

Scrum project management with IBM Rational Team Concert Version 2: Part 1. Set up projects, teams, and plans

Jazz technology helps in agile planning and management

Skill Level: Introductory

[Millard Ellingsworth \(millarde@us.ibm.com\)](mailto:millarde@us.ibm.com)

Software Developer
IBM Corporation

[Thomas Starz \(starz@de.ibm.com\)](mailto:starz@de.ibm.com)

Software Developer
IBM Corporation

08 Oct 2009

For more than a year now, we've been using IBM® Rational Team Concert™ to support our scrum teams, delighting in many of its features, living with some of its shortcomings, and looking forward to the next version. With Version 2, the Jazz and Rational Team Concert teams have delivered dramatic improvements to their scrum and agile estimating and planning support (not to mention a much improved Web client and many other new features). This article series provides an up-to-date tutorial on getting started with Version 2 and the scrum process and highlights new features and capabilities important to scrum teams and their managers. It updates the article published in 2008 that covered [Rational Team Concert Version 1 and scrum project management](#).

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. The purpose is 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 manage your scrum project with IBM® 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 book, *Agile Estimating and Planning* (Prentice Hall, 2005), which describes a scrum team at Bomb Shelter Studios as they work on a new computer game called Havannah. The data for the example is in the file that accompanies this article, "Rational Team Concert scrum sample data" (see [Downloads](#)).

This article is based on the assumption 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 IBM® 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
 - Scrum master
 - Team member
- Ceremonies (gatherings or meetings)
 - Daily Scrum
 - Sprint Planning
 - Sprint Review
 - Sprint Retrospective
- 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. This person 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 **scrum master** manages the scrum process, ensuring that the scrum practices are properly carried out 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; otherwise, the need to keep part-timers up to speed will slow down the rest of the team.

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. This is 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 so that the team can determine whether the items can be completed in time. 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 scrum master or product owner. The Sprint Backlog represents the work that the team has committed to complete during the sprint.

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

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

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 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 sprints, because they work to continually improve their team and practice effectiveness. Given that Rational Team Concert is very configurable and adaptable to the team's process adjustments, these meetings may result in tweaks and improvements that are implemented by changing the process in Rational Team Concert.

As items from the Product Backlog are selected for completion during the sprint planning meeting, 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. This burndown chart shows the amount of work remaining in the project in a graphical display. It is also common for it to include an "ideal" line that indicates a smooth progression of the work from start to finish of the sprint. The *ideal* line indicates how work would progress if even proportions of it are completed each day of the sprint. Although the actual work progress will likely not follow this line, significant deviations are a cause for concern, particularly if the actual progress continues to trend away from the idea.

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

The Rational Team Concert 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.

When managing a project by using the scrum method, you need a place to keep the description of the things that need to be done for the project, namely your **stories** and **tasks**. Although you can simply put cards on a bulletin board, using software might be more desirable, especially if the team members are distributed across multiple locations.

Rational Team Concert provides lightweight ways to manage all that you need. In a **Project Area**, you can manage your builds, sources, plans, and much more. This article focuses on the planning and management aspects of the scrum process. After performing first steps, such as connecting to a repository and defining user IDs for the people who use the tool, we explain how to create **plans** and **work items** and manage them during the iterations of a project release.

Set up the Rational Team Concert repository and users

Before you're ready to look at how Rational Team Concert and the Jazz Scrum Process template support scrum teams, you need to handle the basic project setup.

Create a repository connection

1. Be sure that your Jazz server is started and that your Rational Team Concert client is open.
2. To create a new repository connection, right-click on **Repository Connections** in the Team Artifacts view (see Figure 1) and then select **New > Jazz Repository Connection**.
3. Enter the URI and the user ID and the corresponding password.

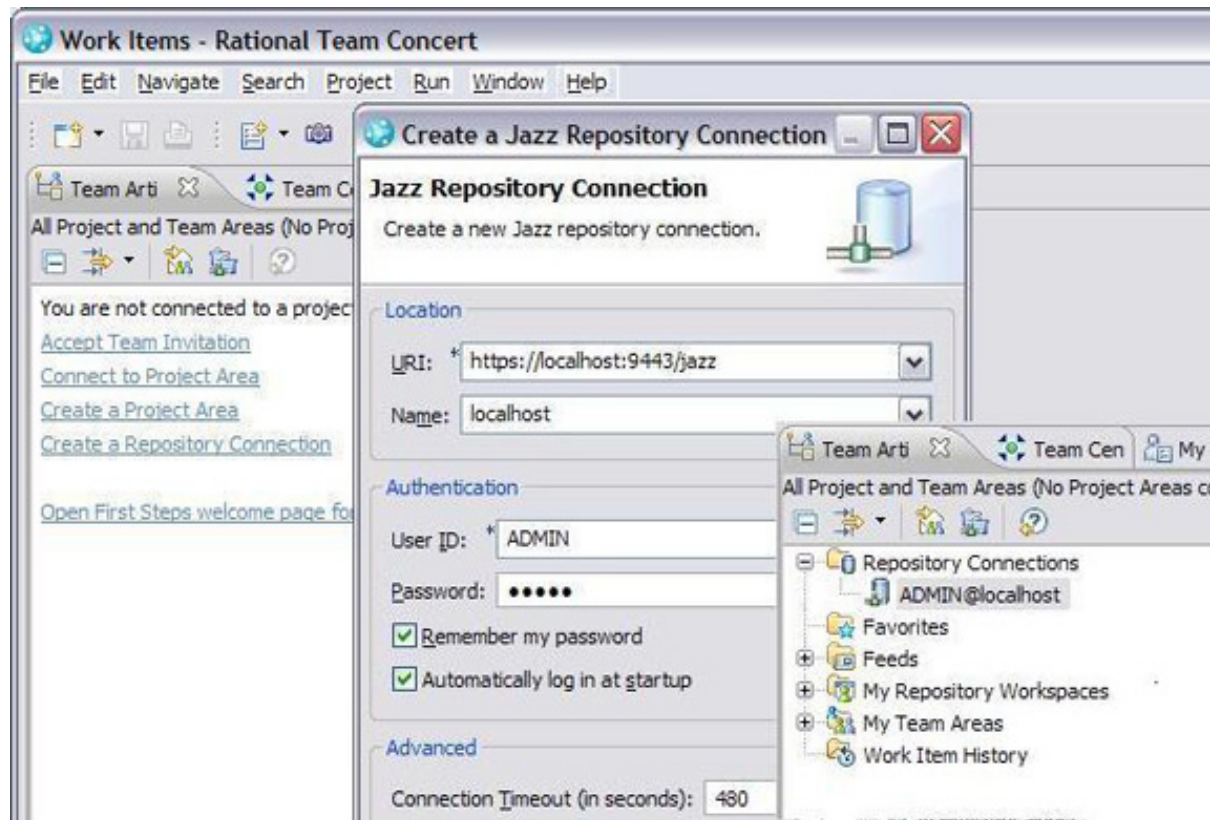
Tip:

When you use an installation of a Jazz server on your local system, the URI might be, for example, `https://localhost:9443/jazz/`, and the initial user ID is `ADMIN` with `ADMIN` as the password, too (both are case-sensitive).

Note:

Notice the lowercase letter S in "https," which shows that Jazz uses a secure connection. Depending on your browser, you could get odd behavior if you forget to include the "s."

Figure 1. Creating a new repository connection



4. Click **Finish**.

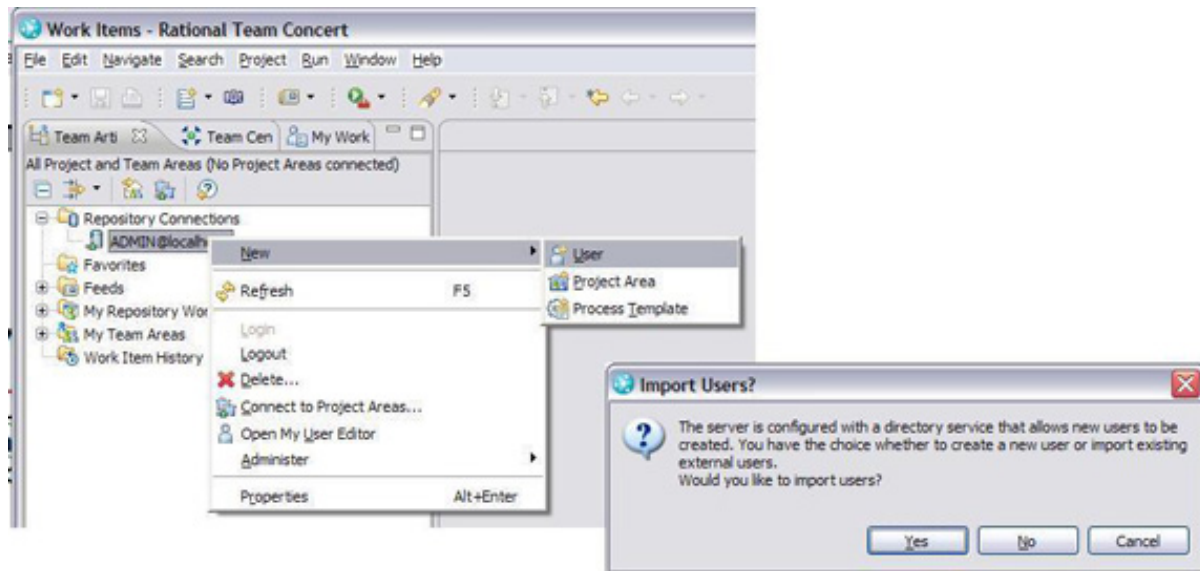
Depending on the system where your Jazz server is installed, it may take some time when you connect to it the first time until the repository connection is created.

