Using scrum methods with Rational Team Concert Version 4: Part 1. Set up the project, team area, and Product Backlog

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

05 March 2013

Starting with Version 1.0, IBM® Rational Team Concert™ has supported the scrum project management approach. Over the years since, both this collaborative software and its support for scrum and agile teams have improved dramatically. This article, updated for Version 4.0.1, replaces previous articles and explains how to use RTC effectively within a scrum team. Part 1 walks you through setting up your project and team and filling your first product backlog.

View more content in this series

The term *scrum* originated in the game of rugby as short for *scrummage*, which is derived from *scrimmage*. In the context of software development, it 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.

New webcast from Millard

A few months after publishing this piece, Millard did a 2-part webcast titled "Pragmatic advice for using Scrum with Rational Team Concert."

It combines advice on how to do scrum well with how to use Rational Team Concert to help drive the right conversations and practices.

You can find it on the Rational Education YouTube channel.

- Part 1: Project organization and configuration advice
- Part 2: Tracking and encouraging sprint progress and Millard's opinions on how to use Velocity

Of the various agile methodologies discussed and promoted across the last decade, scrum has won, at least for now. It is the approach most commonly cited by teams claiming to be agile. (Search for "why scrum won" if you need more evidence.) One often-cited reason is scrum's simplicity. There are only a few roles that you need to fill, a few artifacts to create and maintain, and a few practices to follow. Its simplicity belies, or perhaps is, its power. What the scrum approach does best is put a magnifying glass on what is limiting your success. Teams that thrive with scrum methods use that feedback to fearlessly and continuously improve

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 typically structured in two- to four-week iterations, called **sprints**, with working software produced by the end of each sprint.

To learn from this two-part article 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, in the Downloads section.

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 (see "Get technologies" in the Resources section for a link). If you use the express installer and generally accept the default options, you should be up and running quickly. If you are installing on a Microsoft Windows system, you might want to install outside of the Program Files folder (for example, install instead to c:\ibm\...) to avoid issues with administrative permissions. You will need to reach the point where your browser can connect to your Rational Team Concert and Jazz™ repository. The software is free (no charge) for up to 10 team members.

Tip:
Although you can create a project area and do some of the operations that follow by using the jazz.net sandbox, rather than installing the software on your own system, you can't add arbitrary users in a sandbox project. Therefore, you can't really follow all of the steps and walk through the example that way unless you add other people you know into your sandbox project and change the names in the example. If you are working on this with several team members, the sandbox is a great option and will keep you from needing to install Rational Team Concert locally.

After a quick introduction to the scrum process, you'll start creating 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

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.

Requirements 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 indicate their business values to the stakeholders.

A **scrum team** should be a cross-functional mix of people with the skills necessary 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 slows down the rest of the team.

The **scrum master** manages the 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).

The team holds a **Sprint Planning** meeting at the start of each sprint. At this meeting, the product owner presents the features from the Product Backlog that are 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 on 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 backlog have time estimates so that the team can determine whether the items can be completed within one sprint. If there is insufficient time, a feature can be taken off the Sprint Backlog list 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 finish during the sprint.

During the **Daily Scrum** meeting, each team member 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. If team members are not all in one physical location, you'll find it helpful to read *A Practical Guide to Distributed Scrum* , a book published by IBM Press (2010).

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 results.

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 process. 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 might 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. As mentioned previously, 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 chart is a graphical display of the amount of work remaining in the project. 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 ideal.

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, which is easy when using the software's dashboard.

The Scrum Process template included with Rational Team Concert supports all of the important scrum roles, ceremonies, and artifacts. The software 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 user **stories** and **work items**. Even though you can simply put cards on a bulletin board, using software makes this information more accessible, especially if the team members are distributed across multiple locations.

Rational Team Concert provides ways to manage all that you need. In a **project area**, where you can manage builds, sources, plans, and much more. This article focuses on the planning and management aspects for the scrum process. After performing the first steps, such as connecting to a repository and defining user IDs for the people who use the software, I'll explain how to create plans and work items and manage them during a sprint.

Set up the Rational Team Concert repository and users

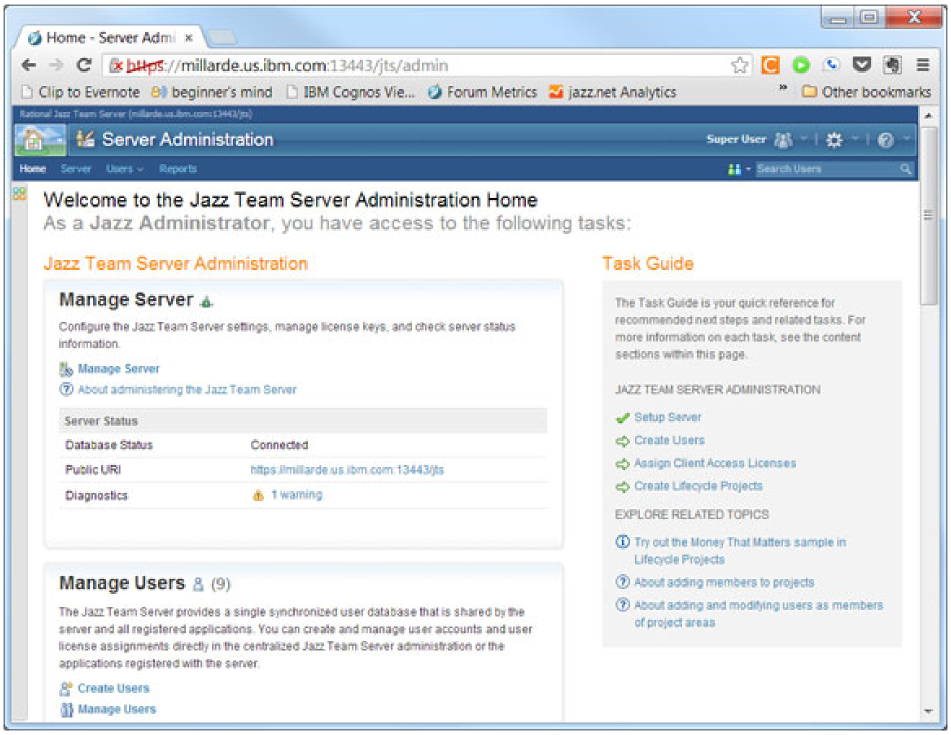
Before you're ready to look at how Rational Team Concert and the Scrum Process template help scrum teams, you need to set up the basic project. This includes adding the users for the example, creating the project, specifying a team and team members, and setting up the project iterations. All examples here are use Rational Team Concert 4.0.1 and the Chrome web browser. Using a different browser) should create very similar results. See "System Requirements for the Rational solution for Collaborative Lifecycle Management" on Jazz.net for a list of supported browsers.

Connect to your server as the administrator

1. When you installed the software, you were asked to provide a URI for the server. This defaults to https://localhost:9443/. If you changed this value, use the value that you entered. Navigate to your ***URI*/jts/admin** (for example, https://localhost:9443/jts/admin).
2. You were also prompted to create a new administrative user (to replace the default ADMIN) during setup. Log in with the credentials that you provided.

Note:
I used a non-default URI root for my server installation (https://millarde.us.ibm.com:13443/), and my admin user is named Super User. These are the values you will see in the screen captures that follow.

Figure 1. Connecting to your server as the administrator



Welcome to the Jazz Team Server Administration Home

As a Jazz Administrator, you have access to the following tasks:

Jazz Team Server Administration

Manage Server

Configure the Jazz Team Server settings, manage license keys, and check server status information.

Manage Server

About administering the Jazz Team Server

Server Status	
Database Status	Connected
Public URI	https://millarde.us.ibm.com:13443/jts
Diagnostics	1 warning

Manage Users (9)

The Jazz Team Server provides a single synchronized user database that is shared by the server and all registered applications. You can create and manage user accounts and user license assignments directly in the centralized Jazz Team Server administration or the applications registered with the server.

Create Users

Manage Users

Task Guide

The Task Guide is your quick reference for recommended next steps and related tasks. For more information on each task, see the content sections within this page.

JAZZ TEAM SERVER ADMINISTRATION

Setup Server

Create Users

Assign Client Access Licenses

Create Lifecycle Projects

EXPLORE RELATED TOPICS

Try out the Money That Matters sample in Lifecycle Projects

About adding members to projects

About adding and modifying users as members of project areas

This is the Jazz Team Server (/jts) admin page, which is different from the Change and Configuration Management (/ccm) admin page that you will use later for project area operations. If you don't see a page very much like this (for example, you see one that says you don't have access to this page or features), you did not log in with your administrator account.

Note:
The first time that you connect to a recently started server often takes a bit longer, because various services are started on demand.

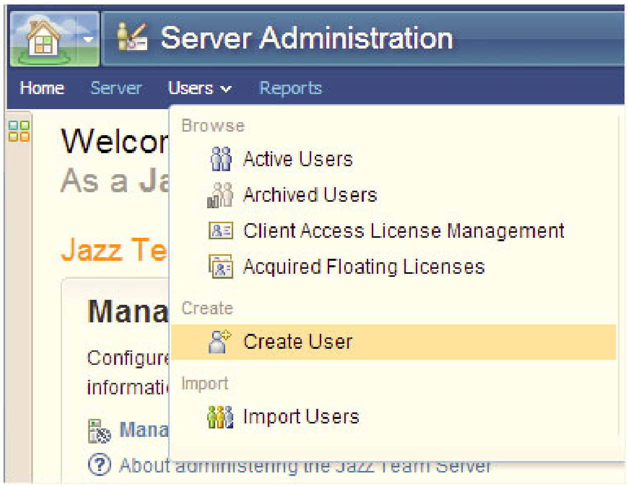
Add or import users

The next step is to create the users for the sample. Users are managed through 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.

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, click the **Users** menu and then select **Create User**.

