Scrum project management with IBM Rational Team Concert Version 2: Part 3. Using the web interface

The web interface provides full access to all features without the need to install client software

Thomas Starz
Millard Ellingsworth
July 13, 2010

This tutorial provides up-to-date instructions for using the web interface provided by IBM® Rational Team Concert™ Version 2.0.0.2 and the new Scrum Process template for scrum project planning and management.

View more content in this series

Overview

Before you start

Before you start with using IBM® Rational® Team Concert™ for scrum planning, make sure that you have a thorough understanding of the scrum basics and that you understand terms such as Product Backlog, Story Points, and so forth. There is a short introduction to scrum project management at the beginning of an earlier article.

About this series

The breadth of IBM® Rational Team Concert™ collaboration capabilities can make it challenging to quickly understand it and make the most of it. This series of tutorials helps newcomers to Rational Team Concert to quickly become productive in using it to enhance their team's capabilities.

About this tutorial

For quite some time, scrum teams have been using Rational Team Concert for planning and tracking their work. It provides several UI options that they can use to their discretion. For example, developers might prefer rich clients, such as Eclipse or Microsoft® Visual Studio. Many other stakeholders do not want to install additional software on their systems, so they prefer web
interfaces. With Version 2, the IBM® Rational® Jazz™ and Rational Team Concert teams have delivered dramatic improvements to their web interfaces so that it now provides all of features that you need for your daily work. The Eclipse interface is described in an earlier article in this series.

**Objectives**

This tutorial provides up-to-date instructions on how to create and manage a scrum project by using only the Rational Team Concert web interface, rather than the Eclipse version.

**Prerequisites**

To fully follow all steps of this tutorial, you need a Jazz server, Version 1.0.0.2, with Rational Team Concert capabilities installed. For some steps, you also need administrative rights for this server. For details, see jazz.net.

**System requirements**

This tutorial uses Rational Team Concert V2.0.0.2 Standard Edition installed on a local system by using the standard configuration with Apache Tomcat application server and an Apache Derby database. You must have a server license and at least seven developer licenses installed, and you also need full administrative rights to perform all of the steps described. Your Jazz server must be on suffix level 2 (for example, 1.0.0.2), too, and you must use the scrum template provided with Rational Team Concert 2.0.0.2. If you use an earlier version of Rational Team Concert, some things will look different or might not be available.

To follow this tutorial, you may use the Express-C download of Rational Team Concert which includes everything that you need, including 10 free developer licenses and easy-to-follow instructions for completing the server and client installations. The wiki on jazz.net shows a compatibility matrix that illustrates which client software can be used with a certain IBM® Rational® Jazz™ Team Server version.

**Introduction**

Earlier articles in this series used the Eclipse-based rich client interface to work with Rational Team Concert, but this tutorial focuses on the web interface, instead. The web interface is particularly useful for all stakeholders of a project that do not want to install a rich client for the tool or for those whose job responsibilities might take them away from their primary workstation to places where they have only browsers and network connections.

With Rational Team Concert Version 2.0, the web interface has been enhanced, so that it now offers almost the full functionality of Rational Team Concert for agile planning. Formerly, many functions were available only by using the rich client interface. With Version 2.0, only work item categories and most of the project configuration must be defined by using the rich client interface.

**Note:**

Throughout this article, we distinguish between the web interface (accessed by using a browser) and the rich client interface (Eclipse or Microsoft Visual Studio).
This tutorial shows how to set up the tool and how to create an initial project by using the Rational Team Concert 2.0.0.2 web interface.

This tutorial is based on the case study that Mike Cohn discussed in his book titled *Agile Estimating and Planning* (see Related topics). It describes a small team that is set up to develop a new computer game called "Havannah." This tutorial shows how to set up Rational Team Concert to reflect this team and project and how the team creates plans and adds work items to these plans.

**Set up the Rational Team Concert repository and users**

Before you can perform the steps described in this tutorial, you need a Jazz team server (with Rational Team Concert capabilities) installed and available for your use (see System Requirements). This tutorial assumes that you've installed your own Jazz server and are starting from scratch. If this is not the case, the initial behavior that you experience might differ slightly from what is described here.

**Log in to your Jazz server**

1. Be sure that your Jazz server is started.
2. Direct your browser to the URI of your Jazz server.
   - When you use an installation of a Jazz server on your local system, the URI might be, for example, https://localhost:9443/jazz/.
3. Enter the user ID and the corresponding password.
   - The initial user ID for a newly installed Jazz server is **ADMIN**, with **ADMIN** as the password, too (both are case-sensitive) unless you deactivated **ADMIN** and activated a different admin user ID during your setup process.

**Note:**

Notice the lowercase letter "s" in the "https" of the URI, which shows that Jazz uses a secure connection. Depending on your browser, you could get odd behavior if you forget to include the lowercase s. Also, notice that the security certificate is self-signed and must be explicitly accepted if you are using the Firefox browser.

