Rhythm 5.0

Rhythm Help

Contents

Rhythm

Welcome to Rhythm

Rhythm is an agile project planning and tracking platform designed to allow you to:

- Gather, prioritize, and manage backlog stories.
- Get comprehensive visibility of all your agile assets.
- Promote effective agile planning and execution.
- Easily track release changes and progress.
- Manage Code Reviews directly from within Rhythm in alignment with your agile process.
- Receive full visibility across the software delivery life-cycle.
- Use Scrum or Kanban agile methodologies.
- Integrate with other issue tracking and requirements management tools.

What's New

Check out the full Release Notes *here.* Here are the new features for this release:

Code Review
Relationship Tracking
Tracking Perspective Provides Metrics Based on Story Count
Update Attachment Files
General Field

Watch and Learn

Rhythm YouTube Play List　　　　　*Rhythm YouTube Play List*

Administration

Creating a Project
Adding Users to Rhythm
Adding Users to a Project
Creating a New Task Status
Input Streams: Creating a Query in Atlas
Input Streams: Configuring

Working with Stories

Creating Stories
Editing Stories
Using Discussions
Creating Relationships
Tagging Stories
Accepting User Stories
Deleting Stories
Moving Stories
Ranking User Stories
Reviewing Version Information
Importing/Exporting User Stories

Working with Tasks

Updating Tasks Status
Blocking a Task
Creating a Detailed Task
Creating a Quick Task
Copying Tasks for Reuse

Planning Perspective

Creating and Applying Filters
Creating a New Release
Creating a Sprint
Using the Sprint Panel

Featured Sections

Getting Started with Rhythm
About Stories
Administration
Planning
Team Room
Tracking

Online Resources

Online Help　　　Find your answer by viewing the *full version* of the product help on the web.

What's New in Rhythm

5.0

The following are the latest features for Rhythm.

Code Review

This release of Rhythm introduces native Code Review functionality directly from within Rhythm. Code Review enables you to:

- Understand code changes in context of defined Stories and Tasks.
- Find bugs early in the cycle before they reach production code.
- Keep code clean by supporting your team's coding standards.
- Review code changes from multiple SCM repositories from within a single Rhythm Story/Task.
- Compare differences in-line or side-by-side in a modern editor
- Quickly find historic code changes related to previously delivered Stories and Tasks.
- Keep team members up-to-date on changes in the code and ensure that the team is in agreement with changes.
- Organize and route reviews to specific reviewers.
- Have line-by-line discussions with mark-down notation.

In addition to assigning Reviewers/Watchers and adding/remove Change sets, the **Code Review** window provides a file navigator allowing you to select any file in the change set and see the differences between the files:
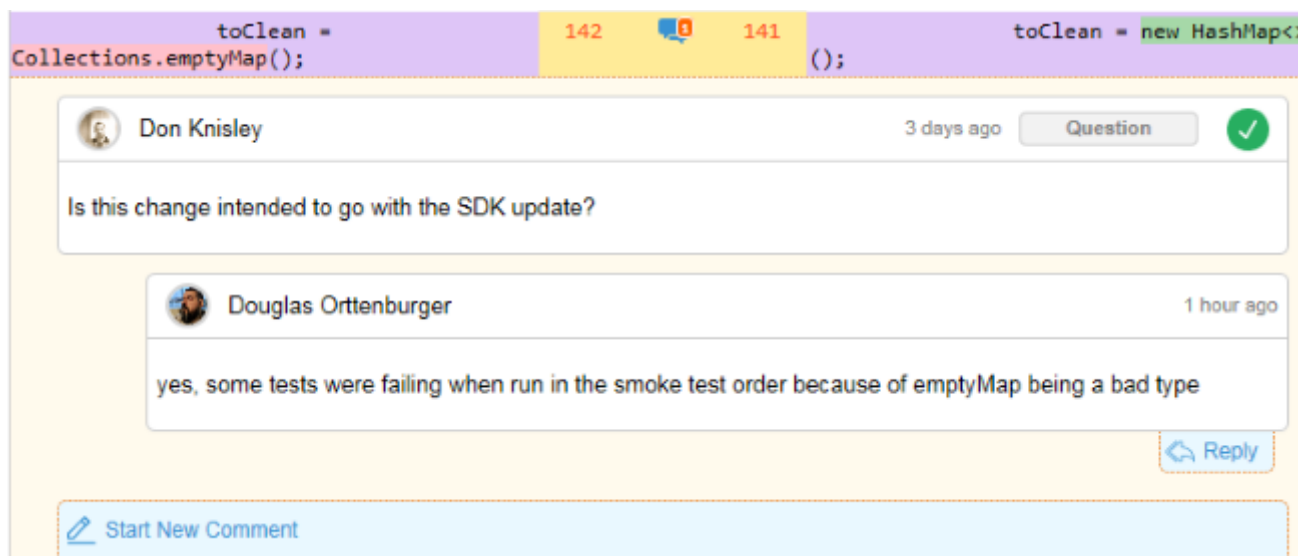
You and your team can then perform full discussions, reviews, approval/rejection cycles within a Code Review:

Relationship Tracking

You can now automatically version Stories when a Relationship to other objects (Sprints, Releases, Requirements, Stories) are added/removed. Including these types of actions in version history provides a greater understanding of plan and scope changes. For example, with version compare you can quickly understand if/when a Story was moved out of release scope. This capability also works with manually created Relationships (between Stories and Requirements). For configuration information, refer to the Installation Guide.