Figure 2. Creating a new user



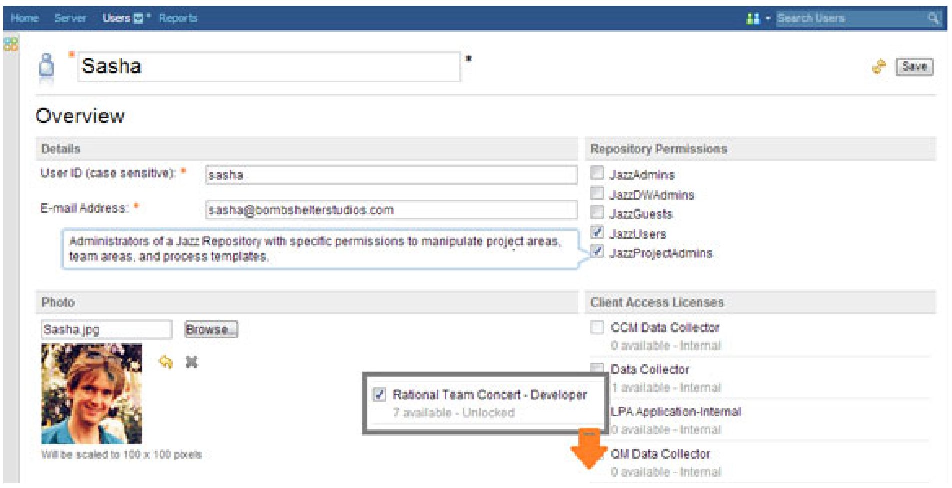
2. Enter the user information for the first user.
   a. Because you are following Mike Cohn's example, Sasha is the first user.
   b. Add a photo if one is available. That's optional, but helpful.
   c. The email address is a required field, because Jazz can send email notification of many of the team's activities. For Sasha, enter `sasha@bombshelterstudios.com`.
   d. The user's initial password is automatically set to equal to the user ID.
3. Choose the repository permissions appropriate for the user:
   JazzUsers
   They use the default permissions in a project area. This typically allows them to create and add to work items, to view plans and reports, and to use other features defined by the project's security settings.

   JazzAdmins
   They can access and update many settings related to the Jazz Team Server, related products (such as RTC), and projects.

   JazzProjectAdmins
   They have administrator capabilities within project areas and only a few Jazz server capabilities (like adding new users).
   Tip:
   The user who will administer the project must be in the repository group called JazzProjectAdmins (in the example, Sasha must be a JazzProjectAdmin; because he will not be able to perform all necessary configuration tasks otherwise).
4. Select the appropriate type of license for each user. All users in the example should get a **Rational Team Concert – Developer** license.

Figure 3 shows the resulting screen. (The license selection is further down the screen and would not normally be visible, but the inset here shows it for illustration purposes.)

Figure 3. Specifying user details



Sasha

Overview

Details

User ID (case sensitive): * sasha

E-mail Address: * sasha@bombshelterstudios.com

Administrators of a Jazz Repository with specific permissions to manipulate project areas, team areas, and process templates.

Repository Permissions

- [ ] JazzAdmins
- [ ] JazzDWAdmins
- [ ] JazzGuests
- [✓] JazzUsers
- [✓] JazzProjectAdmins

Photo

Sasha.jpg Browse...

Will be scaled to 100 x 100 pixels

Client Access Licenses

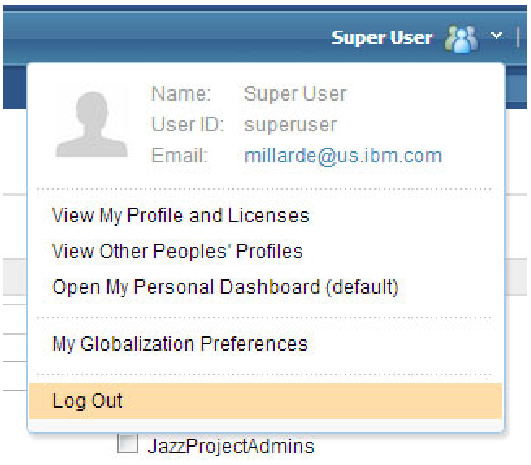
- [ ] CCM Data Collector
  0 available - Internal
- [ ] Data Collector
  1 available - Internal
- [✓] Rational Team Concert - Developer
  7 available - Unlocked
- LPA Application-Internal
  0 available - Internal
- QM Data Collector
  0 available - Internal

5. Click **Save**.

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 additional users named `Allan`, `Delaney`, `Frank`, `Prasad`, and `Rose`.
7. Assign them **JazzUsers** repository permissions with a Rational Team Concert Developer license.
8. Log out of the user's profile menu (this is important, because you need to do the next steps as Sasha).

Figure 4. Logging out of the web UI


Super User
View My Profile and Licenses
View Other Peoples' Profiles
Open My Personal Dashboard (default)
My Globalization Preferences
Log Out
JazzProjectAdmins

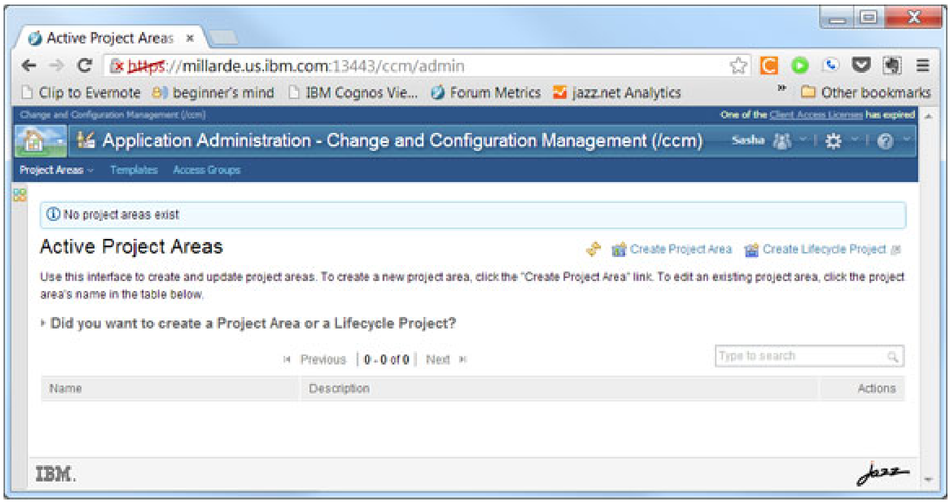
Name: Super User
User ID: superuser
Email: millarde@us.ibm.com

You've now finished what needs to be done as the administrative user.

The Jazz security model gives the administrative user basic admin privileges in Jazz but no particular rights within a given project area. For example, this administrative user cannot create and save new iteration plans in the project area, because this is for your scrum master or product owner to do.

9. Log back into the web app using Sasha's credentials (`sasha` and `sasha`, because passwords default to the same value as the user ID). You will be taken to the last page that the administrator was using, which is probably the page where Rose's user was created. Unfortunately for Sasha, this is in the Jazz Team Server admin interface (/jts/admin) to which he does not have access. You should see a message like this one:
   The user **sasha** is not authorized to access the Jazz Team Server Admin UI.
10. You need to navigate to the Change and Configuration Management administration (/ccm/admin) page in order to set up a project area. Using the browser's address bar, navigate to ***your URI root*****/ccm/admin** (for example, https://localhost:9443/ccm/admin). Your screen should look similar Figure 5.

Figure 5. CCM admin page


Active Project Areas
~~https~~://millarde.us.ibm.com:13443/ccm/admin
Clip to Evernote
beginner's mind
IBM Cognos Vie...
Forum Metrics
jazz.net Analytics
Other bookmarks
Change and Configuration Management (/ccm)
One of the Client Access Licenses has expired
Application Administration - Change and Configuration Management (/ccm)
Sasha
Project Areas
Templates
Access Groups

No project areas exist

Active Project Areas

Use this interface to create and update project areas. To create a new project area, click the "Create Project Area" link. To edit an existing project area, click the project area's name in the table below.

Did you want to create a Project Area or a Lifecycle Project?

Name	Description	Actions

Set up areas for your project and people

Now that the basic Jazz Team Server setup is finished and you have added users, you need to create a scrum project area to hold the team's artifacts, a team area for organizing the team's work, add appropriate members to each, and configure the timeline for the project. Project area configuration is an important topic that is just barely introduced here. See the references in the Resources section for additional articles that talk about this in more detail and provide extended examples.

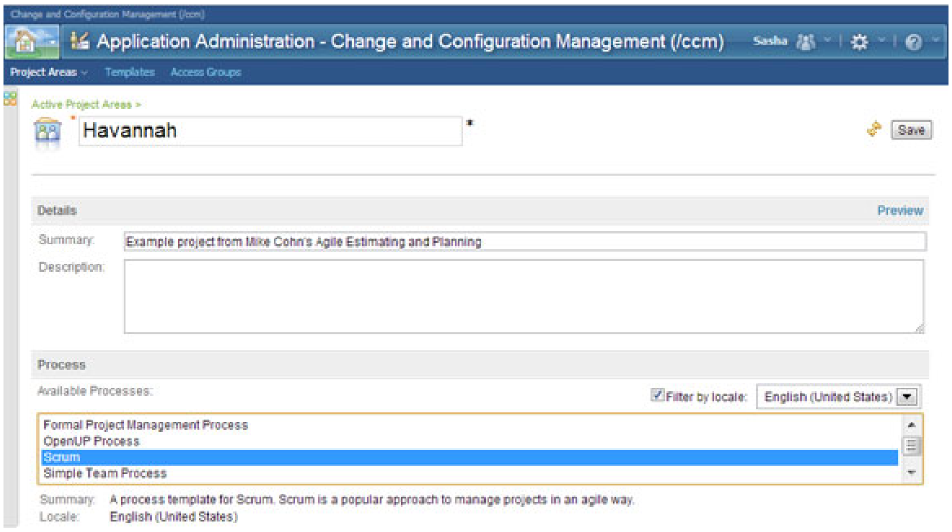
Create a project area

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. Click the **Create Project Area** link in the upper-right of the screen.
2. Enter a name for the project (`Havannah` in this example).
3. When you create the first project area in a repository, you need to deploy the process templates (they are not deployed during installation). Click **Deploy predefined process templates**. This may take a few minutes to finish.

4. After the deployment finishes, from the list of Available Processes, choose the **Scrum** template, and click **Save** (see Figure 6). You'll add members in the next step.