The screen shown in Figure 1 might look slightly different, depending on the browser that you use (see jazz.net for a list of supported browsers).
When you first log in to your Jazz server, you land in the server administration area (see Figure 2). There, you might need to perform some license activation activity, add a server license and client activation licenses, and do other administrative tasks. These tasks are not part of this tutorial (although they are clearly described in the software documentation).

**Figure 2. First screen**
Add or import users

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, follow LDAP Configuration for Newbies on Jazz.net.

The next action is to create the users in Rational Team Concert. User data is stored in the Jazz repository; therefore, 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 directory), you can import users without having to create them, as the sidebar explains. Here, though, 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, select the User Management option at the top of the window.

Figure 3. Creating a new user

2. To add the first user, click the Create User button.

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 email address is a required field, because Jazz can send email notification of many of the team's activities. For Sasha, enter sasha@bombshelterstudios.com.
   d. The user's initial password is automatically set as equal to the user ID, as noted at the top of the screen (see Figure 4).

4. Select the appropriate type of license for this user.
   a. 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.
   b. Build and IBM® Rational®ClearCase® Connector licenses are typically assigned only to administrative users.
   c. Contributor licenses are a good choice for all users who need mostly read access to the repository. With a contributor license, a user can also create work items.

5. Click Save.
Figure 4. Specifying user details

Tip:
To simplify the work with this tutorial, Sasha has been assigned to the Jazz Admins group also. This will allow him to modify settings for all users. Typically, however, users do not get Jazz Admin rights.

Note:
Users will need to enter their scheduled absences and configure their work environments (see the tabs above the editor window) 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.

6. Create the other users that will make up the Havannah team, namely Allan, Delaney, Frank, Prasad, and Rose. To achieve this, navigate back to the list of active users, click Create User, and fill in the user information accordingly.
7. Assign them to the JazzUsers group with Developer licenses.
8. You also have the option to create user accounts for other stakeholders, such as Laura, the CFO, or Phil, the CEO, who do not contribute to the project but need to have access to Rational Team Concert to see progress and status information.

**Note:**
If these users need only read access to Jazz and do not need to update work items or plans, you do not need to assign any client license to them.

9. Finish by clicking **Log Out** at the top-right of the screen (see Figure 6).

Figure 6. Log out

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 deploy project templates or create and save plans in a project area, because this is for your scrum master or product owner to do.

10. Log in again, but as a Sasha, using his login credentials (**sasha** and **sasha**, because passwords default to the same value as the user ID).
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, select **Project Area Management** at the top of the window.
2. In the list of active project areas (see Figure 8) click **Create Project Area**.

3. Enter the project area name and, optionally, a short description.
4. When you first create a project area, you have no process templates defined. If this is now the case, click **Deploy predefined process templates** (see Figure 9). It can take time for the process templates to be deployed to your repository.

5. Select the **Scrum Process template** because the example shows how to manage the project by using scrum methods.

**Note:**
With Rational Team Concert Version 2.0.0.2, a new Scrum template has been introduced. There is also an older (deprecated) Scrum template. Make sure to select the new one.

6. Click **Save**.
Figure 10. Create a project area using the scrum template

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 11.

Figure 11. The project area for the Havannah project

On the right side of this window, you can see the team area hierarchy that is associated with this project. Because the Havannah team is rather small, the new feature introduced with Rational Team Concert 2.0.0.2 is exploited and no explicit team areas need to be created.

Tip:
Larger teams typically consist of several teams and subteams. Jazz technology can support reasonably large teams with multiple timelines and subteams. You can reflect your team structure by defining as many teams and subteams as are appropriate for your project.
Add members and specify their roles

In this example, Sasha created the project area, so he becomes its initial administrator and can now make any changes to the project area.

Figure 12. 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. As the administrator, Sasha can assign these roles to the appropriate members.

The other members of the Havannah team are now added. For small projects, you just add all members of the project to the project area.

1. Click Add at the right side of the "Members" line (see Figure 13.
2. Because only a small number of users have been added to the repository for this example, enter an asterisk (*) to show all available users.
3. Select Allan, Delaney, Frank, Prasad, Rose, and Sasha, because all of them will be working on this project.
4. Click Add & Close.
Figure 13. Add members to the project area

5. Click the Show All icon at the bottom of the member list to show the complete list.
6. Move the mouse over each user. On the right side of the line, select the icon for setting the Process Roles (two little people).

Figure 14. Add process roles
7. Select the process role for each user and add it to the list of selected roles.
   Frank is the product owner for the project. Sasha is the scrum master but also a programmer
   and, therefore, also added as a team member. All other users are team members for the
   Havannah project.
8. Click Finish.
9. Click Save to save the project area again.

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 email. If you have configured email support (when you set
up your server) and you agree to this, project members will get a welcome email message, along
with links and information for connecting to the project.