Tracking Perspective Provides Metrics Based on Story Count

To Support tracking of both Kanban and Scrum methodologies in the **Tracking** perspective, you now have the option of displaying tracking data based on **Story Points** or **Number of Stories**.

Updating Attached Files

When you want to make an edit to a file that is already attached to a Story, you can simply download the file 📂, make the update, and click 🗒 to upload/replace the original file.

General Field

The **General** field is a text-based field that can be used to capture any business or development process data as needed by the project team. It is not used in Rhythm for any explicit purpose. This field is only available in grids.

4.0 - 4.3

Kanban Support

This release of Rhythm introduces full support of the *Kanban* process framework. The sections below highlight all of the new capabilities.

FOR ADMINISTRATORS

The following features are available for Server or Project Administrators:

Use Scrum, Kanban, or Both!

Administrators can choose if the project uses Scrum or Kanban or both.
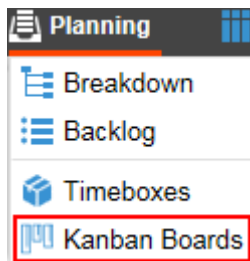

Project Configuration

Choose the Agile methodology for your project. You can change it at any time. Rhythm will display or omit relevant areas based on your selection.

Project Type:

- ◯ Scrum
- ◯ Kanban
- ◉ Scrum & Kanban

Kanban Boards Menu in Planning

When a Project is configured for Kanban, the **Planning** menu contains a **Kanban Boards** option:

Multiple Boards per Project

Use the **Kanban Boards** view to configure one or more boards for each Project. Click ⚙ to edit.

Configurable Board Backlog Source

Decide how your Board gets its Stories.

Board Backlog Source:

○ Sprint
   Stories that are assigned to a Sprint.

○ Release
   Stories that are assigned to a Release.

◉ Custom Filter

Grooming	▼

Fully Customizable Swimlanes

Add as many swimlanes as your board needs. Drag and drop their positioning.

Configure the **WIP Limit**, **Age Indicator**, and **Story Status Mapping**.

Swimlanes: 🗋 New   ⟳   🗑

	Name	WIP Limit	Age Indicator	Story Status Mapping
✓	Define	∞	1 day	Not Started
✓	Dev in Progress	6	4 days	In Progress
✓	QA in Progress	3	2 days	In Progress
✓	Functional Review	∞	No	Completed
✓	Done	∞	No	Accepted

Displaying *WIP limit* and Age Indicators

Choose display options for your board.

☑ Enforce WIP
   Do not allow Stories to be added to Swimlanes that have
   reached their WIP limit.

☑ Show Age Indicators
   Highlight Stories that have been in their current Swimlane
   for too long.

☑ Make Available In Team Room
   List the Kanban Board in the Team Room menu.

FOR ALL USERS

Team Room Menu

Select one of the boards from the menu.
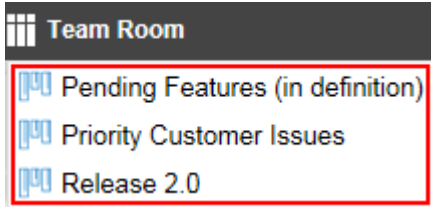
Comprehensive Info Pane

Navigate boards, review team stats, see important board information, use Discussions.

Board Backlog Lane

All Stories meeting the **Board Backlog Source** start in this lane. You can also change Sprints/Releases and use `Find...` to filter specific Stories in the **Board Backlog**.
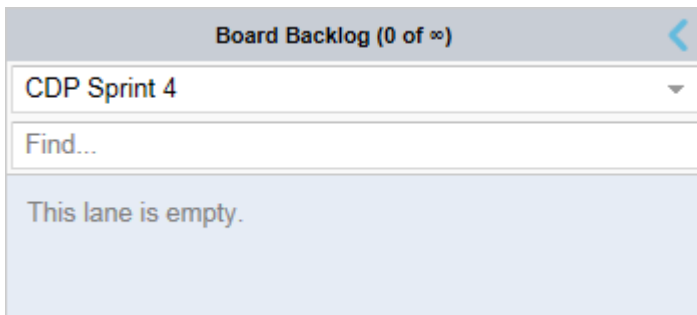
Easy to Navigate Swimlanes

Drag and drop your Stories from lane to lane.
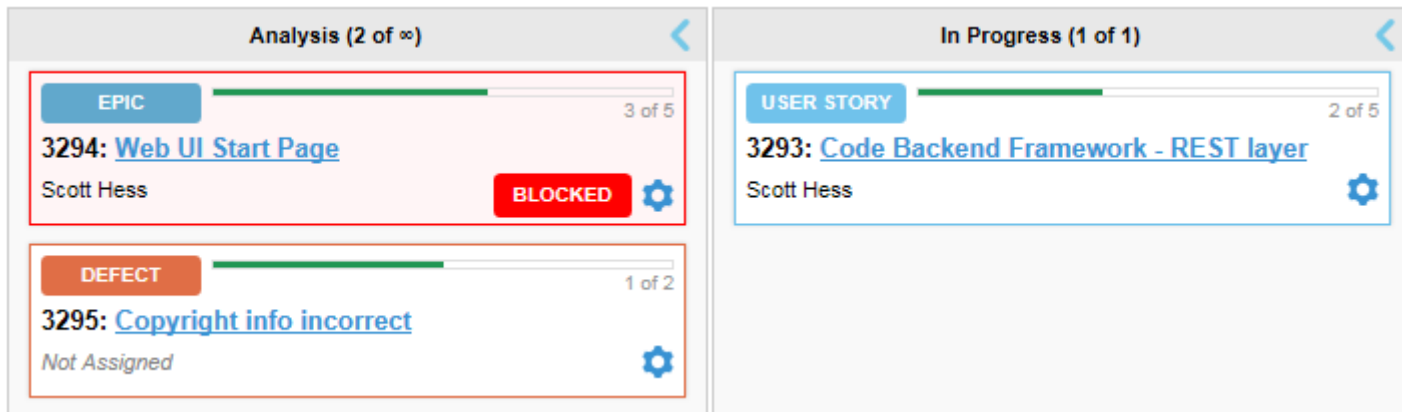
Feedback on Card Movement

