a Story Points by Iteration report, but the example in Figure 51 also explains how to customize the dashboard.) This is a crucial artifact and an important indication of how things are progressing, and it is easy enough to add.

1. Click the **downward arrow** next to General on the tab at the top (Figure 51).
2. Select **Add Viewlet**. The Add Viewlet panel will open at the top of the page (Figure 51).

Figure 51. Adding a viewlet

The screenshot shows the Rational Team Concert interface. The top navigation bar includes 'Project Areas', 'Dashboards' (which is selected), 'Work Items', 'Iteration Plans', 'Source Control', and 'Reports'. On the left, there's a sidebar with 'Recently Viewed' (containing 'Havannah') and 'Shared Dashboards' (containing 'My Profile', 'Havannah' which is selected, 'Havannah Team', and 'All Dashboards'). The main area is titled 'Havannah' and shows a 'General' viewlet. A context menu is open over the 'Welcome' viewlet, with options like 'Add Viewlet', 'Rename', 'Refresh', and 'Layout'. Below the viewlets, there's a section for 'Havannah Description' with the text: 'Example from Mike Cohn's Agile Estimating and Planning'.

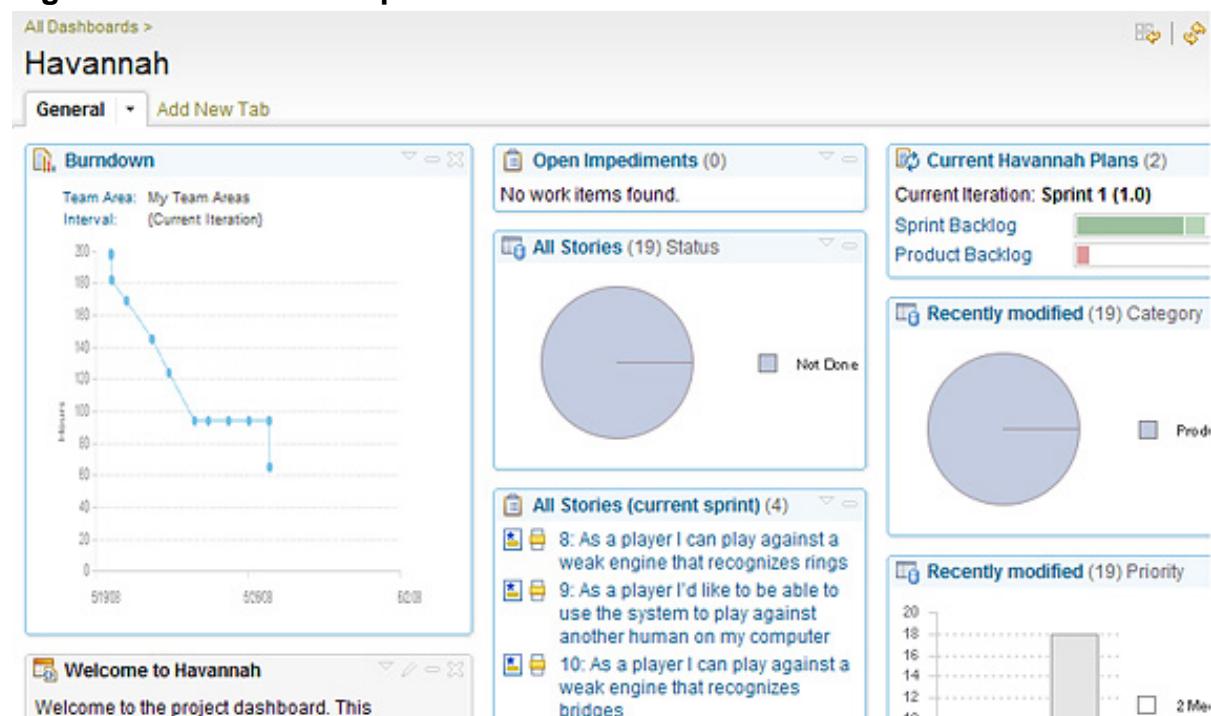
3. Scroll down the list of available viewlets, open the **Trend Reports** and **Work Items** folders, and select **Burndown** (Figure 52). The panel at right will display a description of any of the viewlets when they are selected.

Figure 52. Adding the Burndown Report

This screenshot shows the Rational Team Concert interface with the 'Havannah' dashboard open. The left sidebar shows categories like 'About Me', 'Project/Team', 'Tools', 'Trend Reports', and 'Work Items'. Under 'Work Items', the 'Burndown' viewlet is selected. The main panel displays the 'Burndown' viewlet details: it applies to 'Projects & teams', its description is 'The 'Burndown' viewlet shows the number of estimated hours required to complete work items that are either open or in progress over daily intervals. The target users for this viewlet are PMCs and Development Leads.', and it includes a preview chart showing burndown progress over time.