Figure 6. Create a project area by using the Scrum template


Change and Configuration Management (/ccm)
Application Administration - Change and Configuration Management (/ccm)
Sasha
Project Areas
Templates
Access Groups
Active Project Areas >
Save
Details
Preview
Process

Havannah

Summary: Example project from Mike Cohn's Agile Estimating and Planning

Description:

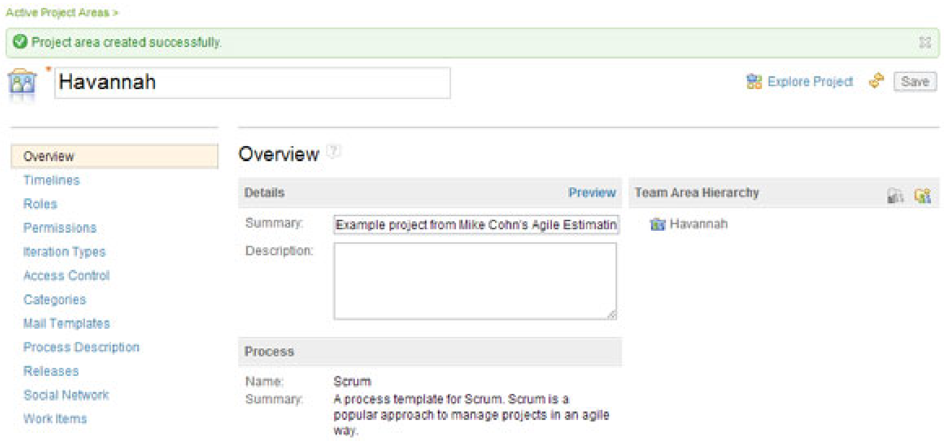
Available Processes: ✓ Filter by locale: English (United States)

Formal Project Management Process
OpenUP Process
Scrum
Simple Team Process

Summary: A process template for Scrum. Scrum is a popular approach to manage projects in an agile way.
Locale: English (United States)

It can take up to several minutes for the project area to be initialized on the server. You will see a message at the top of the screen while it is in progress. When the initialization is finished, you will see a screen similar to the one in Figure 7.

Figure 7. The project area for the Havannah project



Project area created successfully.

Havannah

Summary: Example project from Mike Cohn's Agile Estimatin

Description:

Name: Scrum
Summary: A process template for Scrum. Scrum is a popular approach to manage projects in an agile way.

Add members to the project area and specify their roles

As the creator of the project area (acting as Sasha), you are the initial project area administrator (that is different from a JazzAdmin administrator). This status gives you extra privileges within the project area. 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 area and set their roles (at least the scrum master and product owner).

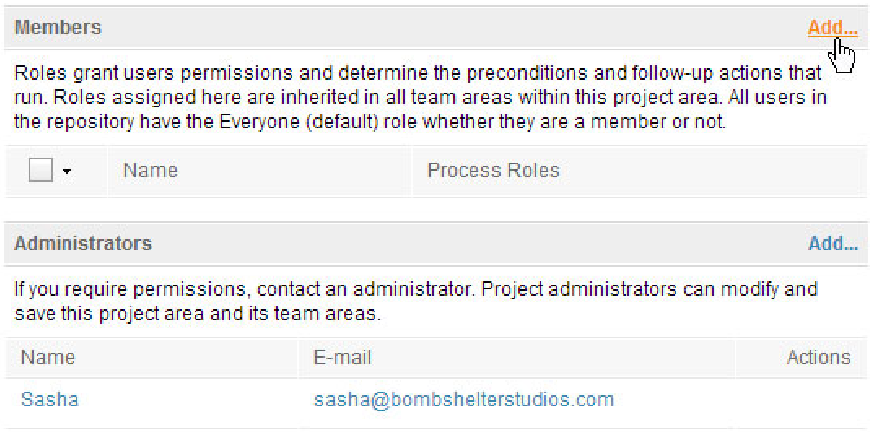
Tip:

Typically, membership at the project area level is for members with management or stakeholder responsibilities (or just people you want to have access to project status). By default, only those specified as members have access. People who get work assignments are normally in a **team area.** Everyone who is a member of an area is visible in plans and in drop-down menus for assignments, so mixing everyone into the project area is often unsatisfying and makes planning messier than it needs to be. Rather than use the bare minimum example, we'll add a few extra steps that will come in handy in real life use.

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

1. Scroll down the project area page until you see the **Members** section. Members added here will be added to the project area-level membership, because you're still working on that page.
2. Click **Add** to start the Add Team Members wizard.

Figure 8. Adding members to project area



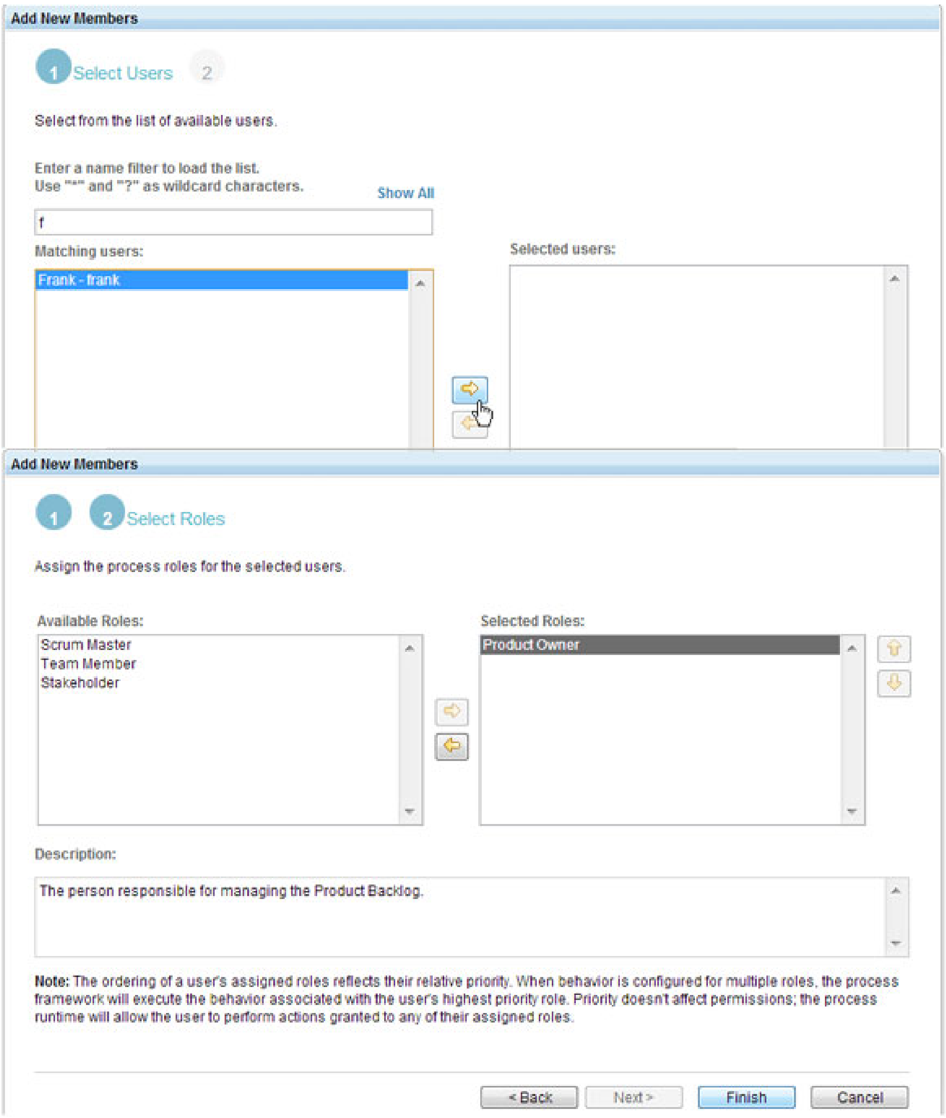
Roles grant users permissions and determine the preconditions and follow-up actions that run. Roles assigned here are inherited in all team areas within this project area. All users in the repository have the Everyone (default) role whether they are a member or not.

If you require permissions, contact an administrator. Project administrators can modify and save this project area and its team areas.

Name	E-mail	Actions
Sasha	sasha@bombshelterstudios.com	

3. To add Frank as the product owner, type an `f` (letter F) in the User Name text area to search for matching names. Because Frank was already added as a Rational Team Concert user, his name will be among the users listed.
4. Click on Frank's name in the **Matching users** list, and then click the arrow pointing to the **Selected users** list to move him to the list.
5. Then click **Next**.
6. Under Available Roles, select **Product Owner**, and then click the arrow pointing to the Selected Roles list and click **Finish** (see Figure 9).

Figure 9. Add New Members wizard


Add New Members

1 Select Users 2

Select from the list of available users.

Enter a name filter to load the list.
Use "*" and "?" as wildcard characters.

Show All

f

Matching users:

Frank - frank

Selected users:

Add New Members

1 2 Select Roles

Assign the process roles for the selected users.

Available Roles:

Scrum Master
Team Member
Stakeholder

Selected Roles:

Product Owner

Description:

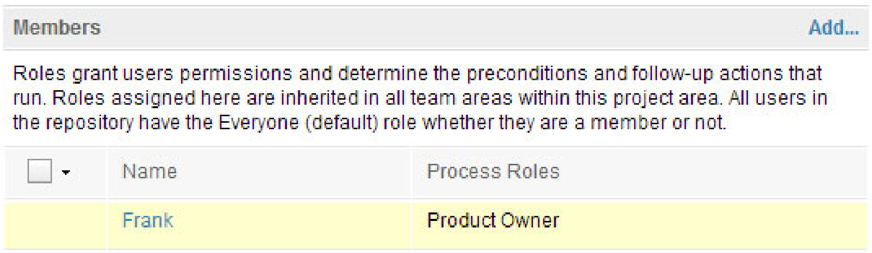
The person responsible for managing the Product Backlog.

Note: The ordering of a user's assigned roles reflects their relative priority. When behavior is configured for multiple roles, the process framework will execute the behavior associated with the user's highest priority role. Priority doesn't affect permissions; the process runtime will allow the user to perform actions granted to any of their assigned roles.

< Back | Next > | Finish | Cancel

As Figure 10 shows, Frank is now added as the product owner for the project area.

Figure 10. Members added to the Havannah project area


Members   Add...

Roles grant users permissions and determine the preconditions and follow-up actions that run. Roles assigned here are inherited in all team areas within this project area. All users in the repository have the Everyone (default) role whether they are a member or not.