10. Because these users are fictitious, you can deselect all check marks. Otherwise, select the
    users that you want to send invitations to (see Figure 15
11. Click OK. (This dialog is only for sending email invitations, so you can cancel it safely if you
do not want them sent.)
12. If you want to send email invitations, you are presented with a further dialog where you can
    specify the text of the email (see Figure 15).

Figure 15. Send team invitation

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.

Defining timelines and iterations

For very distributed teams, it might be an option to make iterations run from Sunday through
the following Saturday to occupy the full elapsed time.

Tip:
You need to be authorized to make changes to the project area. 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 the 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 Timelines tab of the project area window (see Figure 16), you can see the default placeholder iterations that the process template established when you created the project area.

1. Click the Timelines tab of the project area editor.
2. Under Defined Timelines, expand Main Development, and then expand Release 1.0.

There is one Main Development timeline for the project. Each timeline can contain exactly one active iteration. The small blue triangle decorator on the Release 1.0 icon and the Sprint 1 icon indicate that these are the current release and iteration.

Additionally, a Backlog release has been created. It has no particular schedule and is located at the end of the timeline. This is added to be able to create a schedule-independent Product Backlog that contains items that are initially not assigned (or might never be assigned) to a specific release.

**Note:**
The Havannah project is a simple project for the first release of the product. Therefore, the timelines created by default are sufficient for the moment. If the project becomes more complex (for example, if further releases will be developed with parallel maintenance work for previous releases), the timeline definitions become more complex.

3. Select Release 1.0 and click the Edit Properties button to open the Edit window shown in Figure 16.

**Figure 16. Specify timelines**

4. Leave the Iteration Type at the default value.
5. The Identifier must be unique, but you can change it to match your needs.
6. Set the Start and End dates for the release (see Note), and then make sure that the check box for "A release is schedule for this iteration" is checked, and click OK.
7. Edit the properties for the iterations similarly to reflect the start and end dates for your sprints.
8. Click Save.
Find tasks automatically assigned to you

Some tasks are automatically created and assigned to the person who creates the project area.

1. To list these, leave the Admin interface by selecting **Project Areas** at the top left of the window.
2. Then select the **Havannah** project. This will move you to the dashboard for the project and change the items available in the menu bar at the top of the window (discussion of dashboards is not part of this tutorial).
3. Click **Work Items** in this menu bar.
4. From there, run the predefined "Open assigned to me" query. You can follow these tasks to customize your project area, closing each as you complete it.

**Note:**
Because work tracking is time-sensitive, the dates originally entered by Rational Team Concert when creating the project area 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 to four weeks. In our example, we define two-week sprints from Monday to the following Friday, leaving out the non-work days.

**Tips:**
You can make more changes to the project area later. To open it, navigate to the **Project Areas** page and select **Manage Project Areas** on the far-right side of the page (Figure 17). Or, if you have administrative rights (as Sasha has) navigate back to the Admin interface by clicking **Admin** at the right of the menu bar. This will change the menu bar again and offer a Project Area Management selection.

**Figure 17. Link to manage project areas**

As you become more familiar with Rational Team Concert, you will probably want to further adapt the project area to your team’s process and needs (as a result of discussions during your Reflections meetings, for example). The project area is configurable to a large extent; however, this is one of the few tasks that can only be performed from the Eclipse rich client interface.

**Specify team member availability**

For team workload calculations to be accurate, it is important that team members adjust their availability and workday length. Perhaps team members have responsibilities outside of the team
that limits their participation or have a vacation planned. Others, such as the product owner, do not actively contribute to the work. For distributed teams, it is beneficial when team members add their public holidays and other country-specific availabilities.

These availabilities 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. To edit the current user's settings, click the user's name at the top of the window, toward the right.

Figure 18. Edit the settings for a user

2. Select the **Work Environment** tab (Figure 19).
3. Adjust the time zone and regional settings.
4. Adjust the work days and standard working hours.
5. On the right side, move the mouse over **Main Development** and click **Edit**.
6. Adjust the user's availability.

Figure 19. Edit the work environment for a user

Typically, it is the responsibility of each team member to adjust their settings. In this example, because Sasha has Jazz Admin repository permissions, he can also adjust settings for another user.

7. Click **Admin** at the top right of the window.

Figure 20. Go to the Jazz Admin interface

8. Click **User Management**.
9. From the list of active users, select **Frank**.
10. Within the **Work Environment** tab, adjust Frank's availability to 0%, because he is the product owner and does not contribute to work products.
11. Go back to the list of active users by clicking **Active Users** on the left side of the window (see Figure 21).
12. Rose said she will take two days off. Typically, Rose would enter this herself under the Scheduled Absences. But in this tutorial, we let Sasha do this for her, so he clicks the plus sign (+) to add an absence time.