Rhythm provides feedback on whether you can move a card.
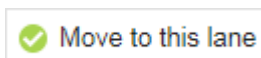
⊘ WIP Limit Reached

Interactive Story Cards

The Story Card provides relevant Story details. Double click anywhere in the card to open the **Story Editor**.
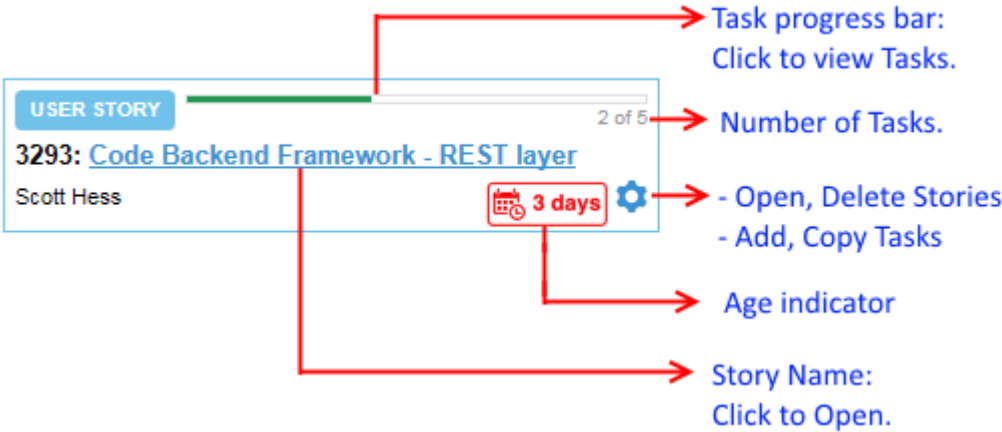
Task List Popup

Click the **Task Progress Bar** to view a Story's Tasks.

ID ↑	Type	Name	Owner	Estim...
5019	DEV	API Calls	Administrator	5.00 Hr
5020	DEV	Database Model	Scott Hess	10.00 Hr
5021	DEV	Schema Editor	Scott Hess	6.00 Hr
5022	QA	Test	Administrator	0.00 Hr
5023	DOC	DOC	Administrator	0.00 Hr

Board-Centric Filtering

Filter your Board on **User**, **Type**, or **Tag**.

User: Any ▼   Type: Any ▼   Tag: Any ▼   🔻

Quick Add

Your board is fully configured but doesn't have the Stories you need? Just add them! They are added to your first lane (after **Board Backlog**).

New Story: Quick   Detailed

Kanban Cumulative Flow

The Info Pane provides access to the Kanban Cumulative Flow diagram.
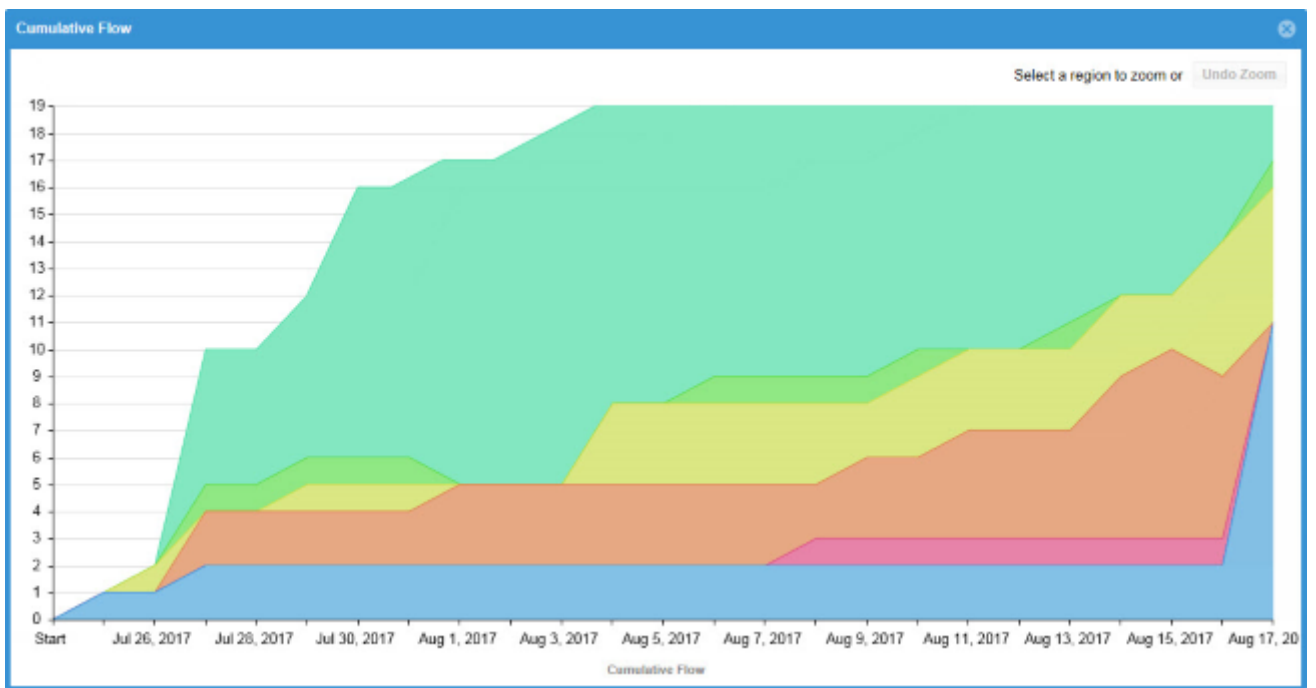
All of these features are documented in detail in the Rhythm Online Help.