	Name	Process Roles
	Frank	Product Owner

7. In the same way, add Sasha to the project area as a **Scrum Master** (this is necessary so that he has permission to change plan configurations).
8. Save the project area by using the **Save** button in the upper-right corner of the page.

After saving your changes, you will be presented with a list of the new 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.

9. Because these users are fictitious, deselect all check marks and then click **OK** or just click **Cancel**.

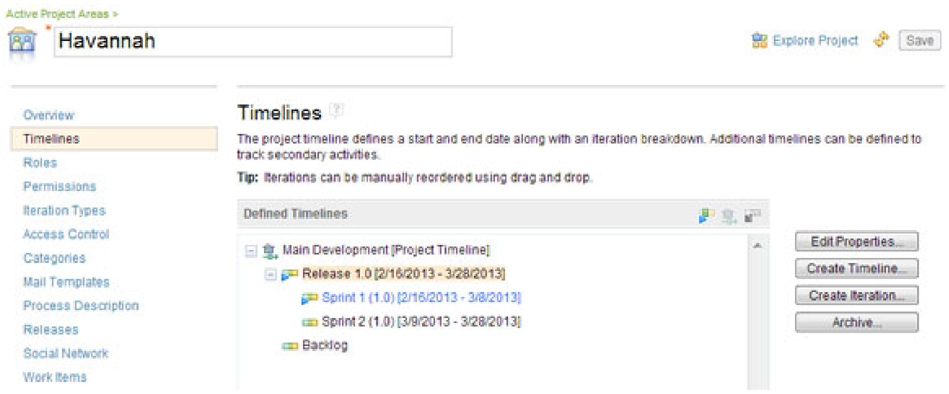
Outline the release timeline

The next step is to plan the timeline 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 an administrator for the project area or a member who has either the scrum master or product owner role. Otherwise, you will get a permission-denied error when attempting to save changes to the timeline.

1. On the left side of the project area window, click the **Timelines** link to view the timelines created for the project. You should see the default iterations that the process template established when you created the project area (Figure 11), including Release 1.0, with Sprint 1 and Sprint 2 under that, and the Backlog iteration.

Figure 11. Default timelines for the Havannah project



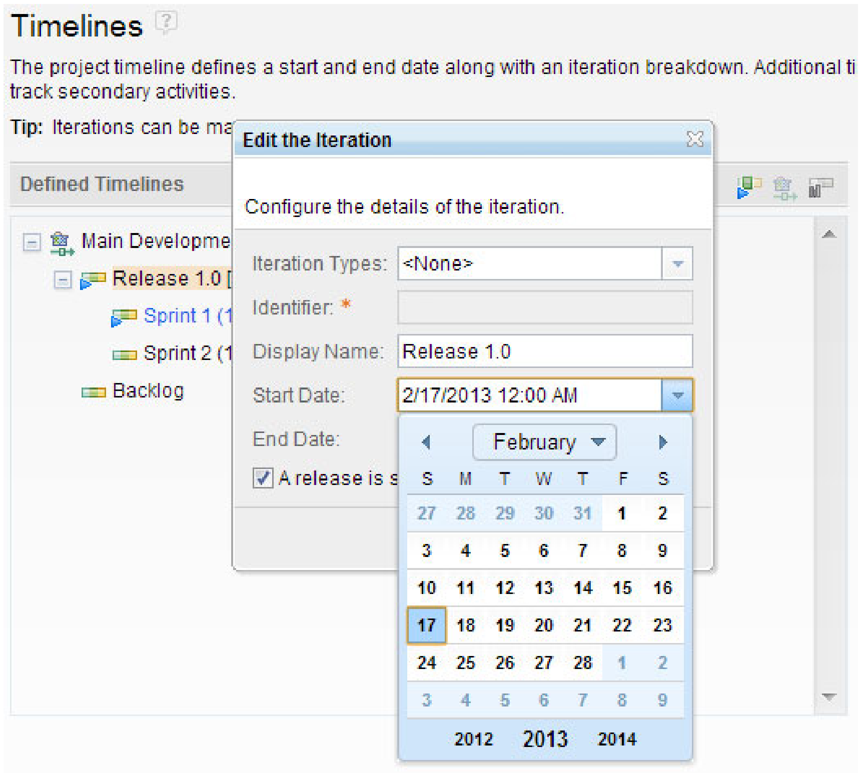
The project timeline defines a start and end date along with an iteration breakdown. Additional timelines can be defined to track secondary activities.

Tip: Iterations can be manually reordered using drag and drop.

Main Development [Project Timeline]
Release 1.0 [2/16/2013 - 3/28/2013]
Sprint 1 (1.0) [2/16/2013 - 3/8/2013]
Sprint 2 (1.0) [3/9/2013 - 3/28/2013]
Backlog

2. Under Defined Timelines, select **Release 1.0**, and click **Edit Properties** to open the Edit the Iteration dialog window shown in Figure 12 (with the Start Date selector also open).

Figure 12. Specify release and sprint timelines



The project timeline defines a start and end date along with an iteration breakdown. Additional ti
track secondary activities.

Tip: Iterations can be ma

Configure the details of the iteration.

Iteration Types: <None>

Identifier: *

Display Name: Release 1.0

Start Date: 2/17/2013 12:00 AM

End Date:

☑ A release is s

S	M	T	W	T	F	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

3. Leave the default value for **Iteration Type**.
4. The **Identifier** is specified on creation and cannot be changed here.
5. Set the Start and End dates for the release (see the Notes that follow). Make sure that the check box for **A release is schedule for this iteration** is checked, and click **OK**.
6. Edit the properties for the sprints similarly. The example will use one-week sprints.
7. **Save**.

Notes:

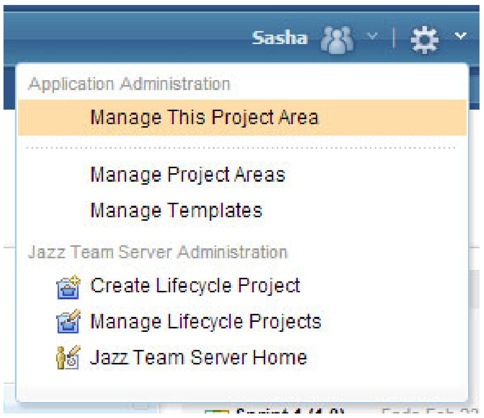
- Because work tracking is time-sensitive, the dates are based on when you are working on this example. Start the release and first iteration today or on some day in the near future. Each member's work allocation accounts for work and non-work days (more on that later), so it's fine to include weekend days so that there are no gaps between the sprints. I typically start iterations on a Sunday and end them on a Saturday.
- 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.

Tips:

- You can make more changes to the project area later. To return to this page from elsewhere in the CCM application, use the tools menu (Figure 13), next to the user profile menu on any application page.
- The project area is configurable, to a large extent. As you become more familiar with Rational Team Concert, you will probably want to adapt it to your team (as a result of discussions during your Retrospective meetings, for example). 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. It's designed that way because agile developers value people over process.

Figure 13. Tools menu for administering the project area

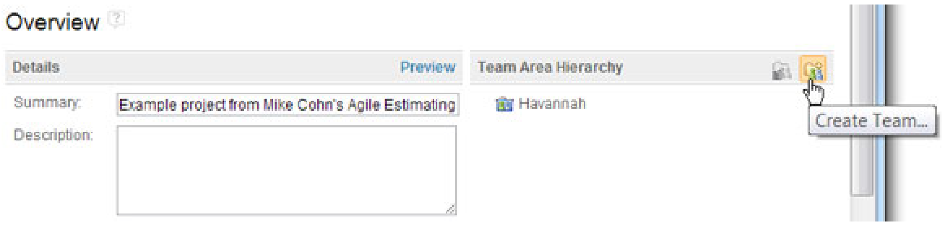


Create a team area and add members

Jazz technology can support reasonably large teams with multiple timelines and subteams. Within project areas, there are **team areas** to organize the team members (and allow for customizations specific to a team). Typically, 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 created and populated with members.

1. Click the **Overview** link on the left to return to that view of the project area. On the far right is the **Team Area Hierarchy**, which should show the Havannah project area.
2. Click the **Create Team** icon in the toolbar (see Figure 14).

Figure 14. Creating a new team area

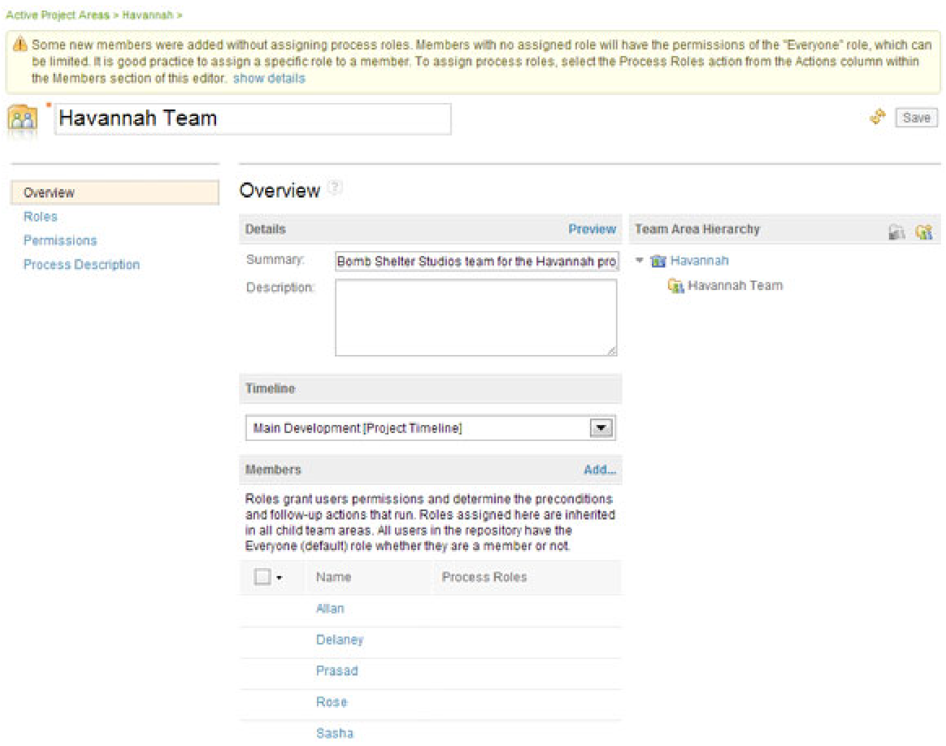


Summary: Example project from Mike Cohn's Agile Estimating

3. Type `Havannah Team` as the name of the team.
4. In the Members list, click the **Add** link (see Figure 15) to bring up the Select Users dialog window as you did for Frank, so you can add **Allan**, **Delaney**, **Prasad**, **Rose**, and **Sasha** as members. Click **Add and Close** when you are finished.
   Note:
   In the **Select Users** dialog window (similar to the one that you used earlier), you can use the `*` (asterisk) wild card character to retrieve all members (do not try that if you are using a corporate LDAP server), and select them all before choosing **Add and Close**.
5. Before you can add process roles, you need to save the team area by using the **Save** button in the upper-right. (Similar to when you added Frank to the project area, you can dismiss the email dialog.)