**Figure 21. User editor for Rose**

13. In the dialog, enter two days of vacation.
14. Click **OK**.

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

15. Preserve the changes by clicking **Save**.

### Create the Product Backlog for the project

One of the most important artifacts in the scrum method is the Product Backlog. Therefore, the stories that are created during project planning are now filled into the Product Backlog. Three plans have already been created automatically when the project area was created:

- **A Product Backlog**
  The Product Backlog is the complete list of all work items that have been collected for this product so far.

- **A Release Backlog**
  The Release Backlog is the prioritized list of items that are planned for a release. Typically, the items that should go into a release are selected from the Product Backlog and moved into a Release Backlog.

- **Two Sprint Backlogs**
  A Sprint Backlog is created for each of the two sprints that have been defined automatically when the project area was created. The Sprint Backlog contains the list of items that are planned for a sprint.

The plans associated to a project can be reached in various ways.

**Note:**
The menu bar adapts to the context where you are in Rational Team Concert. Although the Project Areas selection option is always available, it might be at different positions in the menu bar, depending on your previous activities.
1. One way to access the Product Backlog is to start with **Project Areas** from the menu bar. Another option is to select the project area from the **Select a Project Area** drop-down menu at the top-right side of the window.

**Figure 22. List of project areas**

2. Select **Havannah**.

**Note:**
You will see the project’s dashboard. Customizing dashboards is not part of this tutorial.

3. Select **Plans** from the menu bar to see the list of current plans (Figure 23).

These are the plans that have been automatically created when initializing the project area with the default Scrum Process template. Notice that all plans are empty, so far.

**Figure 23. List of plans for the project**

Notice the different representations of the plans. For the Product Backlog, no progress is tracked. This is because the Product Backlog is intended to be the container for all ideas and work items that might or might not get into the product over time. The work items that will actually be part of the next release will be put into the Release Backlog.
Because a Release Backlog typically contains epics and stories, the progress is tracked in points. Finally, for a Sprint Backlog that contains the tasks being worked on and where tasks are typically estimated in hours, progress is tracked in hours.

4. Move the mouse cursor over the Product Backlog and click on its name.

**Figure 24. Product Backlog**

Initially, a plan contains three pages (or tabs), one Overview page where you can enter information about your product by using wiki-style formatting, the Planned Items page, and the Charts page. Charts might not be available yet, because there are no items available in the Jazz data warehouse.

There are several ways to view a plan, called view modes. For example, for a Release Backlog plan, there are, by default, four predefined view modes available from the "View as" drop-down menu. To exclude specific items from the view, you can use the Exclude icon on the right side of the window. You can also edit existing or create your own customized view modes. To do this, you can use either the Edit or Copy icon. However, be aware that all of these changes are valid for all users of the project.

**Figure 25. View modes**

**Tip:**
In case you also want to see tasks (which are execution items) in a backlog plan, make sure to check Always load all Execution Items. For very large backlogs with a huge number of execution items, this might not be desirable due to performance problems.

To change the settings for a plan, click the **Edit** icon in the list of all plans (see Figure 23, shown previously).
Fill the Product Backlog with initial stories

Within 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 *User Stories Applied* or *Agile Estimating and Planning*, cited in **Related topics**). In the Havannah example, all items are phrased as stories, so you fill your backlog with *stories*.

1. Start adding backlog items by clicking on the + (plus sign) on the right side of the plan.

**Note:**
You can also use the "Click here to create a work item" link (see Figure 24, previously) to create a new item of the default work item type for this plan. For backlog plans, the default type is typically Story.

2. Enter the summary for the new item, as shown in Figure 27.

**Figure 27. Adding items in the backlog**

3. Click **Save** and **Close** at the upper-right of the window at any time to save your work.
4. Add all further items as requested by the product owner.

**Tip:**

Scrum project management with IBM Rational Team Concert

To add work items above or below an existing item, use the Add Work Item Above or Add Work Item Below icons.

5. After the stories are added, the next step for the team is to estimate them. Enter Story Points by using the drop-down menus as shown in Figure 28 (also see Mike Cohn's *User Stories Applied* or *Agile Estimating and Planning*, cited in Related topics, because a discussion of estimating and Story Points is beyond the scope of this article).

Figure 28. Backlog with stories

![Backlog with stories](image)

**Note:**
The changes made in the list view are immediately applied. No explicit "Save" action is necessary, unlike the rich client interface where you need to explicitly save the backlog after such changes.

6. Then, the product owner is to set priorities for them. The values of High, Medium, and Low are available for the Priority attribute. As for estimates, you can use a drop-down menu to assign priorities by using the (see Figure 29).

7. 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. 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 29. Set priority for work items

Tips:

- In earlier versions of Rational Team Concert V2, epics and stories were the only work item types for a Product Backlog. This restriction has been removed with the new Scrum Process template provided in Version 2.0.0.2.
- If you want to organize and view items 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. See Figure 25, shown previously, to see how this can be achieved.