Add or import users

The next action is to create the users in Rational Team Concert. Users belong to the Jazz repository, so they can participate in more than one Project Area and, potentially, use other Jazz-based tools hosted on the server. If you configured Jazz with an external user repository (such as your corporate LDAP server), you can import users without having to create them, as the sidebar explains. Here, let's assume that you used the default Tomcat application server and that you need to create users in this new repository.

1. To create a new user, right-click on the repository connection and then select **New > User**.
2. Select **No** when prompted about whether to import users (Figure 2).

Figure 2. Creating a new user



Larger view of Figure 2

3. Enter the user information for the first user.
 - a. Because we are following Mike Cohn's example, Sasha is the first user.
 - b. Optionally, add a photo if one is available.
 - c. The e-mail address is a required field, because Jazz can send e-mail notification of many of the team's activities. For Sasha, enter sasha@bombshelterstudios.com.
 - d. The user's initial password is automatically set to equal to the user ID.
4. Choose the Jazz security groups appropriate for the user. The user who will administer the project must be in the repository group called JazzAdmins (in the example, Sasha must become a JazzAdmin; otherwise, he will not be able to perform all necessary configuration activities).
 - **JazzUsers** 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.
 - **JazzAdmins** can access the various settings related to the Jazz server and projects.
5. Select the appropriate type of license for the user.

- **Developer** licenses are required for users who create or deploy process templates, create Project Areas or plans, and create or edit attached pages. Developer licenses are also appropriate for team members who will be contributing code and running builds.
- **Build** and **ClearCase Connector** licenses are typically assigned only to administrative users.
- **Contributor** licenses are a good choice for all users who need only read access to the repository. With a contributor license, a user can also create work items.

Figure 3 shows the resulting screen.

Figure 3. Specifying user details

The screenshot displays the 'User' configuration window for a user named 'Sasha'. The interface includes a left-hand navigation pane with options like 'Repository Connections', 'Favorites', 'Feeds', 'My Repository Workspaces', 'My Team Areas', and 'Work Item History'. The main content area is divided into several sections: 'Details' with input fields for 'User ID (case sensitive): sasha' and 'Email Address: sasha@bom', a 'Reset Password...' button, and a note about the default password; 'Photo' with a user photo and 'Browse...' and 'Remove' buttons; 'Repository Groups' with a list of groups (JazzAdmins, JazzDWAdmins, JazzGuests, JazzUsers) and 'Select All'/'Deselect All' buttons; 'Team Areas' which is currently empty; and 'Client Access Licenses' with a list of licenses, where 'Rational Team Concert - Developer [9 of 10 r' is selected. At the bottom, there are tabs for 'Overview', 'Work Environment', 'Scheduled Absences', and 'Mail Configuration'. A 'Save' button is located in the top right corner.

6. Click **Save**.

Note:

Later, users need to configure their work environments and enter their scheduled absences (see the tabs at the bottom of the screen) so that the amount of time that they can spend on a project can be calculated properly. We show an example of this later.

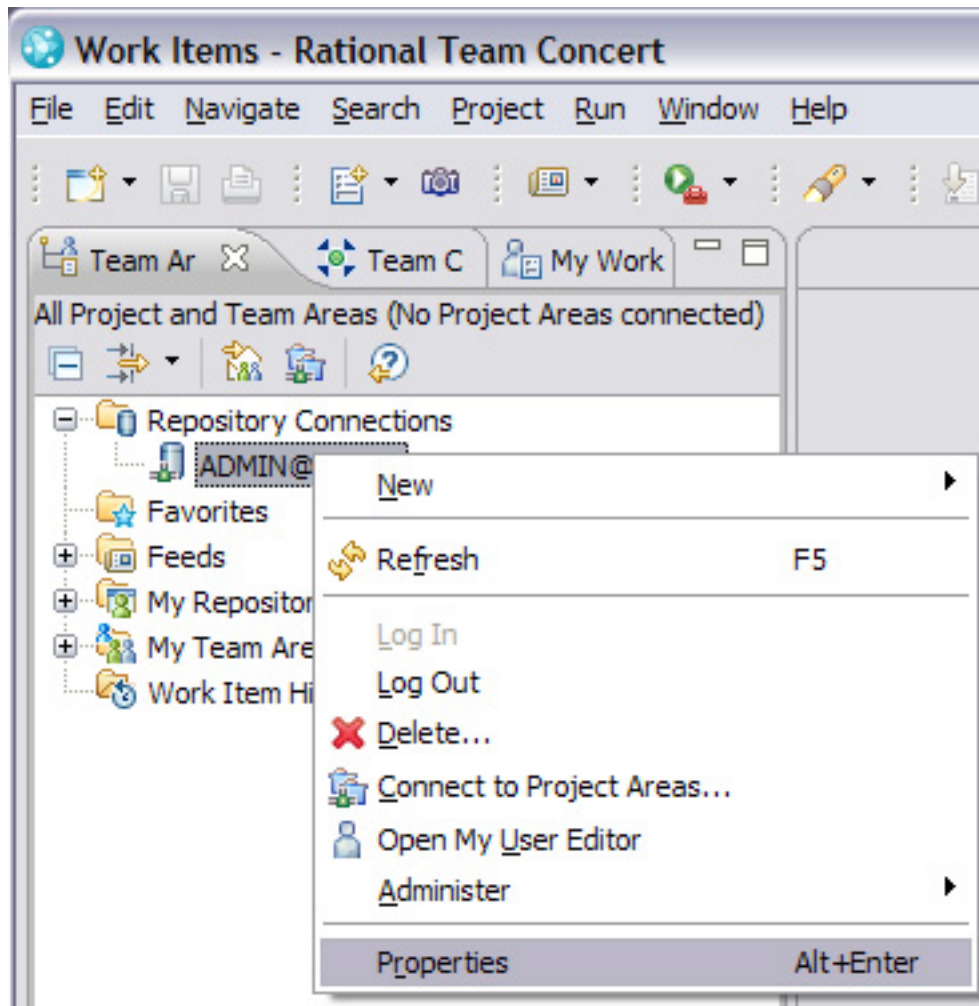
For the Havannah project that is used in this example, more users need to be added in the same way as described for Sasha.

7. Create additional users named Allan, Delaney, Frank, Prasad, and Rose.
8. Assign them to the JazzUsers group with a Developer license.
9. Log out as the ADMIN user.

You've now finished entries as the default ADMIN user. The Jazz security model gives this user basic admin privileges in Jazz but no particular rights within a given Project Area. For example, the ADMIN user cannot create and save new iteration plans in the Project Area, because this is for your scrum master or product owner to do. You are not likely to need the repository connection for the ADMIN user again; therefore, you can reuse it for Sasha.

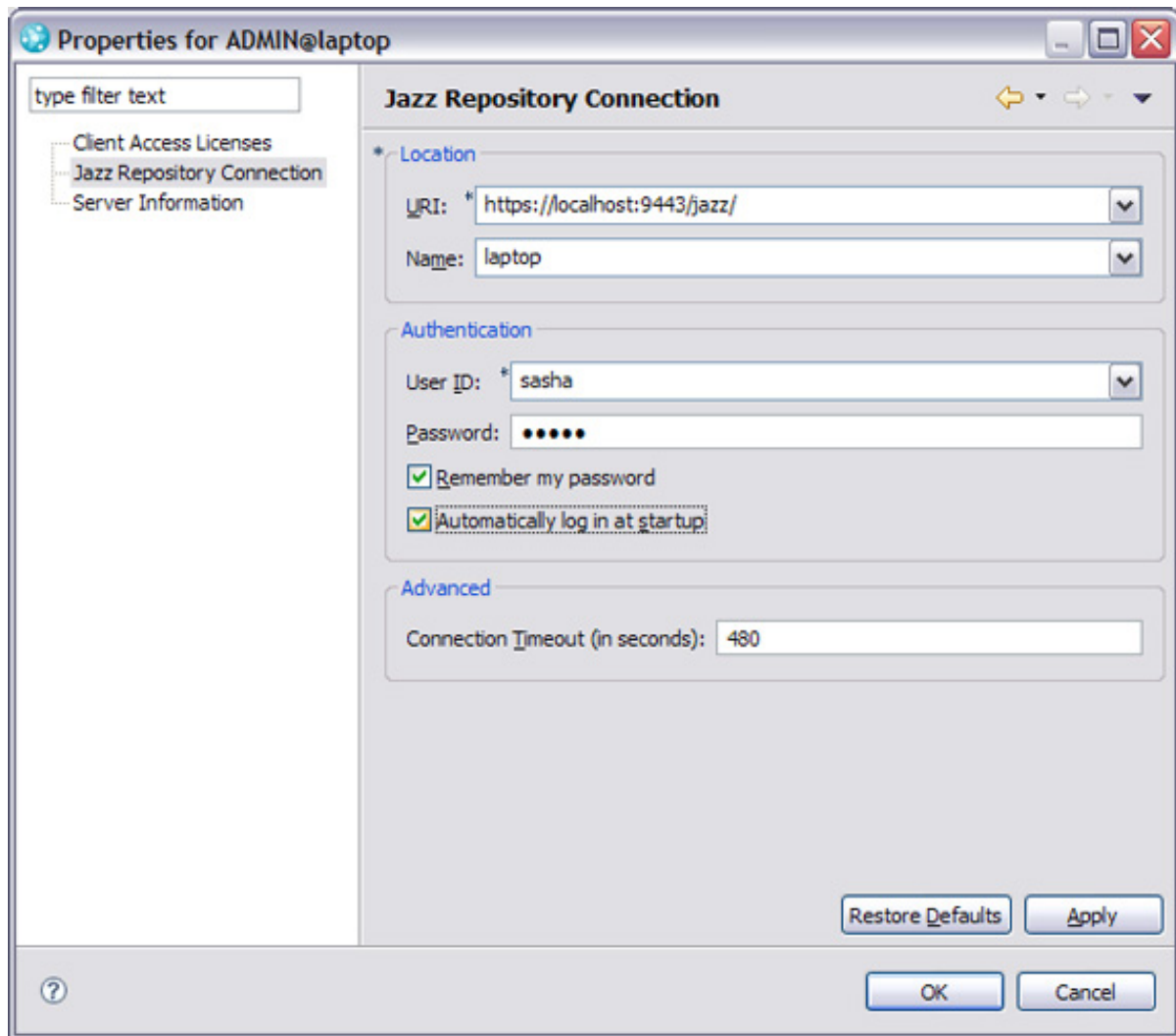
10. Right-click the repository connection and select **Properties** (Figure 4).