Your team area page for Havannah should look like Figure 15.

Figure 15. New team area



Active Project Areas > Havannah >

Some new members were added without assigning process roles. Members with no assigned role will have the permissions of the "Everyone" role, which can be limited. It is good practice to assign a specific role to a member. To assign process roles, select the Process Roles action from the Actions column within the Members section of this editor. show details

Havannah Team

Save

Overview
Roles
Permissions
Process Description

Overview

Details Preview

Summary: Bomb Shelter Studios team for the Havannah pro

Description:

Team Area Hierarchy

Havannah
Havannah Team

Timeline

Main Development [Project Timeline]

Members Add...

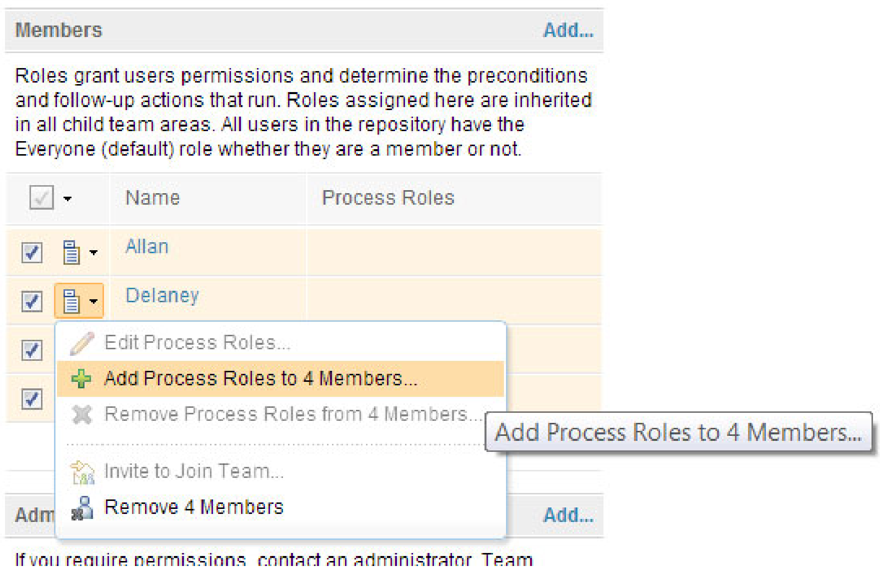
Roles grant users permissions and determine the preconditions and follow-up actions that run. Roles assigned here are inherited in all child team areas. All users in the repository have the Everyone (default) role whether they are a member or not.

	Name	Process Roles
	Allan	
	Delaney	
	Prasad	
	Rose	
	Sasha	

Now you need to add process roles for the members. You will assign the **Team Member** role to Allan, Delaney, Prasad, and the **Scrum Master** role to Sasha.

6. Hover your cursor over the column to the left of the member name. When a check box and drop-down menu icon appear, click the check box next to Allan, Delaney, Prasad, and Rose, and then use the drop-down menu for any of the selected users and select **Assign Process Roles to 4 Members**. See Figure 16.

Figure 16. Selecting multiple users to assign process roles



Members Add...

Roles grant users permissions and determine the preconditions and follow-up actions that run. Roles assigned here are inherited in all child team areas. All users in the repository have the Everyone (default) role whether they are a member or not.

	Name	Process Roles
✓	Allan	
✓	Delaney	
✓		
✓		

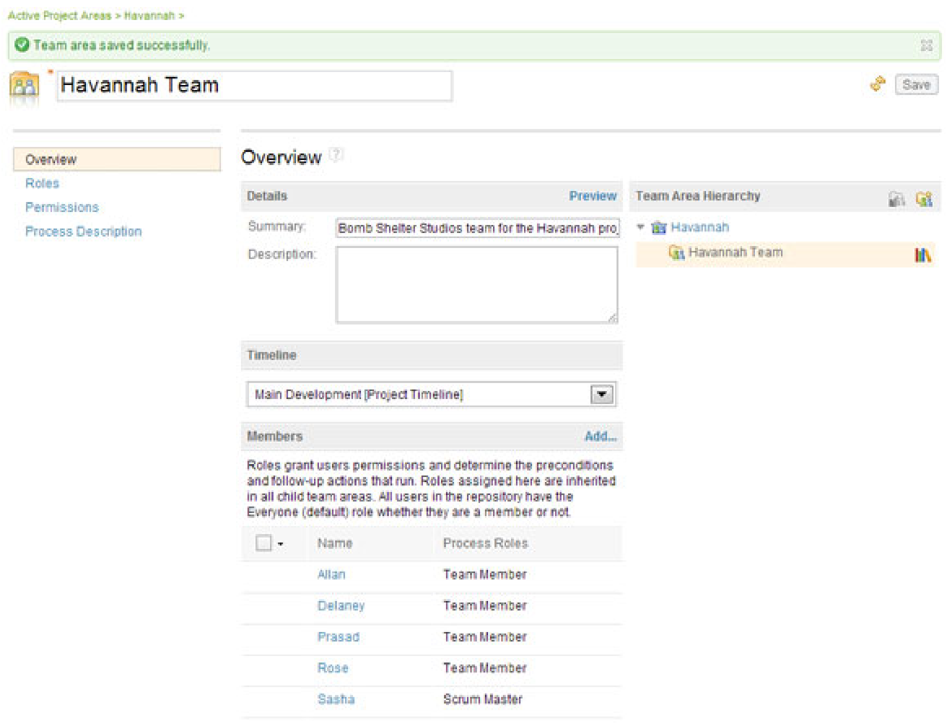
Edit Process Roles...
Add Process Roles to 4 Members...
Remove Process Roles from 4 Members...
Invite to Join Team...
Remove 4 Members
Add Process Roles to 4 Members...
Adm
Add...

If you require permissions, contact an administrator. Team

7. In the Add Roles dialog window, select **Team Member**, and click **Add**.
8. Clear the check boxes on the member list, because you are finished with these users. You can clear the boxes individually, or there is a **Select None** option in the drop-down menu in the header.
9. Repeat the steps to assign the **Scrum Master** role to Sasha. You do not need to select the check box first when using the context menu to modify just one member.
10. Click **Save**.

Your completed **Havannah Team** area should look like Figure 17.

Figure 17. Completed Havannah Team area



Active Project Areas > Havannah >

Team area saved successfully.

Havannah Team

Overview

Details

Summary: Bomb Shelter Studios team for the Havannah pro

Description:

Timeline

Main Development [Project Timeline]

Members

Roles grant users permissions and determine the preconditions and follow-up actions that run. Roles assigned here are inherited in all child team areas. All users in the repository have the Everyone (default) role whether they are a member or not.

Name	Process Roles
Allan	Team Member
Delaney	Team Member
Prasad	Team Member
Rose	Team Member
Sasha	Scrum Master

Team Area Hierarchy

Havannah

Havannah Team

Tip:
Larger teams typically consist of several subteams. You can reflect your team structure by adding as many subteams as you have working on your project.

Specify team member availability

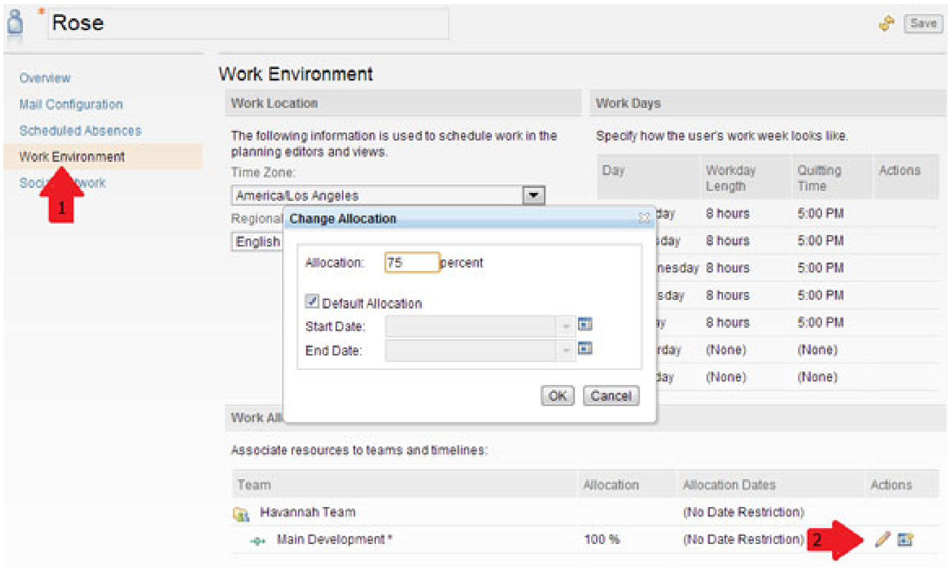
For team workload calculations to be accurate, adjust team members' availability. Perhaps a team member has responsibilities outside of the team that limits participation in this project or someone has a vacation planned. You make these adjustments on the user's page. Typically, users are expected to keep this area current themselves, but because it is an important part of calculating **Team Load** correctly, I'll illustrate it here (this is another thing that's not really necessary for this example but you will definitely need to know how to do in real life).