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

8. Click a story to open its quick editor (in this example, shown in Figure 30, the story says "As a player, I can play against a weak engine that recognizes forks.")
9. In the Description field, provide additional details about what is expected of the story (of course, you could have added a description when initially adding the story to the plan). In this case, it references (imaginary) documentation that is attached to the project area.
10. In the Acceptance Test field (Figure 30), add the Criteria of Acceptance identified by the team and product owner as proof that the story was successfully completed and will meet stakeholder expectations.
Figure 30. Details for an item

11. Select **Save** or **Save and Close** after changing a story.
12. To simply close the story after no changes, click the list item again. Clicking on list items basically toggles the visibility of the item editor.

**Tip:**
To open an item in the Work Items editor, from the quick editor view, click the item number (for example, the story above is number 17). In the Work Item editor, you have access to the full content of an item. For example, the creator of an item is automatically subscribed to it. In case you do not need that, you can click on the Links tab in the Work Item editor and remove the user from the list of subscribers.
Plan the release and sprint

Although the Product Backlog is the container for all not-yet-planned stories, explicit planning activity takes place to decide which stories go into a release or an iteration. This is particularly useful for larger teams with many stakeholders; everyone may be allowed to fill stories into the Product Backlog, and the product owner can decide which of these items are added to the release.

When you create a project area by using the new Scrum Process template, plans are automatically created for the release and the first two sprints. You might recall that, earlier, we adjusted the release timeline dates right after creating the project area.

Fill the Release Backlog

During release planning, items are pulled from the Product Backlog into the release. According to the ranking provided by the product owner based on product research, as many stories can be pulled into the release as can be accomplished in the scheduled time frame (based on the team’s velocity). The release plan can easily be revised later, and items can be moved back to the Product Backlog. In return, other items can be pulled into the release based on stakeholder feedback.

1. If the Product Backlog plan is not opened yet, open it now.
2. Select all but the last two stories.
3. After selecting the 18th check box, move the mouse cursor to the right and select Modify, and then plan all selected items for Release 1.0 (see Figure 32).
4. Changes are saved immediately, and all moved items are shown as inactive (gray).
5. Refresh the plan to show only the remaining two items.

Figure 31. Work Item editor
7. If items are not already shown, refresh the plan.

**Note:**
In case you want to move items that have children (sub-items), make sure to select and move the children, as well. Otherwise, only the selected parent items are moved into the release.

The advantage of the Release Backlog is that you can maintain only those items in the plan that will actually go into the release. The release burndown chart (discussed later in this article) reflects items that are moved in and out, which allows you to precisely monitor whether the release goal can be met.

**Fill the Sprint Backlog**

Iteration planning starts with pulling as many stories from the Release Backlog into the sprint as the team can commit to. The second step is to develop the tasks for each of the stories that the team needs to complete. Although the result of sprint planning is a collection of estimated tasks, which are most likely assigned to various team members, it is important to remember that the purpose of sprint planning is for the team to commit to completing a collection of stories, thereby adding new and deliverable functionality to the product. The planning and estimating helps the team decide how much work will fit into the sprint.
1. Open the Sprint Backlog plan for **Sprint 1**.
2. Select the **Overview** tab.
3. On the Overview page, click **Edit Page** and document the sprint goal (see Figure 33).

**Figure 33. Documenting the sprint goal**

![Image of Sprint Backlog](image)

4. Click **Save**.
5. Assign stories to the sprint

From the Release Backlog, select the relevant stories, and plan them for **Sprint 1**. Proceed the same way as described under “Fill the Release Backlog” previously for release planning (see Figure 32).

6. As an alternative, in the **Iterations** view of the Release Backlog, drag the stories from the release to an iteration (Figure 34).

**Figure 34. Assigning stories to the iteration**

![Image of Release Backlog](image)

This will assign the stories to the Sprint Backlog. Changes are applied immediately. (In Rational Team Concert V1, the stories actually moved from the Product Backlog to the Sprint Backlog, but in Rational Team Concert 2.0, they remain visible in the Product Backlog for easier tracking. In V2.0, you use the Iterations view to monitor which iterations the stories are assigned to.)
Figures 35 and 36 show the results.

**Figure 35. Stories assigned to an iteration (release backlog)**

![Figure 35. Stories assigned to an iteration (release backlog)](image)

**Figure 36. Stories assigned to an iteration (Sprint Backlog)**

![Figure 36. Stories assigned to an iteration (Sprint Backlog)](image)

**Tip:**
You might need to refresh the Sprint Backlog to see the newly planned stories.

**Add tasks to stories**

The next step is to break down the stories that have been added to the sprint. This means that tasks are created for every step that needs to be performed to get the story to "done" (that special scrum definition of "done," which means that the feature is truly finished and can be fielded, if necessary).