Figure 4. Specifying login properties



11. Change the User ID and Password fields to match Sasha's credentials (sasha and sasha, because passwords default to the same value as the user ID). Optionally, you can select (or deselect) the "Remember my password" and "Automatically log in at startup" check boxes, as shown in Figure 5.

Figure 5. Specifying login properties



How to import user data from your LDAP directory

Users can also be imported from an external user repository (such as your corporate LDAP server). For more information on how to connect your team server to an LDAP directory, you can follow [LDAP Configuration for Newbies](#) on Jazz.net.

12. When you click **OK**, you will be logged back in as Sasha, who is Bomb Shelter Studio's scrum master.

Set up areas for your project and people

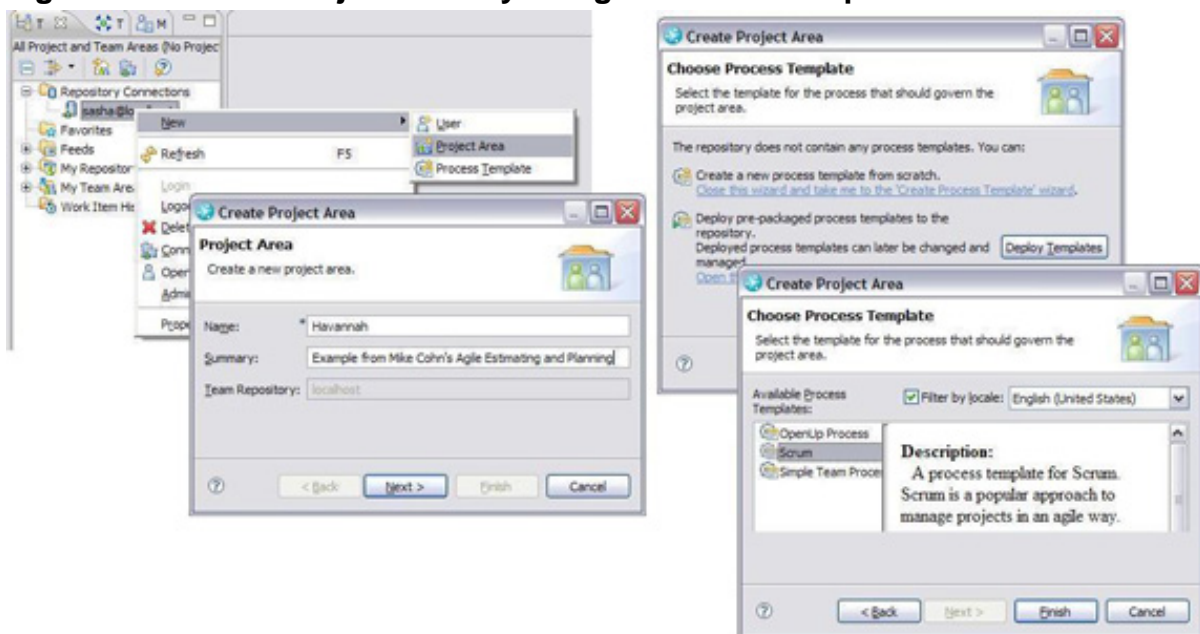
Create a Project Area

The next action is to create a Project Area, which will serve as the container for all

plans, work items, and other things related to the project that you are setting up.

1. To create a new Project Area, right-click your repository connection in the Team Artifacts view and then select **New > Project Area**.
2. Enter a name for the project (Havannah in this example), and click **Next**.
3. When you create a Project Area the first time, you need to deploy the pre-packaged process templates, so click **Deploy Templates**. This may take a few minutes to finish.
4. Under Available Process Templates, choose the **Scrum** template and click **Finish** (see Figure 6).

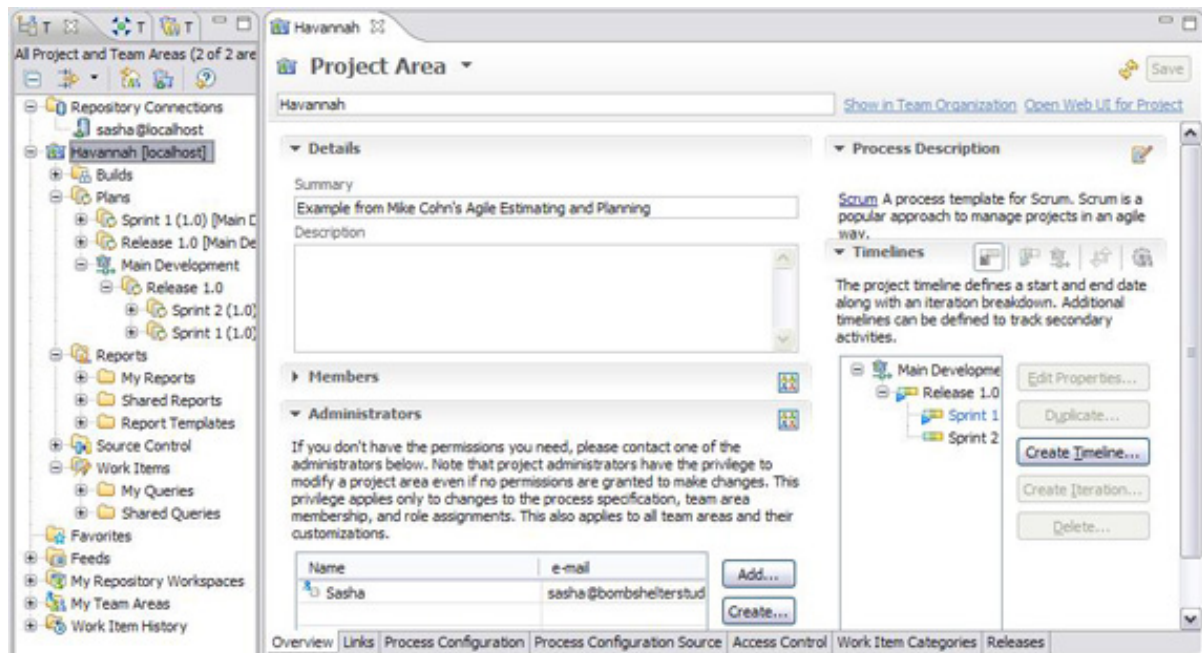
Figure 6. Create a Project Area by using the Scrum template



Larger view of Figure 6

It can take up to several minutes for the Project Area to be initialized on the server. When that is finished, you will see a screen similar to the one in Figure 7.

Figure 7. The Project Area for the Havannah project



Larger view of Figure 7

In the left column in the Team Artifacts view, you can see the new Project Area with folders for Builds, Plans, Reports, Source Control (streams), and Work Items (queries to help locate Work Items in various states). This article focuses on the Plans section. In the Project Area editor, there is a Description area for providing information about the project.

Note:

In the Project Area, there is also a section for adding attachments. This is intended for the Scrum Process template and is not a good place to put your project attachments.