1. In the Members list, click **Rose** to open her user profile.

2. Rose is available to the team only 75% of the time. Therefore, click the **Work Environment** link at the left, and then click the edit icon (the pencil image) on the **Main Development** line in the **Work Allocations** list.
3. By default, if a user is assigned to only one team, 100% of his or her time (resources) is associated with that team. Change Rose's assignment to 75% (see Figure 18).
4. Click **OK**.

This will adjust the number of hours that Rose is available for work assignments. The work schedule will use this percentage to adjust Rose's workload and availability calculations.

Figure 18. Adjust the work environment for a user



Work Environment

Work Location

The following information is used to schedule work in the planning editors and views.

Time Zone: America/Los Angeles

Regional: English

Work Days

Specify how the user's work week looks like.

Day	Workday Length	Quitting Time	Actions
day	8 hours	5:00 PM	
day	8 hours	5:00 PM	
nesday	8 hours	5:00 PM	
sday	8 hours	5:00 PM	
y	8 hours	5:00 PM	
rday	(None)	(None)	
day	(None)	(None)	

Change Allocation

Allocation: 75 percent

☑ Default Allocation

Start Date:

End Date:

OK Cancel

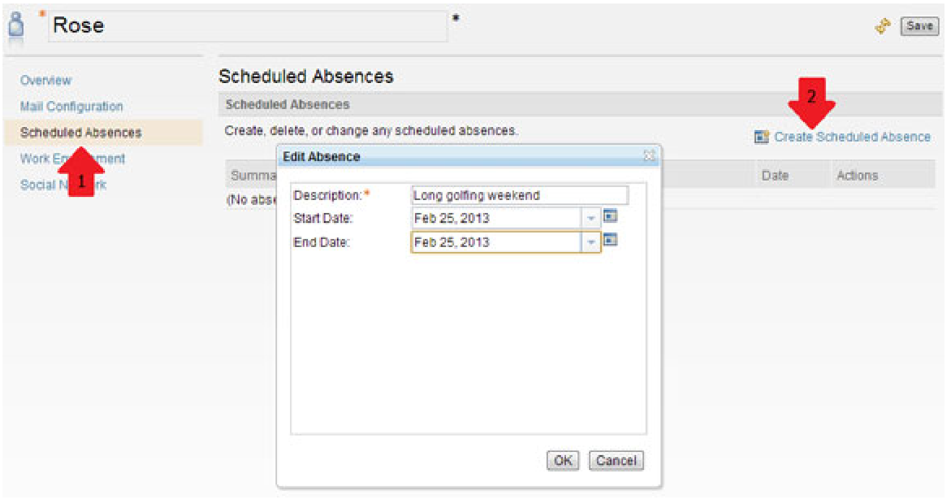
Work All

Associate resources to teams and timelines:

Team	Allocation	Allocation Dates	Actions
Havannah Team		(No Date Restriction)	
Main Development *	100 %	(No Date Restriction)	

5. Rose also has a vacation day coming up. Switch to the **Scheduled Absences** view, click **Create Scheduled Absence** on the right side of the list, and enter a description and the dates for her vacation time (see Figure 19). Any description is fine. Choose a date at least a week away (so that it is not in Sprint 1).
6. Click **OK**.
7. Click **Save** to update Rose's details.

Figure 19. Enter scheduled absences for a user



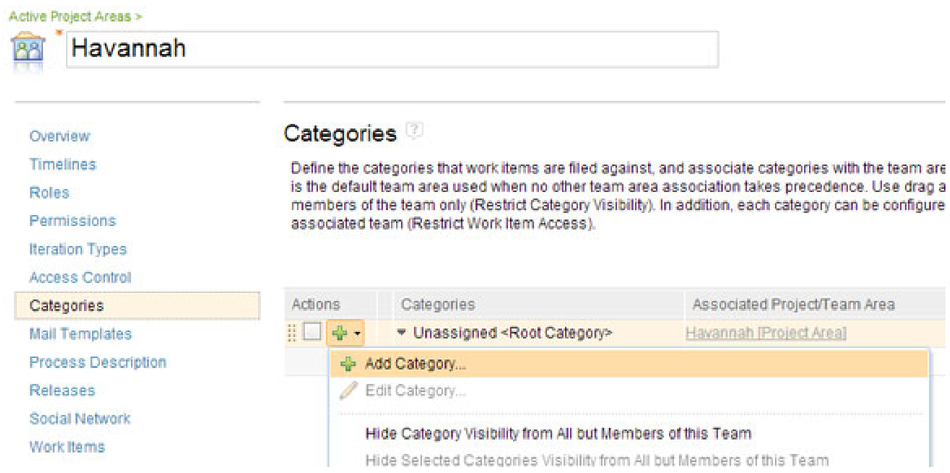
Description: Long golfing weekend
Start Date: Feb 25, 2013
End Date: Feb 25, 2013

Create and associate a work item category with the team area

A detailed discussion of the relationship between work item categories and team areas is not necessary for this example (and to do the topic justice would require a separate article). For now, the main thing to know is that work cannot be assigned directly to teams. It must be *filed against a work item category* that is *associated with a team area*. This indirection comes in handy later as you separate work for new development and maintenance or when you reorganize teams.

1. Use the **Project Areas** menu (just above Rose's name in the upper-left of the page) to navigate back to the project area editor. (The Havannah project should be in the Recent Project Areas list).
2. Select the **Categories** link in the left panel.
3. On the row of the **Unassigned <Root Category>** entry, use the context menu in the first column and choose **Add category**. See Figure 20.

Figure 20. Add a new work item category

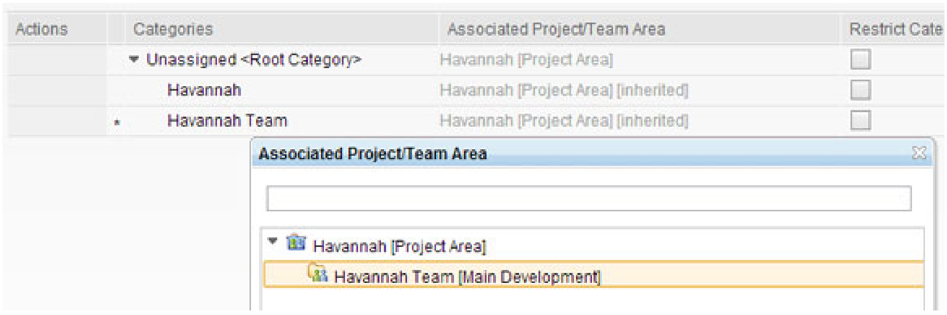


Define the categories that work items are filed against, and associate categories with the team are is the default team area used when no other team area association takes precedence. Use drag a members of the team only (Restrict Category Visibility). In addition, each category can be configure associated team (Restrict Work Item Access).

4. Enter `Havannah Team` as the new category name and select **OK**.

5. Use the context menu on the newly added category, select **Associate**, and choose **Havannah Team** in the Associated Project / Team Area dialog window (Figure 21).
6. Click **Associate** to complete the action, and then **Save** your changes.

Figure 21. Associating Havannah Team with its work item category



Actions	Categories	Associated Project/Team Area	Restrict Cate
	Unassigned <Root Category>	Havannah [Project Area]	☐
	Havannah	Havannah [Project Area] [inherited]	☐
*	Havannah Team	Havannah [Project Area] [inherited]	☐

Associated Project/Team Area

Havannah [Project Area]
Havannah Team [Main Development]

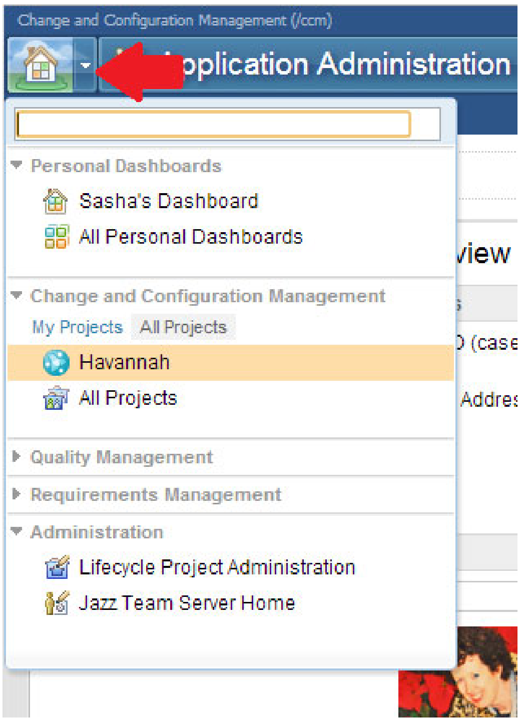
Create the Product Backlog for the project

One of the most important scrum artifacts is the **Product Backlog**.

1. To build it for this example, you need to navigate to the Product Backlog plan that was setup automatically when you created the Havannah project area. Use the **Home** menu (the one that looks like a little house) in the upper-left of the page and, under the **Change and Configuration Management** section, choose **Havannah** (see Figure 22).

Tip:
This menu is an easy way to navigate to various locations in your projects, so it is worth investigating more later.

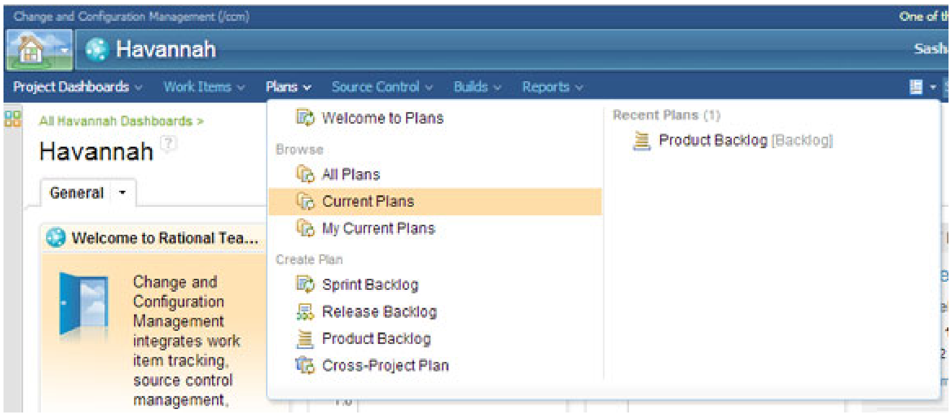
Figure 22. Navigating to project area by using the Home menu



This will bring you to the project dashboard for the Havannah project (more about this view later).