1. Move the mouse cursor over the rectangle on the left of an item to bring up the action selection menu.
2. Move the cursor to the right until you see the "Add Work Item Below," action label (Figure 37), and click that action.
Figure 37. Add tasks to a story

3. Select **Task**.
4. Enter the summary for the task.

Figure 38. Add a task

5. Click **Save and Close**.
6. A task will be created below the story. Notice that it is created at the same indentation level as the story.

**Note:**
If you do not see the newly created task in the plan, make sure that Execution items are not excluded from the plan (see Figure 39).

Figure 39. Filter menu

7. Because this task "belongs" to the story, bring up the action menu for the task and select **Demote** to move it under the story. As an alternative, you can drag it onto the story, which also moves it "under" the story.
Figure 40. Demote the task

Tip:
Make sure that you use a view where items are displayed in a tree mode. In case you selected a view where items are shown in a flat manner, you will not be able to indent tasks. The Work Breakdown view that is opened by default when a sprint plan is created shows items in a tree view.

In a typical Sprint Planning meeting, team members 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.

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

Tip:
Because of the iterative and interactive nature of sprint planning, it might work best to do these steps for one story and then complete the steps below that to provide estimates and owners for the tasks. Then you can return to this section to plan the next story. There is no point in planning more work than there is time to do. Keep an eye on the Team Load indicators to know when to stop.

Estimate the tasks and assign owners

Now it’s time to estimate the tasks that have been entered for your stories. You can follow these steps in the order that fits your team’s practices:

1. To enter estimates for your tasks, click on the task to open the Quick editor. Then enter the estimate. Do this for each task, one after the other.
   Alternative: You can also enter estimates in the list by selecting from the drop-down menu. If your time estimate is not in the drop-down menu, choose More to get a small input screen where you can enter your estimate.

Figure 41. Entering time estimates
When you have entered all estimates, you can see the total estimates for each story, as well as for the whole sprint.

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, you might prefer to finalize a time estimate after an assignment has been made.

2. In the Sprint Backlog, from the list, drag tasks to the members assigned to the Havannah team. After assigning some tasks to team members, the Sprint plan might look as depicted in Figure 42.

**Figure 42. Assign owner**

The arrows at the left of an item indicate that these items are assigned to someone or somewhere else. Typically, tasks are assigned to team members, but the stories remain unassigned, and the team or the product owner decides when a story is finished. One advantage to this approach is that the breakdown of a story into tasks is always visible from this (and other) views that display both stories (top-level work items) and tasks (often not considered top-level work items).

Also, the workload is shown for every team member. Team workload can be monitored as tasks are estimated. To balance the workload and monitor progress, all team members must have entered their work assignments, as well as their scheduled absences.

Notice that Frank has no work hours and Rose has only 54 available work hours because of their limited availability or vacation days. 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. The intention is that all of the work committed is completed, no matter what comes up. So choose a percentage
of slack that fits your team. To help assure success, leave more slack in the early stages; you can tighten it as you better understand the team’s rhythm and capabilities.

**Monitor work and progress during a sprint**

During the sprint, team members accept the work that is assigned to them, start working on tasks, regularly report how much time they believe is remaining on the tasks, and eventually resolve them. Also, they track how much work has already been completed for a particular task.

This is done to support the correct display of the remaining work during an iteration. Given that the scrum method emphasizes work completed, not work started, it is preferable to start and complete a work item before moving on to the next item.

**Using the Developer's Taskboard view**

A great way to assign and monitor work is the **Developer's Taskboard**, which shows the tasks assigned to individuals in columns that indicate which tasks they have already started or completed. Also, it always shows the relationship of the tasks to their parent items (typically stories) in the left-most column. Scrum masters or team members can easily reassign items by dragging them to other team members. They can start working on items or set them to **completed** by simply dragging them to the appropriate column.

**Figure 43. Developer's Taskboard**

To help keep track of the amount of work still open until the story is completed, **Time Remaining** should be recorded for each task that people worked on each day. There are several ways to edit a work item. To open a quick editor window, just click on the item text (see Figure 44).
Figure 44. Quick editor window

Time Remaining is a simple text field; therefore, if the team works on a task for multiple days, each team member needs to update the Time Remaining field each day according to what is left at that time for the particular task.

**Tip:**
To help team members track their histories and successes in estimating tasks, a correction for the estimate should be entered as soon as it is discovered that the original estimate was inadequate.

To open the work item editor, just click the task ID. 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 can 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 (development completed), the scrum master can set the story as "Implemented" (see Figure 46).

Using the Planned Time view

Tracking can also be done efficiently by using the Planned Time view for the Sprint Backlog (see Figure 47). Team members can move tasks, for example, from their inboxes into their weekly plans. Also, absences are shown in this view.
Figure 47. Planned Time view

The Planned Time view makes it easy to see what work is remaining and when the owner is planning on doing it. As teams browse this view, they might find ways to reorder their tasks to better support their teammates. The Planned Time view can also be used to support the daily scrum meeting, thus making it pretty easy to discuss what's done and what's next.