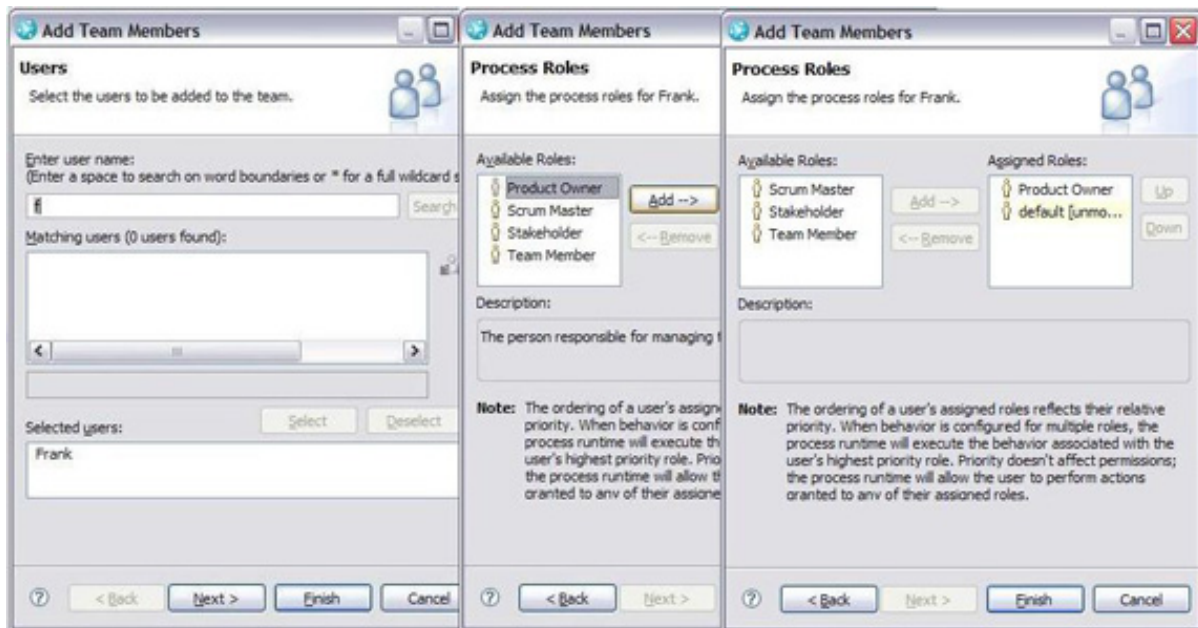
Add members and specify their roles

As the creator of the Project Area, you are the initial administrator. However, most of the permissions for modifying the project structure within the Scrum Process template belong to the scrum master or product owner. Therefore, one of the first things that you need to do is to add members to the project and set their roles (at least for the scrum master and product owner). Typically, Project Area-level membership is for members with management responsibilities. You will add other team members to a Team Area shortly.

In this example, Sasha created the Project Area, so he becomes its initial administrator and can now make any changes to the Project Area. Frank, the Havannah product owner, is an appropriate Project Area-level member, so add him now.

1. Click the arrow to the left of **Members** to expand that panel.
2. Click **Add** to start the Add Team Members wizard.
3. To add Frank as the product owner, type an **f** (letter F) in the user name text area and search for matching names. Because Frank was already added as a Rational Team Concert user, his name will be in the Matching Users list.
4. Click **Select** to move him to the Selected Users list.
5. Then click **Next**.
6. Select the **Product Owner** role, add it to the **Assigned Roles** list, and click **Finish** (see Figure 8).

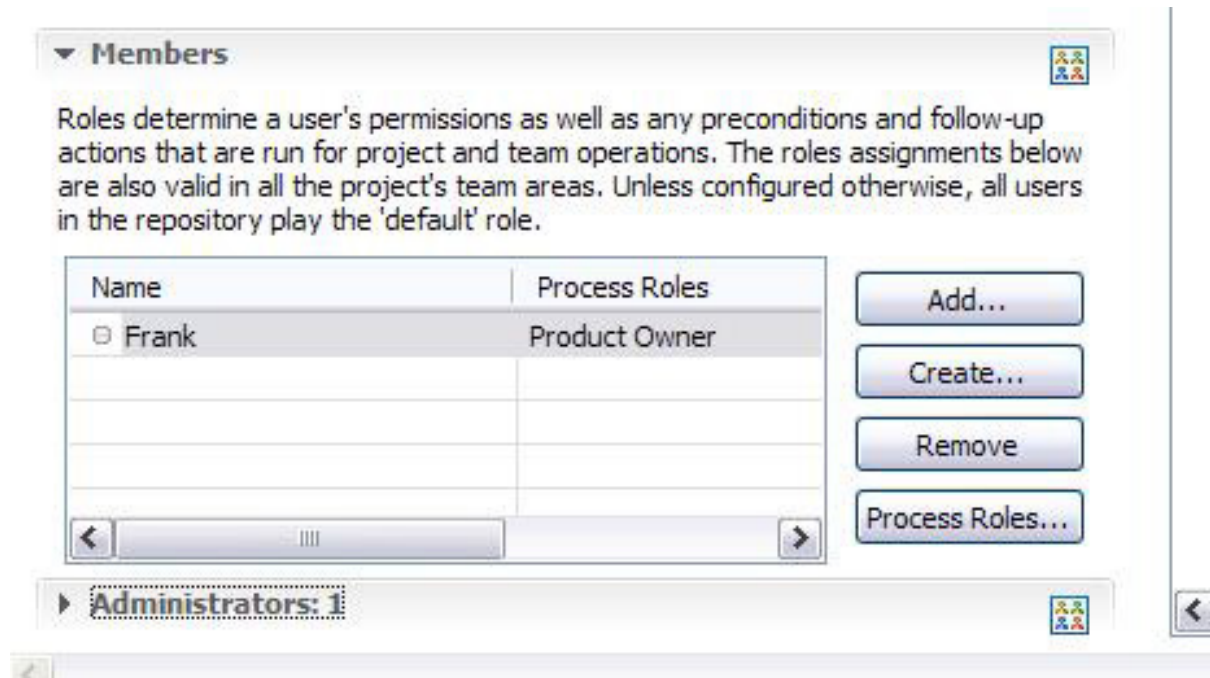
Figure 8. Add team members



Larger view of Figure 8

As Figure 9 shows, Frank is added as the product owner for the project.

Figure 9. Add members to the Havannah project



7. Save the Project Area (using either the **Save** button in the upper-right corner of the Project Area editor or the **File > Save** action).

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) and you agree to this, project members will get a welcome e-mail message, along with links and information for connecting to the project.

8. Because these users are fictitious, de-select all checkmarks and then click **OK**. (You can use Cancel from this dialog instead, because it is only for sending e-mail.)

Outline the release timelines

The next step is to plan the timelines for the project, which means that you specify the start and end dates for the release and the sprints. When setting up your own Project Area, adjust the dates to suit your needs. It is fine to select a start date in the future.

Tip:

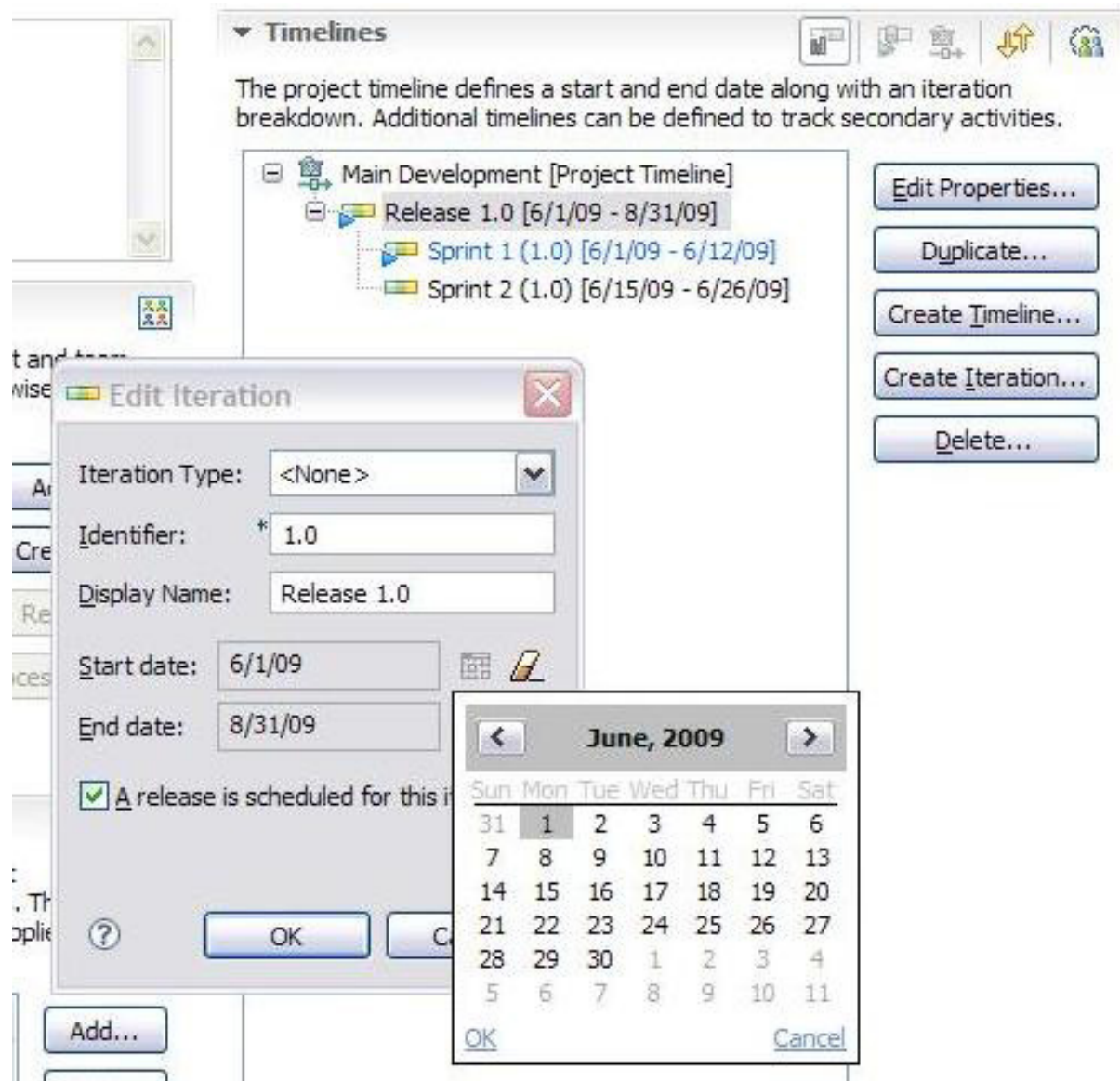
You need to be authorized to make changes to the project. The user who created the project has administrator rights and can make the changes described in this section. When setting up your own project, you must make sure that you are either administrator for the Project Area or a member who is in either the scrum master or

product owner role. Otherwise, you will get a permission-denied error when attempting to save changes to timelines.