Work Record Management

The **Task Editor** now contains a **Work Record** pane with a grid listing all work records for the Task. You can add/remove/edit any work record in the list.

Rollup Column

The **Rollup** column displays the total planned points for all items contained within a parent item. It is available in the **Breakdown** view and **Sprints** Panel. Leaf items show the same value in both the **Points** and **Rollup** columns. Whereas the **Points** field is a manually entered value, the **Rollup** field is calculated. This allows you to compare the two values when actual (**Rollup**) may be greater than planned (**Points**).

Tests Tab in Story Editor

The **Tests** tab in the **Story Editor** displays the test results returned by your connected testing tool.

License Management

Atlas Planning and Tracking Suite now provides the ability for Administrators to manage *Named* licenses per user, in addition to *Concurrent*.

Web-Based Training

Use the following link to order the new, free Rhythm web-based training. Order it *here*!

User Capacity Planning

This release introduces user capacity planning:

Individual User Capacity	Each user can now customize their availability at both the Project level and the Sprint level. You can see these values reflected for each user by using a Quick Filter for a user in the **Team Room** and also in the Task **Owner** field.

|	
Administrator Manages Availability	Although individual users can modify their own availability (see above), each administrator can open the **Edit Sprint** window and use the **Team Availability** grid to change the availability for each user for the *current* Sprint and Project.
	Note: This applies to *Private* Projects only.
Task Owner Assignment	To aid in deciding who is available to work on a certain Task, the **Team Room** and **Story Editor** now display all users availability when you attempt to assign a user to a Task that is in a Sprint.

User A	
Remaining: 1.00 Hr Total: 11.00 Hr	
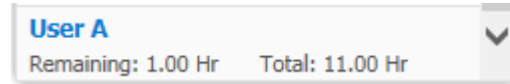
This indicates if a team member is over- or under-allocated for the Sprint.

Quick Find

When you are in the **Breakdown** and **Backlog** views of the **Planning** perspective, you can use the **Quick Filter**'s text field to search for text in the current view. This capability is invaluable for quickly finding items in the **Backlog**:

1. Place cursor in the **Quick Filter**'s text field.
2. Type in the text to search for.
3. Click `Enter`.

This will behave exactly as if you opened the **Filter** dialog box and created a filter using `Name Contains <Text>` or `ID Equals <Text>` (the second part is only if the text is numeric).

Search

Rhythm now provides search capabilities for Stories. Using the search, users can easily scan all Stories and their attributes, including the content of attached files. Used together with the Quick Find and Filtering, all data is quickly at your finger tips.

Change Package Support

Rhythm now provides visibility to change sets/packages and their contents directly from the Rhythm user interface. When your native SCM environment is connected to the Atlas Hub, Rhythm includes change sets/packages in the relationship diagram and Story/Tasks lists. Understanding related files and impacts provides users greater understanding of work context.

These Stories display the Change Packages in the **Team Room** via the **Impact** column.

Impact

</> 3
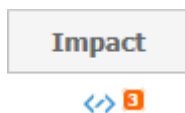
Additionally, any Change Packages associated with a Task are visible in the **Story Editor**.