Track and report progress

A very flexible way for team members to track their work is by using queries. There are many predefined queries available, plus you can also easily create your own queries. To use a predefined query, go to Work Items and click on the query.

Monitor work by using queries

To build your own custom query, you can either start from scratch or modify an existing query. If an existing query almost matches your needs, you can use steps similar to these to customize it and then save it with a new name:

1. In the Work Items view (Figure 48), click Create Query.
2. Start adding conditions by clicking on the plus sign in the middle of the window.
3. In the Add Condition window, select Type and click Add attribute condition.
4. For Type, for example, select Story.
5. Add another condition to select Planned For and enter Backlog.
6. Optionally, configure the way that the query results are displayed.
   a. On the **Result Layout** tab, add or remove columns and specify the sort column and order.

7. Give the new query a name, for example, *All stories in Product Backlog*, and click **Save**.
8. Click **Run** to run the query and see the results.

**Monitoring activity**

One way to easily prepare for the Daily Scrum meeting is to use the predefined **Recently Modified** query to identify tasks that have been updated in the last day or so (Figure 50). You can
configure the number of hours that qualifies as "recently" by using a query argument. The default value is 12 hours.

This list will quickly show who has been doing what. It helps determine who has not reported any progress lately.

**Figure 50. The "recently modified" query**

Another way to see what is going on is the Project Events viewlet that can be added to the project's dashboard. Dashboards, in general, are a good way to monitor activity and show status. Several dashboards can be configured with various viewlets to address the team's and stakeholder's needs. Figure 51 shows an example of such a viewlet that contains the events for the project. The settings for this viewlet can be edited to show the events of interest.

**Figure 51. The Project Events viewlet**

**Progress bars**

To quickly see progress and status information, progress bars are available at many places in Rational Team Concert. These progress bars already provide a good view of the current status of the progress for an iteration, the progress of a particular story, the workload for a team member, and much more. For active members of the project team, these load bars may present all of the information that they really need to track progress within a sprint.

**Burndown reports**

The team can also use burndown reports to keep track of how work is progressing and to view the history of their progress. They can use a Product (Release) Burndown report to track overall project progress, and they can use a Sprint Burndown report to see how work is progressing for the sprint.

**Release Burndown report**

Reports are accessed through the Reports menu choice (Figure 52).
1. Open the **Release Burndown** report (in scrum terms, this is called the **Product Burndown** report) by selecting **Reports** from the menu bar and then expanding **Work Items**.

2. Click the **Release Burndown** report.

The report in Figure 52 displays the Story Points remaining at the beginning of each sprint, as well as the total amount of planned work. In this example, the planned work remained constant over the first four sprints. During the fourth sprint, additional work was added to the Product Backlog.

**Figure 52. Open a release burndown report**

![Release Burndown report](image)

**Tip:**

For tracking trends and historical reporting, Rational Team Concert uses data warehouse technology to collect and somewhat compress what would be a large amount of data, even for a relatively small project. If you are trying to look at reports while working through this article and find that they have no data in them, you didn't do anything wrong. The report in Figure 52 was created after monitoring the project progress for six sprints. When you've just started, it's likely that the data warehousing "snapshot" collection hasn't run yet. This process typically runs at midnight in the Jazz server's time zone, so you need to have a server that's up all of the time for the process to run automatically. A user with admin rights can force a snapshot collection from the web UI (see Figure 53), although this is not recommended in a production environment, because the process can be resource intensive and can affect user response during normal working hours.

**Figure 53. Update All Snapshot Data**

![Update All Snapshot Data](image)

**Note:**

Scrum project management with IBM Rational Team Concert
There are numerous reports provided by Rational Team Concert. You can also use the Business Intelligence and Reporting Tools (BIRT), an Eclipse-based reporting system, to create your own reports. Figure 54 shows a few examples.

**Figure 54. Examples for reports**

![Examples for reports](image)

**Sprint Burndown report**

The Sprint Burndown chart provides an instant answer to the question "Are we on track to finish all of the work that we committed to do?" Assuming that all of your team members update their work items appropriately, the line trends closer to zero (work remaining) as work is completed. The burndown rate should be easily visible to all of the team all of the time.

To display the Sprint Burndown report:

- Open the **Sprint Backlog** (iteration plan) and go to the **Charts** tab.
- Alternatively, go to **Reports** and select **Burndown** (see Figure 55).

For the Sprint Burndown report to look similar to the one in Figure 55, you must have updated and logged work completed by various team members for several days.