On the right side of the Project Area editor window, under Timelines (see Figure 10), you can see the default placeholder iterations that the process template established when you created the Project Area, including Sprint 1 and Sprint 2 under Release 1.0.

1. Select **Release 1.0** under Process Iterations and click **Edit Properties** to open the **Edit Iteration** drop-down menu shown in Figure 10.

Figure 10. Specify release and sprint timelines



2. Leave the Iteration Type at the default value.
3. The Identifier must be unique, but you can change it to match your needs.
4. Set the Start and End dates for the release (see Note), make sure that the check box for "A release is schedule for this iteration" is checked, and click **OK**.

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 sprint should last two weeks (typically Monday to the following Friday, leaving out the non-work days).

Note:

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

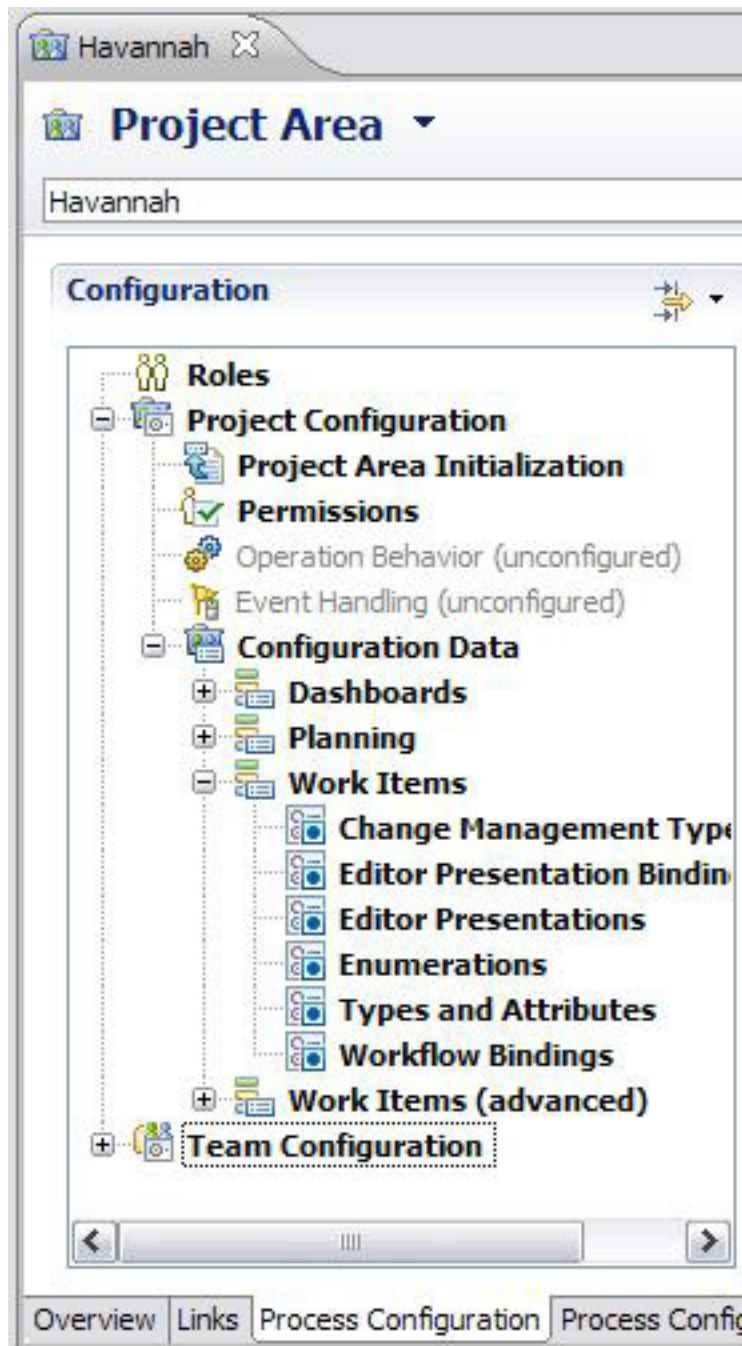
Find tasks automatically assigned to you

Some tasks are automatically created and assigned to the person who creates the Project Area. To list these, go to **Work Items > Shared Queries > Predefined**, and run the **Open assigned to me** query. You can follow these tasks to customize your Project Area.

Tips:

- You can make more changes to the Project Area later. To open it, right-click on it and select **Open** (double-clicking will only expand or contract it).
- The Project Area is configurable, to a large extent. On the Process Configuration tab, under Project Configuration, there are numerous options to adapt the project to your needs. As you become more familiar with Rational Team Concert, you will probably want to adapt it to your team (for example, as a result of discussions during your Reflections meetings). Process customization is a powerful tool for making the process fit your team's needs, rather than changing your team's behavior to fit the process.

Figure 11. Project configuration

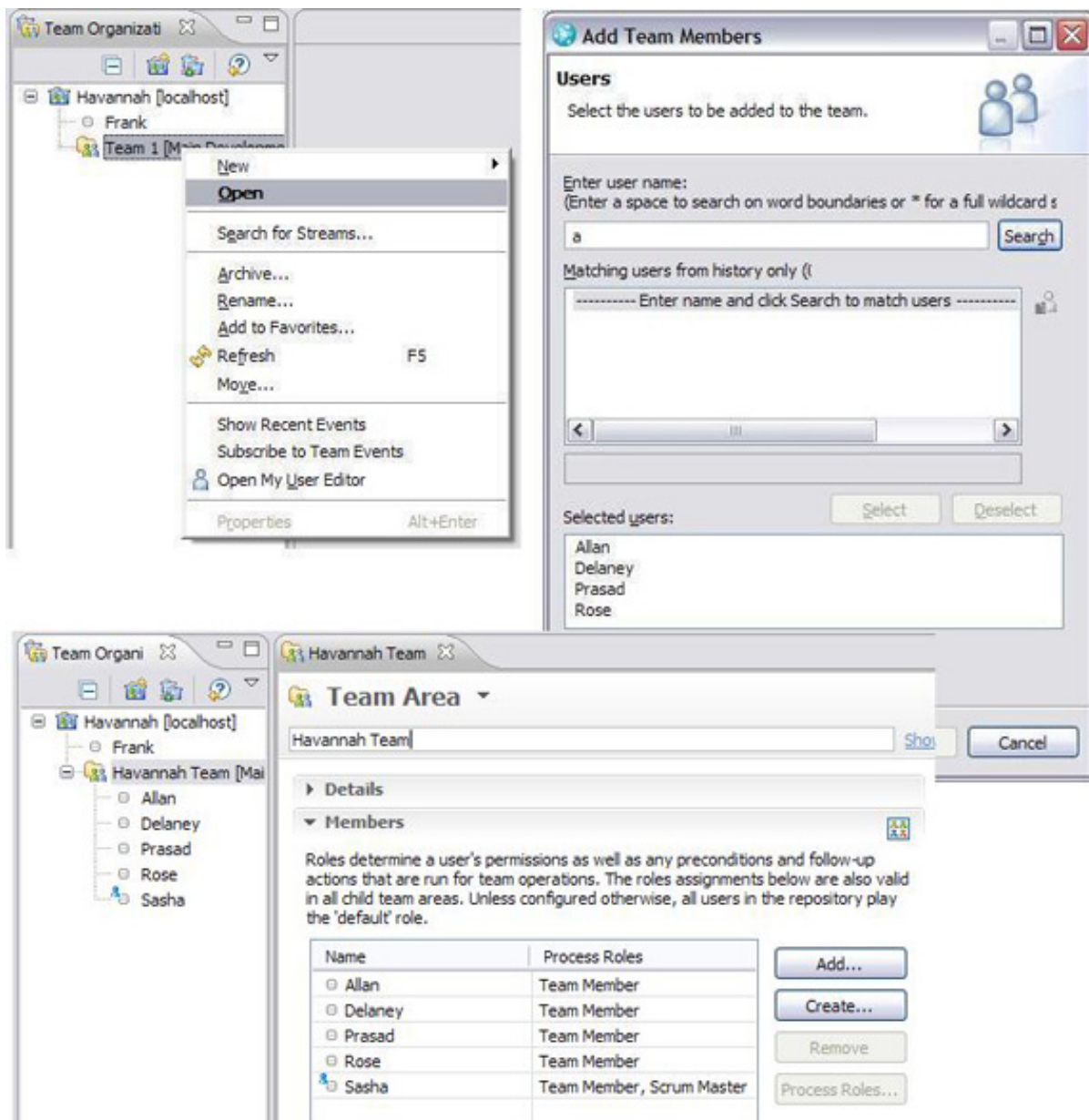


Specify a Team Area and add members

Jazz technology can support reasonably large teams with multiple timelines and subteams. 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 this example has just one team, it still needs to be populated with members.