When viewing relationships in a Story, Change Packages display in the diagram and you can click them to open the **Change Packages Viewer**.
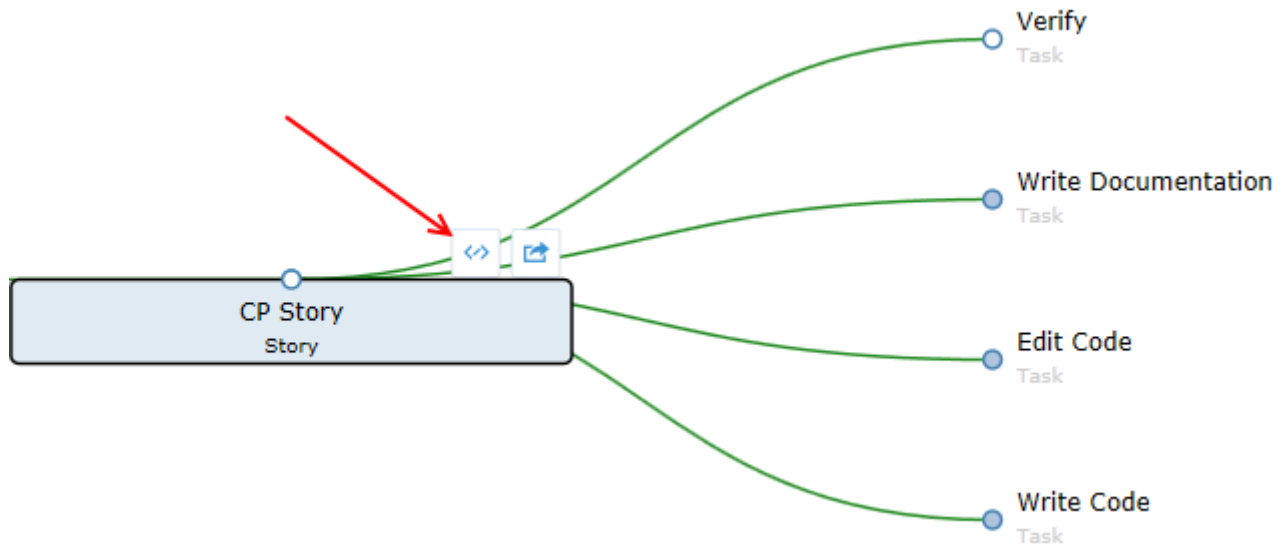
Accessing Stories in Native SCM Environments

Rhythm provides direct Story and Task links to the StarTeam Cross-Platform Client when Atlas Hub file activity is detected. Task links can be used to establish StarTeam Cross-Platform Client work context for developer activity, for example, check-ins. Check-in's in task context automatically create related changes packages that are in turn visible in Rhythm. Just click the **ID** field in the **Story Editor**.

Relationship Impact Management

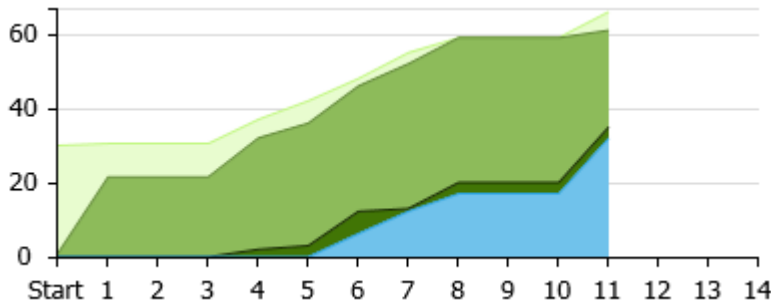Rhythm now provides more ways to understand related work and the impact of change. Using the relationship diagram, users view related items and can filter on item type, relationship type, or relationship status. This aids in quickly understanding how data is related and how change propagates through related items.

Team Room Cumulative Flow

The **Team Room Cumulative Flow** chart represents the amount of work in the Sprint in different states for each day of the Sprint. The vertical axis represents the points in the Sprint. The horizontal axis represents each day of the sprint. It includes **Points Not Started** (light green), **Points In Progress** (green), **Points Completed** (dark green), and **Points Accepted** (blue). The height of the chart defines the scope. If something is added or the estimate is increased, then the whole chart gets taller. If something is removed or the estimate decreased, then the chart gets smaller. It provides the ability to see changes in scope in the sprint.

Additionally, when a Quick Filter is applied, the chart adjusts to the items selected in the Filter.
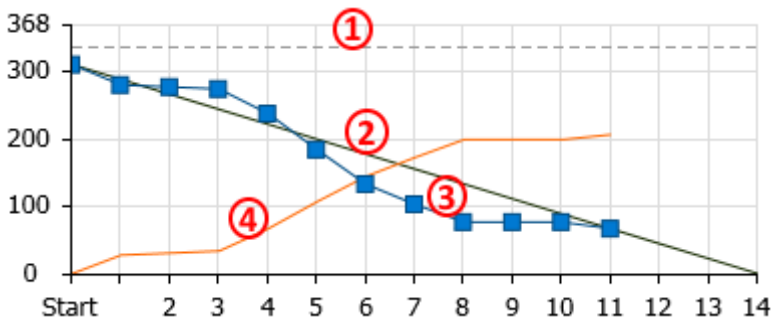
Tracking Cumulative Flow

The **Tracking Cumulative Flow** chart represents the amount of work in the Release in different states for each Sprint of the Release. The vertical axis represents the total number of Points or Stories in the Sprint (depending on whether the release was planned and tracked with Points or Stories). The horizontal axis represents each Sprint of the Release. It includes **Points Not Started** (light green), **Points In Progress** (green), **Points Completed** (dark green), and **Points Accepted** (blue). The height of the chart defines the scope. If something is added or the estimate is increased, then the whole chart gets taller. If something is removed or the estimate decreased, then the chart gets smaller. It provides the ability to see changes in scope in the Release.

Team Room Burndown Chart

The **Team Room Burndown** chart now displays (1) Capacity (2) ideal Burndown (3) current Burndown and (4) Burnup. Additionally, when a Quick Filter is applied, the chart adjusts to the items selected in the Filter.