**Figure 55. Sprint Burndown report**

![Sprint Burndown report](image)
Tip:
If team members do not update their data by the end of each day, the status of their work items will not be properly reflected in the report. Although updating the status of their tasks will help the trend line to catch up at the next processing point, there is no way to update the history. The task will show as done (finished) on whatever day it is closed, not whenever the work was actually finished.

Jazz does not skip non-work days in its plotting, because it's hard to preset which days are non-work days for a globally distributed team. Therefore, flat lines often represent weekend times but could also be symptomatic of a lack of progress (or progress updates).

Figure 56. Sprint Burndown report at the end of the sprint

Tip:
You might need to edit the report to display the sprint that you are interested in at the moment.

Schedule the Sprint Review and Retrospective meetings

An important part of the scrum process is the Sprint Review meeting. The first part of this is the demo to the stakeholders. Using 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 sprint's Overview page or as an attachment to that page.

The next part of the Sprint Review meeting is the 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 57) that can be used to make sure that the reflection meeting occurs and to track the team's comments and plans.
Do it over again for the next sprint

The life of a healthy scrum team is one filled with a rhythm of success. Plan a while, work a while, deliver, and repeat.

You have now finished your first sprint, so it's time to start on the next one:

1. The plan for Sprint 2 was created when creating the project area by using the new Scrum Process template.
2. Document the sprint goal on the Overview page
3. Then start adding items to this sprint, using the same approach as for Sprint 1.

If a story does not get completed in the current sprint, there are various practices to solve this problem. One option is to completely move an item to the new sprint.

4. Open the Release Backlog plans and select the Iterations view mode.
5. Move the items in question to the next sprint
6. Either drag all items to the next sprint, one item after the other (make sure that you do not filter execution items from the plan and that you also move all children of a story, if any), or as an alternative, select all items that should be moved. Bring up the action menu and modify them the same way as you did when moving them from the backlog into the sprint. Make sure to move all items (for example, the story and its children). Merely reassigning the story only will not move the children or subtasks.
Figure 58. Planning the next iteration

Select the next sprint as "current"

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

1. Select Project Areas from the option bar at the top of the window.
2. On the right side of the window, select Manage Project Areas.
3. Select the Havannah project.
4. Click the Timelines tab.
5. Expand the Main Development line.
7. Click Set the Selected Iteration as Current.
8. Click Save.

Figure 59. Setting the current iteration
Add more iterations

When you need to add a new iteration for the project, you do that in the same window where you select a current iteration (see Figure 59 above).

1. Select the release where you want to create a new iteration. For this example, select Release 1.0.
2. Click Create Iteration.
3. Fill in an Identifier and a Display Name for the new iteration.
4. Adjust the dates and click Finish. Notice that the time is set to 12:00 a.m. by default. To make sure that the last day of the iteration is correctly included, either set the date to the day following the end of the iteration or modify the time to, for example, 11:59 p.m.
5. Save the project area changes.

Figure 60. Creating a new iteration

Remember to also create a new plan for the new iteration.

6. Go back to the Havannah project area and select Plans
7. Click Create Plan (see Figure 61).
8. Enter Sprint Backlog for the name, Havannah for the owner and associate it with the Sprint 3 iteration.

Note:
When a new plan is created, it is by default associated with the current iteration. So if you just create the plan after setting the new iteration to Current, you do not need to modify the values.

9. Under Advanced Options, make sure to select Always load all Execution Items. The Plan Type is set to "Assign Automatically," which results in correctly setting the plan type to Sprint Backlog.
10. Save the new plan
Learn more, and consider these options

Even though you've covered a lot of ground in this exercise, there's so much more to Rational Team Concert and Jazz than scrum-based agile development planning, and there's even more to agile planning than has been described here. For starters:

- Consider using Jazz source control and the Jazz build engine to power your continuous integration efforts.
- Consider using dashboards for fast information and status exchange, and your managers will become addicted to the web dashboards. Try the various reports to see which ones fit how your team manages their process.

Acknowledgements

We thank Daniel Haischt for the time he spent to carefully review our draft and his many valuable comments that made this article as complete as it is now.

We also thank Juergen Baumann who performed a complete test run through all steps and provided many corrections, as well as suggestions and challenging questions, that helped us to become as clear as possible on the necessary details.
Related topics

• Check jazz.net for these helpful articles:
  • An introduction to planning in the web UI with Rational Team Concert 2.0.
  • A good introduction on how to configure Rational Team Concert with LDAP, under LDAP for Newbies
• Find out about IBM Rational Team Concert features and benefits:
  • Rational Team Concert Knowledge Center
  • Rational Team Concert product page
  • Webcast: Using Rational Team Concert in a globally distributed team
• Learn about other applications with IBM Skills and Training
• Subscribe to the IBM developerWorks newsletter, a weekly update on the best of developerWorks tutorials, articles, downloads, community activities, webcasts and events.
• Download Rational Team Concert

© Copyright IBM Corporation 2010
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)