1. Click the **Team Organization** tab in the left pane. If it isn't open, use **Window > Show View > Team Organization** to open it.
2. Expand the **Havannah** folder (which was automatically created when the Project Area was created).
3. The default name assigned to the team when the project is created is Team 1. Right-click on **Team 1** and open the Team Area editor.
4. Type `Havannah Team` as the name of the team.
5. Add **Allan**, **Delaney**, **Prasad**, and **Rose** as team members. Proceed the same way as described earlier when you added Frank to the Project Area. In the Add Team Members wizard, you can use the * (asterisk) wild card character to return all members (do not try that if you are using a corporate LDAP server) and select them all before assigning the Team Member role.
6. Add **Sasha** in the role of Scrum Master and, optionally, assign him the role of Team Member, also, because scrum masters are not automatically Team Members in the context of Rational Team Concert roles).
7. Click **Save**.

Figure 12. Add members to a Team Area

**Tip:**

Larger teams typically consist of several subteams. You can reflect your team structure by adding as many additional subteams as are appropriate for your project.

Specify team member availability

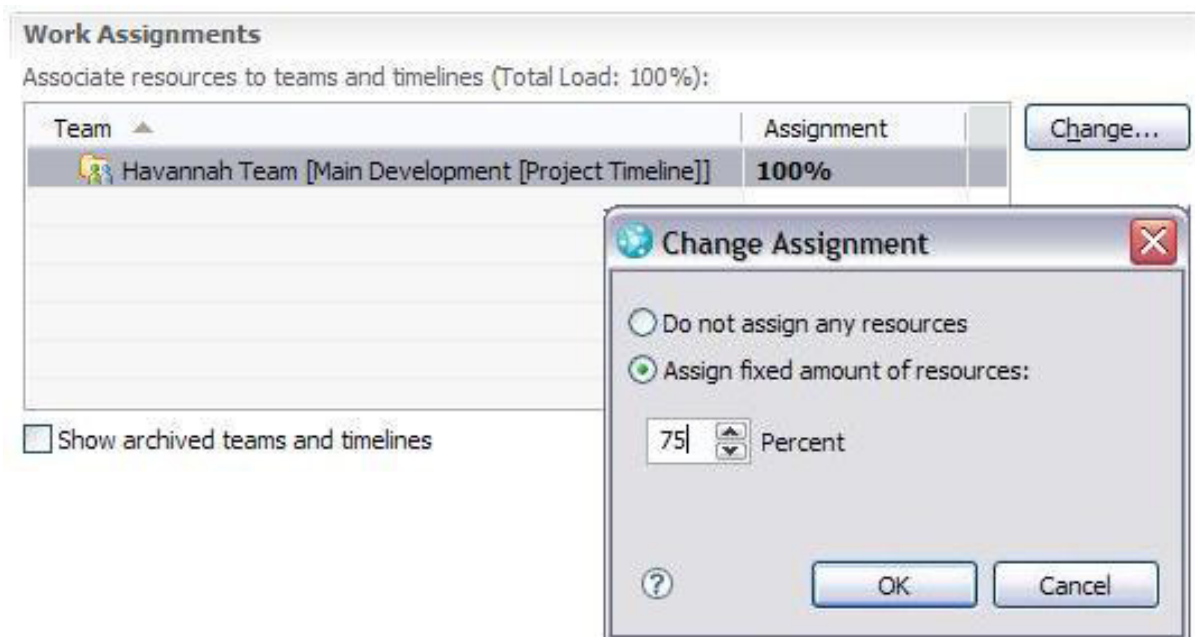
For team workload calculations to be accurate, adjust team members' availability. Perhaps a team member has responsibilities outside of the team that limits their participation or has a vacation planned. These are adjusted on the user's page. Typically, users would be expected to keep this area current themselves, but because it is an important part of calculating **Team Load** correctly, we'll illustrate it

here.

1. If you haven't already, click the **Team Organization** tab in the left pane and expand the **Havannah folder** and the **Havannah team**.
2. Open the user editor for **Rose**. You can double-click on a user name to open details for that user.
3. Rose is available to the team only 75% of the time. Therefore, click the **Work Environment** tab at the bottom and then, in the Work Assignment section, click the **Havannah Team** line and the **Change** option (see Figure 3).
4. By default, if a user is assigned to only one team, 100% of his or her time (resources) is associated with the team. To lower the assignment for Rose, select the assignment and click **Change**. Lower her assignment to 75% (see Figure 13).
5. Click **OK**.

This will adjust the number of hours that Rose is available for work assignments.

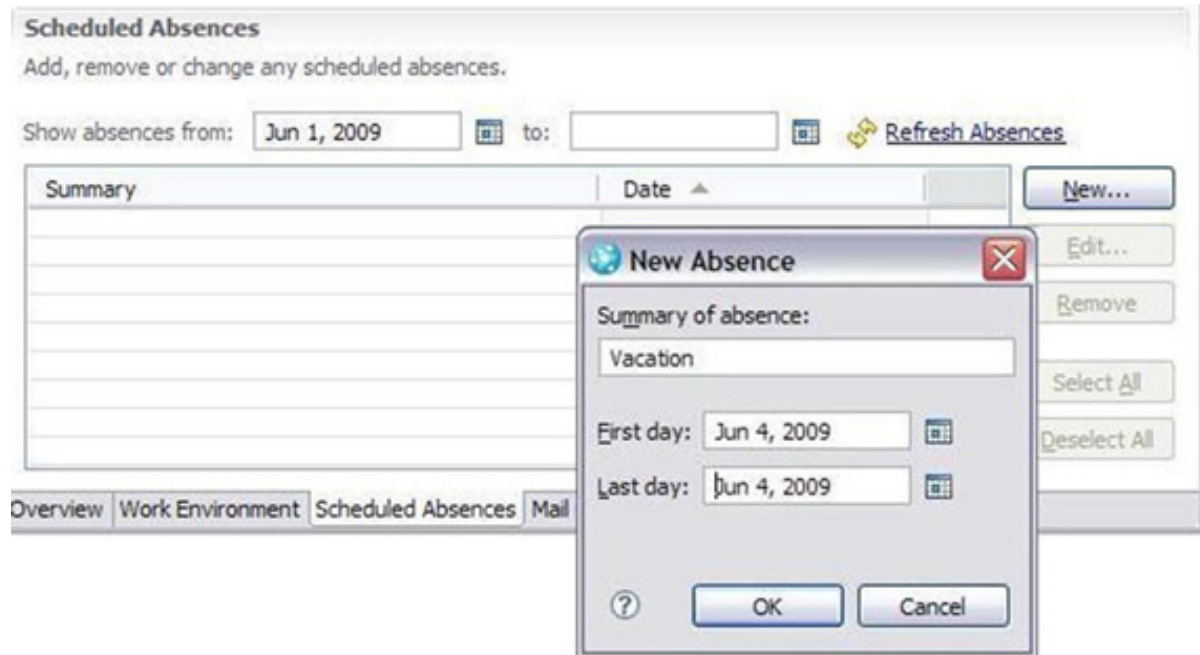
Figure 13. Adjust the work environment for a user



6. She also has a vacation day coming up. Switch to the **Scheduled Absences** tab, click **New** on the right side of the list, and enter a description and the dates for her vacation time (see Figure 14).

7. Click **OK**.
8. Click **Save** to update Rose's details.

Figure 14. Enter scheduled absences for a user



Create the Product Backlog for the project

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

1. To build it for this example, go to the **Team Artifacts** view (see the upper-left side of Figure 15, which follows) and, in the Havannah project, select **Release 1.0**, right-click to get the context menu, then select **New > Plan**.

Tip:

You might 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.