Sprint Activity Window

The **Sprint Activity** window is available in the **Team Room** for the **Sprint List** and **Sprint Cards** views. It provides activity that occurs on all Tasks, Stories, and Discussions for all users. You can filter Tasks by **Date** (From, To), **Type**, **User** and many more (see below). You can use this information to:

- Provide detailed support or explanation for data in the **Team Room** Charts (status changes, work record hours, etc).
- Filter by **Type** to see what was `Added/Created` or `Removed/Deleted` and by whom. You can also isolate Tasks by: `Edited`, `Worked`, `Discussed`, `Carry Over`, `Blocked/Unblocked`.
- Use the data however you need to as the basis for your daily stand-ups or Sprint retrospective.

From the top-right of the **Team Room**, click the **Sprint Activity** button: .

Decorators for Story Tabs

In the **Story Editor**, the tabs for **Discussions** , **Attachments** , **Relationships** , and **Tags** will now show orange decorators when items exist.

Team Room Points By Type

The **Team Room Points By Type** chart is now clickable and will apply a filter with the items selected. Use this bias chart to quickly understand the allocation distribution for Epic, Themes, Stories, Defects, and Tech Todo's. Additionally, when a Quick Filter is applied, the chart adjusts to the items selected in the Filter.

Team Room Health Auto-Filters

The **Team Room Health & Progress** chart is now enabled with clickable auto filters. When warning labels are clicked, for example `1 Story without Tasks`, the full **Team Room** is automatically filtered to show only those items matching the criteria. Locating health and progress issues has never been easier.

Filter Sensitive Team Room Charts

Team room filtering has been updated to apply to not only the grids, but also to all visible charts and health. As selection filters are applied/removed, **Team Room** charts and health are updated to reflected the target data. The available filtered **Team Room** charts enable you to quickly identify trends and problem areas. Click the clear icon () to clear any filter.

Copy Tasks Preview Window

The new **Import Task** preview capability enables you to quickly identify which Story to copy Tasks from. It provides a one-click action to provide visibility to selected Story Tasks before electing to copy them. This lets you accurately copy predefined Tasks from existing Stories and templates.

Project Quick Switch

You can now use the **Switch Project** menu (available in all perspectives) to quickly change from the current Project to any of your *Favorite* Projects.
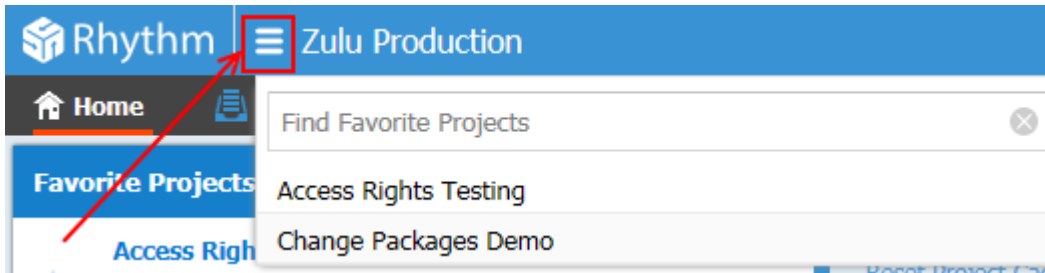
Attachments

Stories allow you add file attachments, just like Atlas. Refer to the topic titled *Using Attachments with Stories* in the online help.

Bulk Edit in Grids

You can now perform bulk edits on multiple rows for any visible column in grids.

Customizable Swimlanes

Task **Status** values are now customizable.

For Administrators	In **Admin Tools**, click the **Task Status/Swimlanes** tab to create new **Status** values and enable the ones you want the team to use.	
Users	**In grids and the Story Editor**	The **Status** control will contain the number of **Status** Values that are defined. Just click the Value.
	In the Team Room	When you select a Story in the **Sprint Cards** view, your tasks display in columns based on the Status Values. Simply drag-drop the Task to change the **Status**.

Discussions

Stories in Rhythm allow you to have Discussion Topics, similar to those in Atlas. Refer to the topic titled *Discussions* in the online help.

Import/Export

You can now import Stories into or export Stories out of a CSV file.

Import	Use your favorite CSV editing tool, such as Microsoft Excel, to create Stories and import directly into Rhythm.
Export	Export selected Stories out of Rhythm and use your tool of choice to consume the data.

Input Streams

Input Streams ( ) enable business and quality teams to work in seamless cadence with the Agile development team. Using Micro Focus Connect, *Input Streams* allow you to pull in external Requirements and Defects into the Rhythm **Breakdown** view for you to review and discuss. If you then choose to *Accept* them, new Epics/Defects are created in Rhythm with clear traceability back to the original external asset. This link provides a mechanism for feedback to the original author.

Relationships

You can now create direct, traceable relationships from Story to Story. Further, you can view and track the relationship directly back to the Requirement or other related asset in Atlas and monitor and action any changes. Refer to the topic titled *Relationships* in the online help.

StarTeam Agile Replacement

StarTeam Agile customers should now use Rhythm as their agile tracking tool. If you plan to continue to use StarTeam Agile, you need to open the `ALMConfiguration.xml` file and change this value: `<goToStarTeamAgile>TRUE</goToStarTeamAgile>`. Additionally, refer to the installation guide to learn about enabling the StarTeam Agile .WAR file.

Tags

You can now *Tag* Stories. This provides the ability to tag similar Stories with relevant labels to group and review the items together. Refer to the topic titled *Using Tags* in the online help.

Versions

Stories now save versions for every time the Story is changed. You can easily see how the Story changed by comparing the differences. Refer to the topic titled *Viewing and Comparing Versions of a Story* in the online help.