2. Use the **Plans** menu to navigate to the **Current Plans** page (after you have visited any plan, it will appear in the Recent Plans list for easier access later, as shown in Figure 23 to illustrate this feature). Until you access a plan, this feature is not visible.

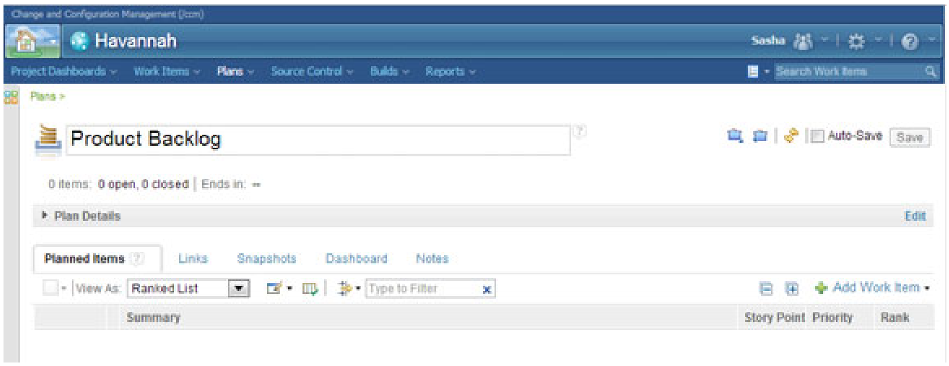
Figure 23. Navigating to Current Plans page



3. From the Current Plans page (not shown) select **Product Backlog**.

A tabbed editor will open for the **Product Backlog** iteration plan (see Figure 24). Within the **Notes** tab, you can enter information about your product by using wiki-style formatting. This is a good place to link to external documentation that should be easily available to all project members. The **Planned Items** tab (which should already be selected) is where you will build the product backlog.

Figure 24. Empty product backlog

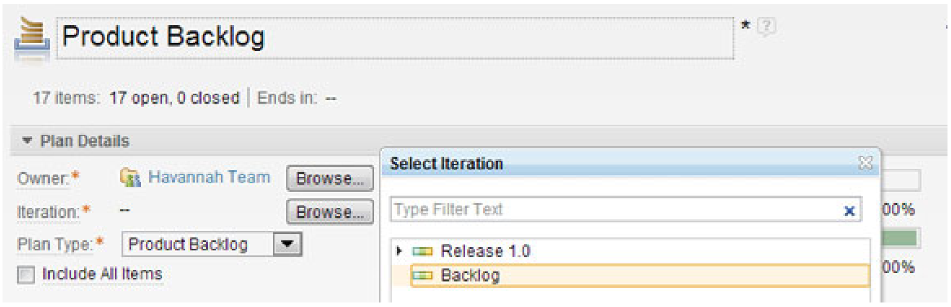


You need to make one more change before you start. Even though it is possible (and would be sufficient for this example) to have all work items assigned to the project area, rather than a team area, it's not scalable. If your project got a little bigger and you decided to have another scrum team doing some of the work, it would require a bit of rework. You would need to create two teams and then move work that was filed against the project area to them. So it's just cleaner to work with team areas all the time.

Therefore, before adding items to the Product Backlog, you need make the **Havannah Team** area its owner rather than the **Havannah** project area.

4. Click the **Plan Details** bar to display the detail area.
5. Click the **Edit** link on the bar to display the editing controls (it's on the far right, not shown in Figure 25).
6. Next to the Owner, use the **Browse** button to open a dialog window to select a new value. Select **Havannah Team**, and then click **OK**.
7. This will probably reset the assigned iteration, so use the Browse button there to reset it to **Backlog**.
8. **Save** the changes. You can click the **Plan Details** line again to collapse it to save screen real estate.

Figure 25. Edit the Product Backlog configuration


Product Backlog
17 items: 17 open, 0 closed | Ends in: --
Plan Details
Owner: * Havannah Team Browse...
Iteration: * -- Browse...
Plan Type: * Product Backlog
Include All Items
Select Iteration
Type Filter Text
Release 1.0
Backlog
00%
00%

Fill the Product Backlog with work items

Within the **Planned Items** tab of the backlog, begin adding backlog items. In scrum terminology, the items in the product backlog are called ***Stories*** or, if they are larger, **Epics** (see Mike Cohn's books, *User Stories Applied* and *Agile Estimating and Planning*, cited in Resources).

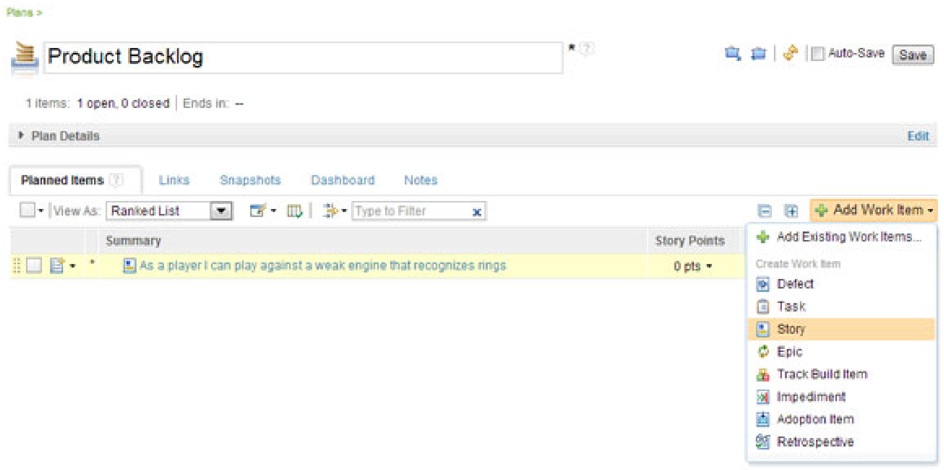
A Story is a description of functionality to be implemented in the product, often stated in terms of what user role the feature supports and what business value it offers. A properly described Story includes a summary or headline, brief description or discussion of the feature, and conditions for the feature's acceptance by the product owner.

Tip:
As you add items in this and later examples, if you use a different order than presented in the sample data, you might get slightly different results that don't exactly match the screen captures. That should not affect your overall results or the value of this tutorial.

In the Havannah example, all items are phrased as Stories. For brevity, most Stories used in this example include only the summary, but don't confuse this with a properly constructed Story.

1. Click the **Add Work Item** link on the right side of the plan's toolbar, and select **Story** to add a new Story item to the plan.
2. Enter the summary for the new item. For this example, use this summary (shown in Figure 26, which also shows the open Add Work Item menu):
   `As a player, I can play against a weak engine that recognizes rings.`

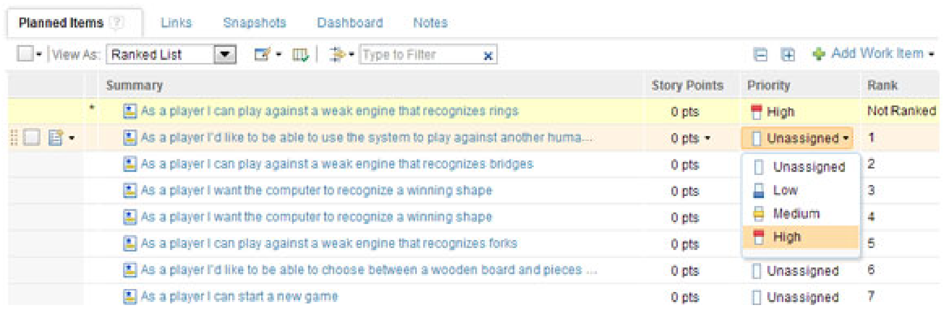
Figure 26. First entries in the Planned Items view in the Product Backlog



3. Add all other Stories requested by the product owner. See the Havannah example document in the Downloads section.
4. Click **Save** on the upper-right of the Plan tab at any time to save your work. Be sure to click Save when you are finished.
5. After you have added the Stories, the next step for the product owner is to set priorities for them. Priority attribute values are High, Medium, and Low. You can assign priorities easily by using the drop-down menu embedded in the list items (see Figure 27).

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. Then you can also rank the items within the priorities (not all Highs are created equal). You reflect this ranking by dragging the items to order them within their priority grouping (each row has a drag handle on the far left). The system remembers ordering done with the drag action. (This example does not show re-ranking of Stories.)

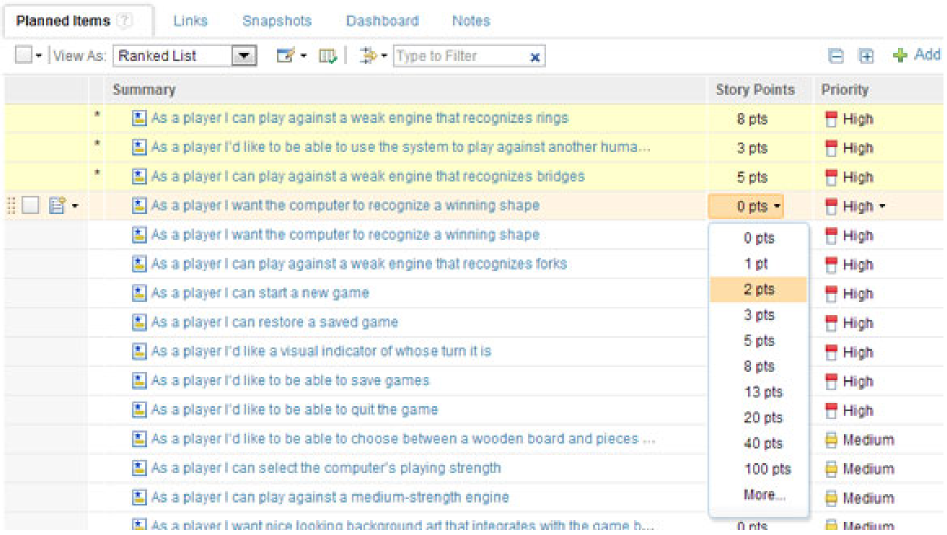
Figure 27. Set priorities for work items



Summary	Story Points	Priority	Rank
As a player I can play against a weak engine that recognizes rings	0 pts	High	Not Ranked
As a player I'd like to be able to use the system to play against another huma...	0 pts	Unassigned	1
As a player I can play against a weak engine that recognizes bridges	0 pts		2
As a player I want the computer to recognize a winning shape	0 pts		3
As a player I want the computer to recognize a winning shape	0 pts		4
As a player I can play against a weak engine that recognizes forks	0 pts		5
As a player I'd like to be able to choose between a wooden board and pieces ...	0 pts	Unassigned	6
As a player I can start a new game	0 pts	Unassigned	7

6. When the team adds an estimate for the item complexity, enter **Story points** by using the drop-down menus as Figure 28 shows. (Also see Mike Cohn's books cited in Resources, because a discussion of estimating and Story Points is beyond the scope of this article.) Assign Story Points according to the sample documentation.
7. **Save** all changes.