2. In the New Plan window that opens, enter **Product Backlog** for the name.
3. Select **Product Backlog** as Plan Type

4. For Team Members, Timeline, and Iteration, leave the defaults.
5. Click **Finish**.

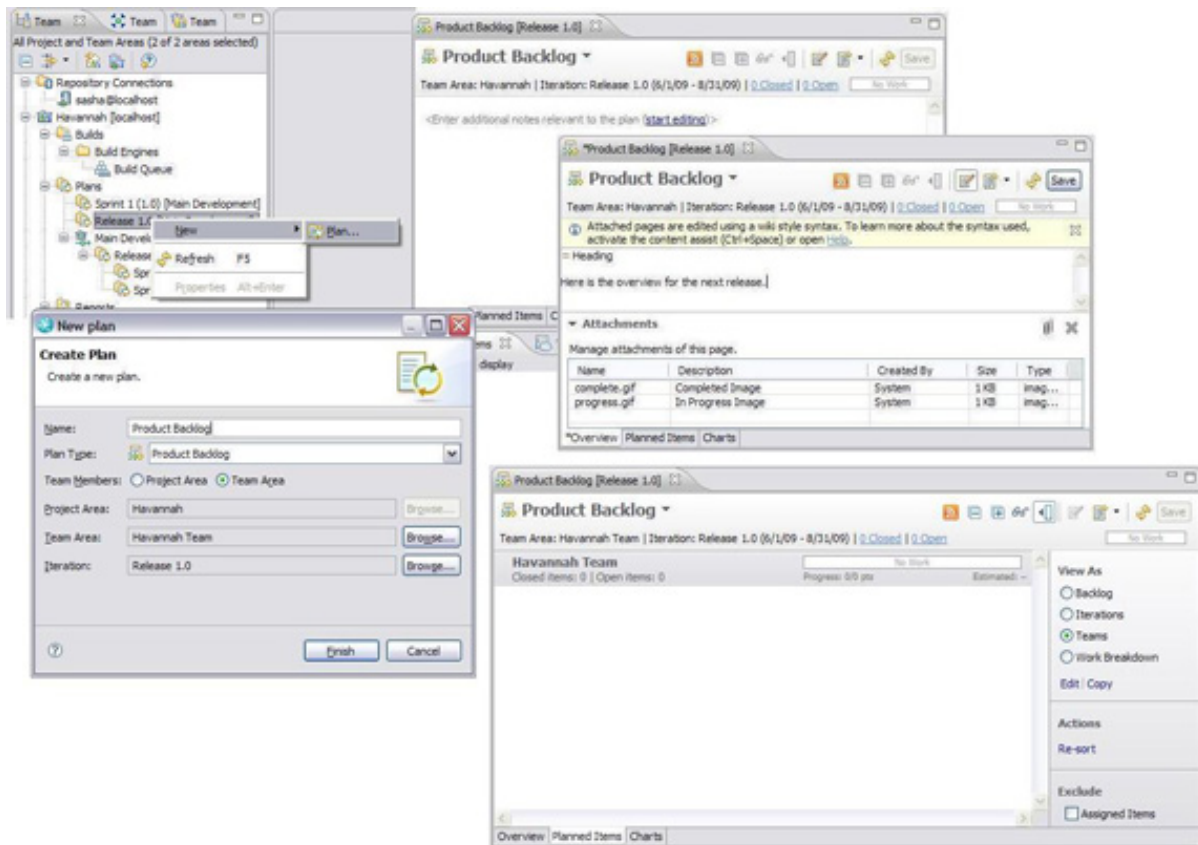
Tip:

If you cannot save the plan, you might have to go back and add the Scrum Master role for Sasha's settings.

A multipage editor will open for the Product Backlog iteration plan (see Figure 15).

6. On the **Overview** page, you can enter information about your product by using wiki-style formatting.
7. The Attachments area at the bottom of the editor view (visible only when in editing mode for the page) 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. The page area can also contain links to Web sites or external document repositories as needed.
8. Click the **Planned Items** tab to begin adding backlog items.

Figure 15. Editing the backlog



Larger view of Figure 15

There are several ways to view a plan, called *modes*. On the right side of the window, you can specify how the items in the plan should be grouped and what kind of items should be excluded from the view. There are predefined view modes, namely Backlog, Iterations, Teams, and Work Breakdown. You can also define your own customized view mode. To do this, you can use the **Edit | Copy** option on the right- panel, under "View As." Additional modes are stored on the server and will be available to other team members, and changes to the modes that are saved will affect all users of the Project Area.

Group plan items by type

For Rational Team Concert Version 1.0, the items of a plan could be grouped only by Owners, Categories, Tags, Planned Time, or Folders. The features of Version 2.0 are much more powerful. Although the earlier grouping modes can still be configured for V2.0, too, it is better to use only these newer modes:

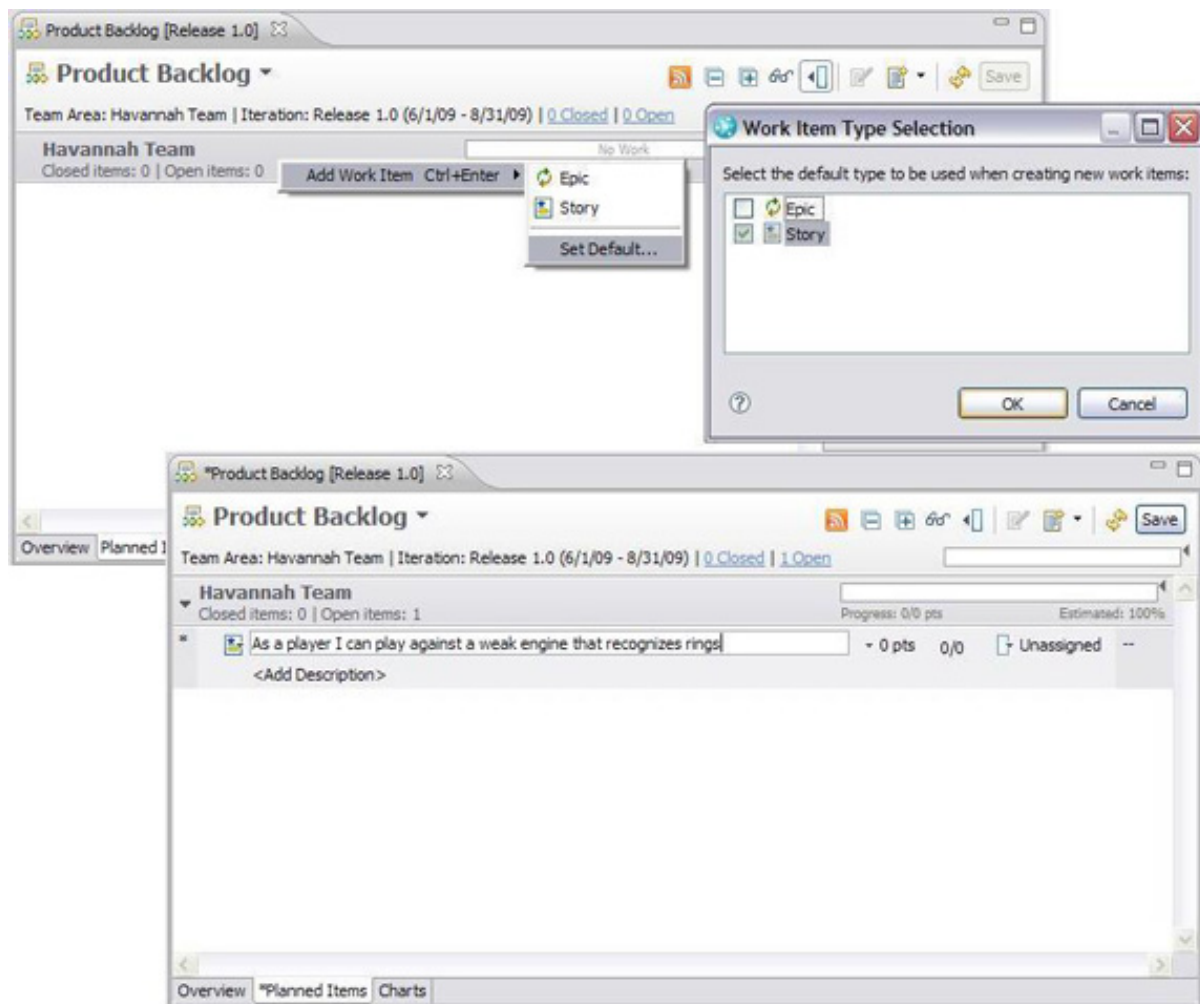
- Backlog
- Iterations
- Teams
- Work Breakdown

Fill the Product Backlog with work items

On the Planned Items tab of the backlog, begin adding backlog items. For scrum project management, the items in the product backlog are called *stories* or, if they are larger, *epics* (also see Mike Cohn's books, *User Stories Applied* or *Agile Estimating and Planning*, both cited in Resources). A *story* is a description of functionality to be implemented in the product, often stated in terms of what user roles the feature supports and what business value it offers them. A properly described story includes a summary or headline, some description or discussion of the feature, and conditions for the feature's acceptance by the Product Owner. In the Havannah example, all items are phrased as stories, so you fill your backlog with stories. (For brevity, most stories used in this example include only the summary, but don't confuse this with a properly constructed story.)

1. Set the default for new items to **Story**. This allows you to quickly enter new work items by using **Ctrl+Enter** as a keyboard shortcut.
 - A. Right-click on the **Havannah Team** folder.
 - B. Select **Add Work Item > Set Default > Story**.
2. Right-click on the **Havannah Team** folder again, select **Add Work Item** and then **Story** (or use Ctrl-Enter).
3. Enter the summary for the new item as the lowest screen in Figure 16 shows.

Figure 16. Planned Items in the backlog



Larger view of Figure 16

5. Add all further items as requested by the product owner.
6. Click **Save** on the upper-right of the Plan tab at any time to save your work.
7. After the stories are added, the next step for the product owner is to set priorities for them. The values of High, Medium, and Low are available for the priority attribute. You can assign priorities easily by using the drop-down menu embedded in the list items (see Figure 17).

Tip:

If the predefined priorities are not appropriate for your project needs, you can easily customize the list of available priorities in the Project Area.