4. Click **Add Viewlet**, and the Burndown Report will be added the dashboard (Figure 53).

Figure 53. Burndown Report in the Dashboard



Now you have the two most important artifacts for your Scrum team right at the top of the dashboard. Exploring the Reports tab may turn up some other interesting reports that are worth adding, but this will be enough to keep on track. As you look at ways to use Rational Team Concert and Jazz to support other aspects of your project, you can add **Build Health** and **Code Coverage** and **Test Failure** reports to the dashboard. It would not be unreasonable to expect team members to have this page as their browser home page.

The dashboard Summary items also include links to the full detail. Clicking on the **Recently Modified** query link near the bottom (Figure 54) will move to the Work Items tab and display the full results for that query.

Figure 54. Viewing results of Recently modified query

The screenshot shows the Rational Team Concert interface. At the top, there's a navigation bar with links for Project Areas, Dashboards, Work Items (which is the active tab), Iteration Plans, Source Control, and Reports. On the right of the top bar, it says "Your Trial License expires in 131 days | ADMIN | Log Out" and "Havannah". Below the navigation bar is a search bar and a toolbar with icons for Create Work Item, Create Query, and Recently Viewed.

The main content area is titled "Recently modified" and displays a table of work items. The table has columns for Type, Id, Summary, Owned By, Status, Priority, Severity, and Modified Date. There are 8 items listed:

Type	Id	Summary	Owned By	Status	Priority	Severity	Modified Date
	43	Engine can find path from one corner to another (that is, form a bridge)	Allan	In Progress	Medium	Low	2 minutes ago
	34	Have the move engine try to block a human player who is trying to make a ring	Allan	In Progress	Medium	Low	2 minutes ago
	31	Have move engine pursue a ring even if the human player tries to block it	Allan	Resolved	Medium	Low	3 minutes ago
	33	Automate test cases for trying to make a blocked ring	Prasad	In Progress	Medium	Low	3 minutes ago
	32	Identify test cases for trying to make a blocked ring	Prasad	Resolved	Medium	Low	4 minutes ago
	30	Write automated tests for unblocked rings	Prasad	Resolved	Medium	Low	4 minutes ago
	38	Draw empty board	Sasha	In Progress	Medium	Low	5 minutes ago
	27	Code state management classes	Sasha	In Progress	Medium	Low	5 minutes ago

This can be useful to review before the Daily Scrum, and it can be a good way for someone who is coming back from vacation or a business trip to quickly catch up. Also notice that work items can be created from the Work Items tab by using the link at the top of the left column. Thus, anyone with Web access to your Project Area can add Defects or User Stories or Enhancement Requests easily.

Team members can also update their progress and add comments to their work items from this view. Clicking on a Work Item brings up its information (see Figure 55).

Figure 55. Viewing a Work Item

Task 47

Summary: * Automate tests for forming a bridge around obstacles

Resolved Fixed

Overview Links Approvals History Loaded: Jun 7, 2008 11:18 AM

Details

Type:	Task	Owned By:	Prasad
Severity:	Normal	Priority:	Unassigned
Found In:	Unassigned	Planned For:	Sprint 1 (1.0)
Creation Date:	May 19, 2008 5:32 PM	Estimate:	4 hour Correction
Created By:	Sasha	Time Spent:	6 hours
Project Area:	Havannah	Due Date:	
Team Area:	Havannah Team	Resolution Date:	May 28, 2008 5:37 PM
Filed Against:	Product Backlog	Resolved By:	Prasad
Tags:			

Quick Information

Parent (1): 10

Description **Preview**

The Iteration Plans tab provides a different view, which is focused on the Stories and the team members: Click **Iteration Plans > Sprint Backlog > Planned Items > Group By Owner**.

In the official release of Rational Team Concert, you can also show Progress bars or [work] Load bars for the team members (Figure 56).

Figure 56. Viewing iteration plans

Sprint Backlog

Team Area: Havannah Team | Iteration: Sprint 1 (1.0) (5/19/08 - 5/30/08) | 19 Closed
15 Open

Progress: 154 / 206 | +24 h Estimated: 100%

[Overview](#) [Planned Items](#) [Charts](#)

Exclude:

Group By:

Sort By:
 Priority
 Estimate

	Allan Closed Items: 5 Open Items: 3	<div style="width: 75%; background-color: #669933; height: 10px;"></div>	Load: 21 / 32 +11 h	Estimated: 100%
	Delaney Closed Items: 0 Open Items: 0	<div style="width: 0%; background-color: #CCCCCC; height: 10px;"></div>	No Work Planned	Estimated: 0%
	Frank Closed Items: 0 Open Items: 0	<div style="width: 0%; background-color: #CCCCCC; height: 10px;"></div>	No Work Planned	Estimated: 0%
	Prasad Closed Items: 8 Open Items: 3	<div style="width: 25%; background-color: #669933; height: 10px;"></div>	Load: 11 / 32 +21 h	Estimated: 100%
	Rose Closed Items: 4 Open Items: 2	<div style="width: 40%; background-color: #669933; height: 10px;"></div>	Load: 10 / 24 +14 h	Estimated: 100%
	Sasha Closed Items: 2 Open Items: 2	<div style="width: 10%; background-color: #669933; height: 10px;"></div>	Load: 8 / 32 +24 h	Estimated: 100%

Keep going

Even though you've covered a lot of ground in this article, there's so much more to Rational Team Concert and Jazz than Scrum-based agile planning, and there's even more to agile planning than has been described here. For example, consider using Jazz source control and the Jazz build engine to power your continuous integration efforts. For team members who are not developers, introduce them to the Web UI for checking work items and project status. Your managers will become addicted to the Web dashboards. Try the various reports to see which ones fit how your team manages their process.

When you get in, look around. You are likely to be pleasantly surprised over and over again.

Share this article



Digg
this
story



Post



Acknowledgements

It's hard to give credit where it is due when there are many to thank, but the Scrum section of Jazz.net certainly draws from the contributions of Mike Cohn of Mountain Goat Software and from Ken Schwaber, who co-founded the Scrum Alliance (see Resources) in 2002, plus the work of Jeff Sutherland and Ken Schwaber before that. Many thanks also to Erich Gamma, Dirk Baeumer, and Scott Malabarba for reading and providing helpful comments on this article.

Downloads

Description	Name	Size	Download method
Rational Team Concert Scrum sample data	ScrumSampleData	5.4 KB	HTTP

[Information about download methods](#)

Resources

Learn

- Get Jazz and Rational Team Concert downloads and updates and participate in the forum discussions at [Jazz.net](#). Registration required (free). This is also where you can enter and review enhancement requests and bug reports.
- Learn more about the Scrum process at any of these sites:
 - [Scrum Alliance](#)
 - [Mountain Goat Software](#)
 - "[All about Agile: How to Implement Scrum in 10 Easy Steps](#)" by Kelly Waters on the Agile Development site
- Practice continuous improvement using the IBM Rational Self Check for Software Teams (previously referred to as Team Pulse) for your reflections. Start by reading [Introducing IBM Rational Self Check for Software Teams](#) by Per Kroll and William Krebs (IBM developerWorks, June 2008).
- Learn more about Agile estimating and planning:
 - Visit [Mike Cohn's Web site](#)
 - Read his books:
 - [Agile Estimating and Planning](#) (Prentice Hall, 2005)
 - [User Stories Applied: For Agile Software Development](#) (Addison-Wesley Professional, 2004)
- Read about stakeholder goals documents and outside-in development in Carl Kessler and John Sweitzer's Jolt Award-nominated book, [Outside-in Software Development: A Practical Approach to Building Successful Stakeholder-based Products](#) (IBM Press, 2007).
- Visit the [Rational software area on developerWorks](#) for technical resources and best practices for Rational Software Delivery Platform products.
- Explore [Rational computer-based, Web-based, and instructor-led online courses](#). Hone your skills and learn more about Rational tools with these courses, which range from introductory to advanced. The courses on this catalog are available for purchase through computer-based training or Web-based training. Additionally, some "Getting Started" courses are available free of charge.
- Subscribe to the [Rational Edge newsletter](#) for articles on the concepts behind effective software development.

- Subscribe to the [IBM developerWorks newsletter](#), a weekly update on the best of developerWorks tutorials, articles, downloads, community activities, webcasts and events.
- Browse the [technology bookstore](#) for books on these and other technical topics.

Get products and technologies

- Download [trial versions of IBM Rational software](#).
- Download these [IBM product evaluation versions](#) and get your hands on application development tools and middleware products from DB2®, Lotus®, Tivoli®, and WebSphere®.

Discuss

- Participate in the [discussion forum for this content](#).
- Check out [developerWorks blogs](#) and get involved in the [developerWorks community](#).

About the author

Millard Ellingsworth

Millard Ellingsworth develops software and encourages adoption of agile development as a member of the IBM Enterprise Content Management group. He serves his team as ScrumMaster and others as an Agile Coach and workshop leader. As a musician, he is not particularly fond of jazz, but as a software developer, he quickly realized what a game-changing opportunity it represents.

Trademarks

This is the first trademark attribution statement.

This is the second trademark attribution statement.