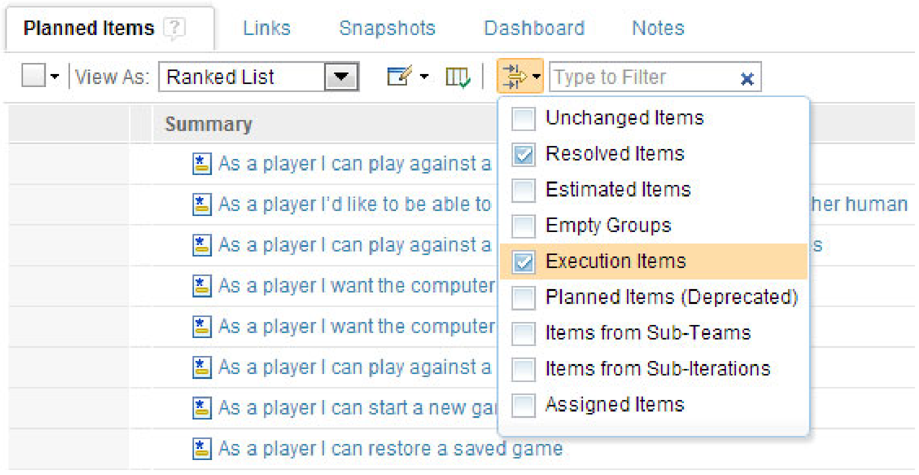
Figure 28. Set complexity for work items as Story Points



Summary	Story Points	Priority
* As a player I can play against a weak engine that recognizes rings	8 pts	High
* As a player I'd like to be able to use the system to play against another huma...	3 pts	High
* As a player I can play against a weak engine that recognizes bridges	5 pts	High
As a player I want the computer to recognize a winning shape	0 pts	High
As a player I want the computer to recognize a winning shape		High
As a player I can play against a weak engine that recognizes forks		High
As a player I can start a new game		High
As a player I can restore a saved game		High
As a player I'd like a visual indicator of whose turn it is		High
As a player I'd like to be able to save games		High
As a player I'd like to be able to quit the game		High
As a player I'd like to be able to choose between a wooden board and pieces ...		Medium
As a player I can select the computer's playing strength		Medium
As a player I can play against a medium-strength engine		Medium
As a player I want nice looking background art that integrates with the game b...	0 pts	Medium

Tip:
By default, *Epics* and *Stories* are the work items intended for a Product Backlog (referred to as **plan items** as opposed to **execution items**). In case your team also wants to see tasks in the Product Backlog (or some other work item type, such as defect), you can remove the filter for execution items from the plan's settings (see Figure 29 for how to do this, but *do not* actually change this setting). Be careful, because this could include a very large number of items in the plan, which would make it slow to load and more difficult to work with. As you can see in the screen capture, resolved items are excluded by default.

Figure 29. Removing the Execution Items filter from the Product Backlog



- ☐ Unchanged Items
- ☑ Resolved Items
- ☐ Estimated Items
- ☐ Empty Groups
- ☑ Execution Items
- ☐ Planned Items (Deprecated)
- ☐ Items from Sub-Teams
- ☐ Items from Sub-Iterations
- ☐ Assigned Items

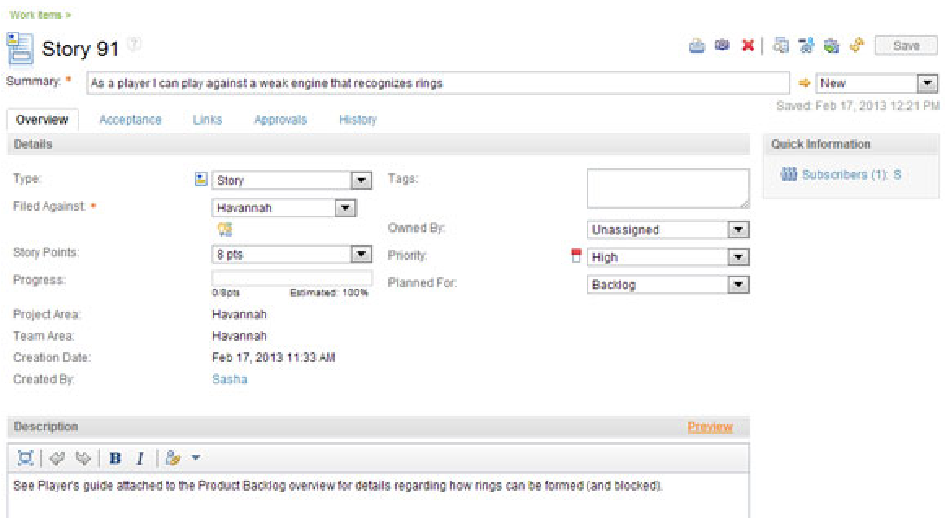
As a player I can play against a
As a player I'd like to be able to ... her human
As a player I can play against a ... s
As a player I want the computer
As a player I want the computer
As a player I can play against a
As a player I can start a new gai
As a player I can restore a saved game

Of course, it won't be much of a Product Backlog if all you have is Story summaries.

8. Click a Story's summary link to open the editor (in the example in Figure 30, the Story says *As a player, I can play against a weak engine that recognizes rings*).
9. In the **Description** field, provide additional details about what is expected of the Story. You can add whatever you like. In the figure, my example references (imaginary) documentation

that is attached to the project area. You might need to click the **Edit** link at the top of the Description field to open the editor.

Figure 30. Details for a Story item



Story 91

Summary: * As a player I can play against a weak engine that recognizes rings → New

Saved: Feb 17, 2013 12:21 PM

Details

Type:	Story	Tags:	
Filed Against: *	Havannah	Owned By:	Unassigned
Story Points:	8 pts	Priority:	High
Progress:	0/8pts Estimated: 100%	Planned For:	Backlog
Project Area:	Havannah		
Team Area:	Havannah		
Creation Date:	Feb 17, 2013 11:33 AM		
Created By:	Sasha		

Description

See Player's guide attached to the Product Backlog overview for details regarding how rings can be formed (and blocked).

10. Click the **Acceptance** tab. Add the acceptance tests identified by the team and product owner as proof that the Story was successfully completed and will meet stakeholder expectations. You can also make up whatever data you like for this practice example.
11. **Save** all changes.

Your have now established your project area and team area and prioritized Stories in your product backlog. It's time to start planning your first sprint, so see Part 2.

Acknowledgements

Many thanks to Kohji Ohsawa and Stephanie Bagot for their technical reviews and suggestions. Thanks to Mike Cohn for permission to use the Havannah example.

Downloads

Description	Name	Size
Sample files	Havannah_sample_data.zip	125KB

Resources

Learn

- Check these resources for more information related to this article:
  - Get Ready to Sprint with Rational Team Concert: Keep your agile team on track and productive by focusing on stories they can finish, also by Millard Ellingsworth (IBM developerWorks, December 2012).
  - Progressive refinement of user stories in the Product Backlog by Mike Cohn (developerWorks, January 2013).
  - Browse other Rational software articles about agile development and check the Agile transformation section of developerWorks.
  - Learn more about scrum project management on the Scrum Alliance website
  - Learn more from Mike Cohn's books, *Agile Estimating and Planning* (Prentice Hall, 2005) and *User Stories Applied: For Agile Software Development* (Addison-Wesley Professional, 2004) or on his website.
- Find out more about Rational Team Concert:
  - Find Rational Team Concert articles and links to many other resources on IBM developerWorks, and check the product overview page, features and benefits, system requirements, and the user information center.
  - Check the Rational Team Concert page on Jazz.net.
  - Watch the Using Rational Team Concert in a globally distributed team webcast or a demonstration of the Dashboards and reports, or listen to the podcast about IBM Rational Team Concert and Jazz.
- Explore the Rational software area on developerWorks for technical resources, best practices, and information about Rational collaborative and integrated solutions for software and systems delivery.
- Improve your skills. Check the Rational training and certification catalog, which includes many types of courses on a wide range of topics. You can take some of them anywhere, anytime, and many of the Getting Started ones are free.

Get products and technologies

- Download Rational Team Concert from Jazz.net (site access requires registration, no charge). You can try it **free** on up to 10 users for as long as you want (requires registration). If you'd prefer, you can try it in the sandbox instead, without installing it on your own system.
- Evaluate IBM software in the way that suits you best: Download it for a trial, try it online, use it in a cloud environment, or spend a few hours in the SOA Sandbox learning how to implement service-oriented architecture efficiently.

Discuss

- Participate in the forum at Jazz.net (requires registration, which is free). This is also where you can enter and review enhancement requests and bug reports.
- Get connected with your peers and keep up on the latest information in the Rational community.

- Rate or review Rational software. It's quick and easy.
- Share your knowledge and help others who use Rational software by writing a developerWorks article. Find out what makes a good developerWorks article and how to proceed.
- Follow Rational software on Facebook, Twitter (@ibmrational), and YouTube, and add your comments and requests.
- Ask and answer questions and increase your expertise when you get involved in the Rational forums, cafés, and wikis.

About the author

Millard Ellingsworth

Millard Ellingsworth lives in the hills west of Portland, Oregon, where he works on developing the IBM Rational Collaborative Lifecycle Management community, improving how teams work together to build software that matters. During the small pockets of free time that leaves him, he divides his attention between playing golf, noodling on the guitar, woodworking, and tinkering with Android development. You can follow him on Twitter as @millard3 and on Google+.