Rhythm

Rhythm is the new delivery-focused application within the Atlas Planning and Tracking Suite allowing agile teams to breakdown and manage work delivery in agile cadence. Some features include:

- Enables definition and breakdown of program objectives into actionable stories.
- Defines/manages sprints and team releases.
- Collaborate and status with the Team Room's List and Card views.
- Tracks progress against defined dates and payloads.

Getting Started with Rhythm

Welcome to Rhythm! Rhythm is an Agile delivery management platform where Agile execution teams plan, status and track their work. Agile teams can organize their **Backlog**, plan Releases and Sprints, run daily stand-ups, and track delivery progress. In Rhythm, you can use either the Scrum methodology or the Kanban methodology.

When Rhythm is used together with the Atlas Requirements capabilities, the Atlas Planning and Tracking Suite provides full program definition and delivery management - keeping the business in sync with the delivery team.

⚠️ **Important:** Before proceeding, make sure you are familiar with the ways in which Atlas and Rhythm integrate. See *Using Rhythm with Atlas*.

1. *Start Rhythm*.
2. **Create a Project**
   a) The administrator can use an existing Atlas Project or *create a new Rhythm Project*.
   b) The administrator needs to set Project access rights for Users. See *Setting Project Access Rights*.
   c) *Set the Project Type*, either `Scrum` or `Kanban` (or both).
   d) If you want to customize Task **Status Values** beyond the default values (`Pending`, `In Progress`, `Finished`) so that the **Team Room Sprint Card's** view Swim lanes reflect this (as well as the **Status** control in the **Task Editor**), see *Changing Task Status Values*.

e) **SCM Tool Configuration** - Native Code Review configuration is provided by default for the StarTeam SCM tool (Atlas Hub default). If you want to use a different supported system, see *Creating a New SCM Configuration*.

3. **Build and break-down the Backlog**

    The Product Owner needs to *Breakdown Epics into consumable Stories* for the project team. This work is done in the *Breakdown View*. Additionally, the Product owner can:

    a) Push Requirements from Atlas into Rhythm as Epics. See *Using Rhythm with Atlas*.

    b) Pull Requirements (and Defects) into Rhythm from the Atlas Hub. See *Configuring Input Streams* and *Input Streams*.

4. Decide which Agile methodology you want to use:

    a) If you use traditional set milestones for incremental delivery, see *Using Rhythm with Scrum*.

    b) If your team is more about the flow of work, see *Using Rhythm with Kanban*.

Starting Rhythm

To start Rhythm directly from a browser, type in: `http://<hostname>/atlas/#perspective=agile`.

If you receive a message similar to `This application has an update, reload?`, click **Yes** to ensure you are using the latest version installed on the server.

To start Rhythm from within Atlas:

1. With Atlas opened, hover over the **Select Application** button and click it to display the options.
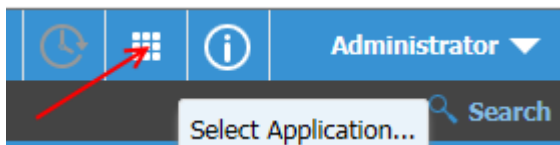
2. Select **Rhythm**. A new browser tab opens the application.

Using Rhythm with Atlas

Rhythm and Atlas are two of the main components of the Atlas Planning and Tracking Suite. You use Atlas to define, to capture, and to track the business needs of the organization as Requirements. You use Rhythm to plan, to status, and to track the day-to-day Agile activities of the project team. Atlas pushes Requirements into Rhythm as Epics where the product team breaks them down into smaller consumable Stories. These Stories are then worked through the full Agile process. Daily status, state and all related tracking data is integrated back into Atlas for real-time portfolio visibility for all stakeholders.

The specific integration points are as follows:

Projects

You can create new projects in Rhythm, but Projects are *common* between Atlas and Rhythm so any Project created in one is available in the other. For more information, see *Working with Projects*.

Users

Although Users are common between Atlas and Rhythm, the administrator needs to set project access rights for each user. See *Setting Project Access Rights*.

Starting and Logging In

You can open Rhythm from within Atlas, but there are other ways to start the application:

- An administrator logged into the install machine can use the **Start** menu to launch Rhythm: **Start** > **All Programs** > **Micro Focus** > **Atlas** > **Rhythm**.

- Other team members can navigate to: `http://<hostname>/atlas/#perspective=agile` to start Rhythm.

Requirements

You *Push* Requirements from Atlas into Rhythm as Epics, using one of the following methods:

Individual Requirement	From the **Actions** menu ⚙ of a Requirement, select **Send to delivery Backlog**.
Full Plans	In the Atlas **Plan** perspective, edit a plan and click **Send to delivery Backlog**.

💡 **Tip:** You can also *Pull* Requirements and Defects into Rhythm using *Input Streams*. Any defect or requirement that was synchronized into the Atlas Hub using Micro Focus Connect is available to be added into Rhythm. For more information, see *Input Streams*.

Proceed to *Getting Started with Rhythm*.

Choosing an Agile Methodology

Both Kanban and Scrum are lean/agile methodologies and are highly adaptive. They are pull-based scheduling systems that enable the JIT (just in time) principle of Lean. Scrum is more prescriptive with key roles and ceremonies pre-defined. Kanban is flow-based and provides the ability to change priority at any time. Rhythm provides the ability to use either or both methodologies together.