8. You then also rank the items within the priorities. You reflect this ranking by dragging the items to order them within their priority grouping. The system remembers ordering done with the drag-and-drop action (or the use of the Alt+Cursor Up and Down shortcut). If you drag a story, such as a medium-priority story, to a different grouping and drop it between two stories of different priorities, Rational Team Concert automatically reassigns the priority value to match its neighbors.

Figure 17. Set priority for work items



9. When the team adds an estimate for the item complexity, enter Story Points by using the drop-down menus as Figure 9 shows. (Also see Mike Cohn's books, *User Stories Applied* or *Agile Estimating and Planning*, cited in Resources, because a discussion of estimating and Story Points is beyond the scope of this article.)

Figure 18. Set complexity for work items



Epics and Stories

By default, *Epics* and *Stories* are the work items for a Product Backlog. In case your team also wants to see tasks (or some other type of work item, such as Defect) in the Product Backlog, you must add that work item type as a Top-Level Work Item Type in the Project Area configuration:

1. Open the **Project Area** from the Team Artifacts view.
2. Within the **Process Configuration** tab, expand **Project**

Configuration, Configuration Data, and Planning.

3. Select **Top-Level Work Item Types** and add **Tasks** (or another work item type) by checking the corresponding check boxes.

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

10. Double-click a story to open its editor (in this example, shown in Figure 19, the story says "As a player, I can play against a weak engine that recognizes rings.")
11. In the Description field, provide additional details about what is expected of the story (a description can also be added when initially adding the Story to the plan). In this case, it references (imaginary) documentation that is attached to the Project Area.
12. The creator of an item is automatically subscribed to it. In case you do not need that, you can click the **Subscribers** link in the Quick Information section at the bottom-left side and select **Unsubscribe Me**.

Figure 19. Details for an item

The screenshot shows the 'Story 9' editor in the IBM Rational Product Backlog tool. The interface includes a top bar with the project name 'Product Backlog [Release 1.0]' and the story title. Below the title is a 'Summary' field containing the story text. The main area is divided into two sections: 'Details' on the left and 'Description' on the right. The 'Details' section contains various fields for configuring the story, such as 'Type' (set to Story), 'Filed Against' (set to Backlog), 'Story Points' (set to 8 pts), 'Progress' (0/0 h), 'Team Area' (Havannah Team), 'Creation Date' (May 31, 2009 6:48 PM), 'Created By' (Sasha), 'Owned By' (Unassigned), 'Priority' (2), and 'Planned For' (Release 1.0). The 'Description' section contains a text area with the description 'See Player's guide attached to the Product Backlog overview for details regarding how rings can be formed (and blocked)'. At the bottom, there is a 'Quick Information' section with links for 'Subscribers', 'Add Subscriber...', and 'History'. The 'Subscribers' link is highlighted, showing a list of subscribers with an 'Unsubscribe Me' button.

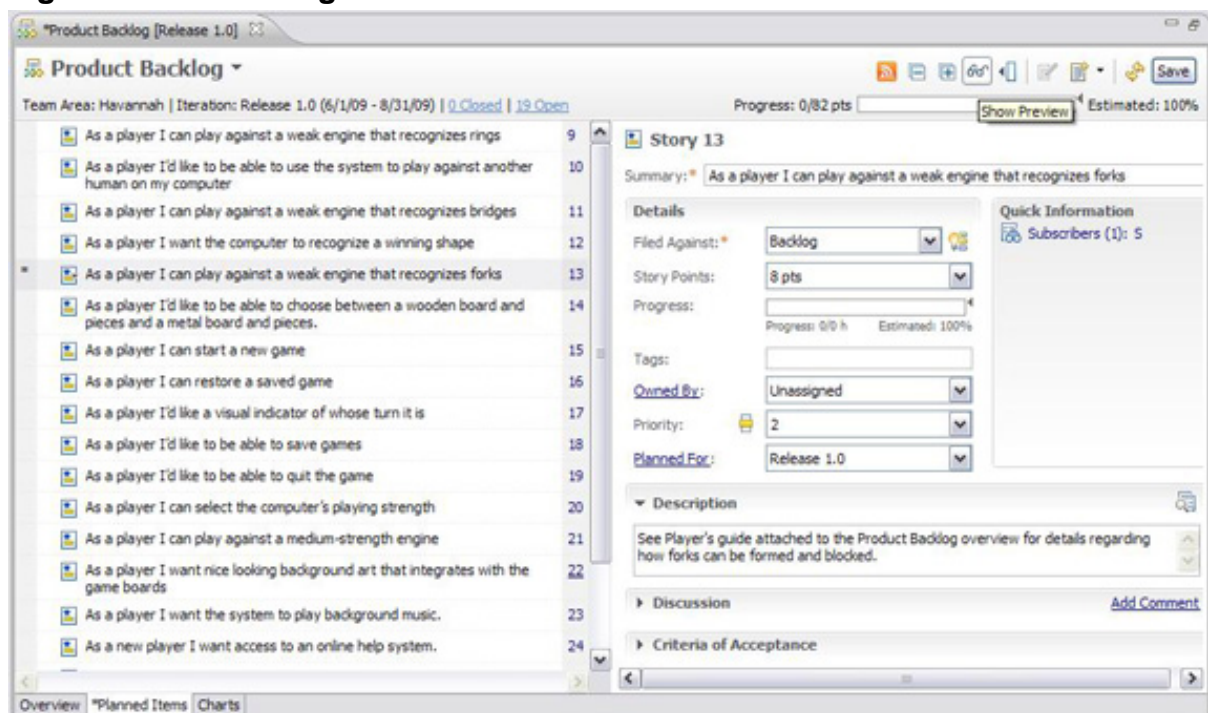
Larger view of Figure 19

- Click the **Acceptance Test** tab at the bottom. Add the acceptance tests identified by the team and product owner as proof that the story was successfully completed and will meet stakeholder expectations.

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 (see Figure 20), and the editor will split to show the iteration plan and the currently selected item side-by-side.

Figure 20. Previewing items



Larger view of Figure 20

Tip:

If you want to organize and view Epics and Stories in a hierarchical order in your backlog, you must use a view mode that shows your work items in a tree view. The Work Breakdown mode shows items in a tree view, but you can configure any other mode accordingly, as well.

What's next

Subsequent parts of this series will cover planning and managing the sprint (see

"More in this series") and agile software development estimating and planning details.

Acknowledgements

We'd like to thank Dirk Baeumer, Jazz Agile Planning Team Lead, for his technical review and David L. Schmidt, Technical Lead of Tivoli Software Engineering, for his general review, comments, and encouragement.

Downloads

Description	Name	Size	Download method
Rational Team Concert Scrum sample data	ScrumSampleData.pdf	54KB	HTTP

[Information about download methods](#)

Resources

Learn

- Find out about IBM Rational Team Concert features and benefits:
 - [Rational Team Concert Information Center](#)
 - [IBM developerWorks page for Rational Team Concert](#), with links to many other resources
 - [Jazz.net forums](#)
 - Webcast: [Using Rational Team Concert in a globally distributed team](#)
 - Demo: [Dashboards and reports](#)
 - Podcast: [IBM Rational Team Concert and Jazz](#)
 - Trial downloads (free):
 - [Enterprise Edition](#)
 - [Express Edition](#)
 - [Standard Edition](#)
 - [Rational Team Concert Quick Reference](#)
- Learn more about scrum project management:
 - [Scrum Alliance](#)
 - [How to Implement Scrum in 10 Easy Steps](#)
- Learn more from Mike Cohn's books, both [User Stories Applied: For Agile Software Development](#) (Addison-Wesley Professional, 2004) and [Agile Estimating and Planning](#) (Prentice Hall, 2005) and browse his [Web site](#).
- Read about Stakeholder Goals documents in Carl Kessler and John Sweitzer's Jolt Award-nominated book: [Outside-in Development](#) (IBM Press, 2001).
- Learn about other applications in the [IBM Rational Software Delivery Platform](#), including collaboration tools for parallel development and geographically dispersed teams, plus specialized software for architecture management, asset management, change and release management, integrated requirements management, process and portfolio management, and quality management. You can find product manuals, installation guides, and other documentation in the [IBM Rational Online Documentation Center](#).
- 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 [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

- Get Jazz and Rational Team Concert downloads and updates, and participate in the forum discussions at [Jazz.net](#). (Requires registration, which is free.) This is also where you can enter and review enhancement requests and bug reports.
- Practice continuous improvement by using the IBM Rational [Self-Check for Software Teams](#) tool to support your Reflections meetings.
- 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

- Check out [developerWorks blogs](#) and get involved in the [developerWorks community](#).

About the authors

Millard Ellingsworth

Millard Ellingsworth develops software and encourages adoption of agile software development as a member of IBM's Enterprise Content Management group. He serves his team as scrum master and others as an agile coach and workshop leader.

Thomas Starz

Thomas Starz is a software developer and agile development coach with IBM Software Group and works as part of Tivoli Service Automation Manager team. Thomas was among the early adopters of agile methods in his organization. He

gathered his first agile experiences through practicing with student teams before he introduced agile practices in product development. He is a Certified Scrum Master and Practitioner and an enthusiastic user of Rational Team Concert. While working as a mentor and coach, he has also helped several teams get started with Rational Team Concert.

Trademarks

IBM, Rational, and the IBM logo are trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.