A summary of some key differences:

	Scrum	Kanban
Work Assignment	Work is pulled into sprints in batches matching Sprint cadence.	Work is pulled in as other work completes.
Dates/Delivery	Value is delivered in staged increments (Sprints). Dates are set in advance.	Value is continuously delivered. Due dates set as needed.
Change/Adjustment	Change occurs at Sprint boundary (mid-Sprint changes are strongly discouraged).	Since work is not pre-assigned, change can happen at any time.
Ideal Application	Best with stable priority and incremental delivery.	Projects with very dynamic content, and widely varying priority.
Measurements	*Velocity* (average delivery per sprint).	*Cycle time* (time for item throughput).
Government	*Capacity* controlled (projected work per sprint).	*WIP* (work in progress) limited.
Process	Lightweight process with roles and ceremony, but still a prescriptive process.	Easily adapted to custom workflow and process.

Both are valid choices for managing delivery, but one may be better suited to your specific project. Feel free to experiment with your project configurations until you find the best process for your team.

Using Rhythm with Scrum

Before proceeding with these steps, make sure you have reviewed *Getting Started with Rhythm*.

✏️ **Note:** The Project *Type* is set to `Scrum` by default, but if you will be using both `Scrum` and `Kanban` in your Project, see *Setting the Project Type*.

1. **Plan the Sprint and Release Timeboxes**

a) The Scrum Master should define one or more of the product's Releases:

- *Create a Release*.
- *Set the Weekly Capacity of a Release* (in Story Points).

b) For Sprints, the Scrum Master needs to:

- *Create a Sprint from the Backlog*.
- *Set the Sprint's Capacity* (in Story Points).
- Ensure that all team members individually enter their availability for the Project/Sprint. See *Managing User Capacity*.

2. **Groom the Backlog for delivery planning**

   The Product Owner should:

   a) *Refine and Rank the Backlog*.
   b) *Target Stories for a Release*.

3. **Sprint Planning**

   a) The development team should conduct a planning meeting in order to *estimate Stories*.
   b) After the Stories are estimated, the team and Scrum Master need to *Add Stories to the Sprint* from the 🖥 **Planning** > ☰ **Backlog** > 🏃 **Sprints Panel**, keeping in mind the indicators provided in the **Health & Progress** area. If you are over capacity, it will tell you. It also indicates if Tasks/Stories are/are not estimated and have owners.
   c) The team members *create Tasks* for their Stories.

   - Estimate the effort for each Task.
   - Assign an Owner to each Task. See *Assigning and Estimating Tasks*.

     💡 **Tip:** It is likely that many of your Stories will contain the same types of Tasks. If so, we recommend setting up a template Story with all Tasks and using that on which to base each new Story. See *Copying Tasks from Other Stories*.

4. **Sprint Execution**

   When the team is in agreement upon the Stories, the team should commence work and use Rhythm every day in the *Team Room*. Use the Team Room's **Sprint List** and **Sprint Cards** to do the following:

   - The Scrum Master runs the daily stand ups, checking on the Sprint health. Use a Quick Filter to filter what data you want to see in the **Team Room**. Note that the graphs in the **Team Overview** change based on the filter.
   - The project team members *add work hours to Tasks* and Status them as needed.
   - The team members also use Rhythm's *integrated Code Review feature* to review, comment on, and accept all code changes.
   - As Stories are completed, the Product Owner uses the Team Room to review and *Accept* them.

5. **Release Tracking**

   The Product Owner and other stakeholders monitor the Release in the *Tracking* perspective.

Using Rhythm with Kanban

Before proceeding with these steps, make sure you have reviewed *Getting Started with Rhythm*.

1. **Define your Board(s)**

   The Administrator is able to create and configure the Kanban board(s) to match the needs of the team. Once the board is defined, the **Board Backlog** lane is dynamic, based on the Filter/Sprint/Release selected in the **Board Backlog Source** option. When a Story is moved out of the **Board Backlog** and added to a lane, it will remain in that lane even if the **Board Backlog Source** changes. You can then progress the Story through the lanes accordingly, or you can explicitly perform a `Remove` action to remove it from the board.

   a) *Set the project type* as either `Kanban` or `Scrum & Kanban` from the **Project Settings**.

b) *Create a Board*. Each project can have multiple boards.

c) *Define where the board gets its Stories*. All Stories that meet the condition you select will be added to the **Board Backlog** lane. Select either:

Sprint	Select this option to use your Sprints as the source for your board's Stories. Typically only used if you are moving from Scrum to Kanban or if you plan to use both together. If you select this option, you first need to *Create a Sprint from the Backlog*.
Release	Select this option to use Releases as the source for your board's Stories. This option allows you to define a target payload for a specific release and use a board to manage its delivery. Using this option allows you to use Rhythm's **Tracking** perspective. If you select this option, you first need to *Create a Release*.
Custom Filter	Select this option to use a pre-defined custom filter as the source for your Board's Stories. This option provides the ultimate flexibility. You can include anything that you can filter on, including Tags. For more information, see *Working with Filters*

d) *Create the Swimlanes*.

> ✏️ **Note:** The default first lane named **Board Backlog** is not editable and it is the starting lane for all items that meet your criteria in the previous step.

e) *Select other Board options such as adding to Team Room, Enforcing WIP, and Showing Age Indicators*.

2. **Identify the Items to Include in the Board**

Individual team practices may require item review to assure appropriate content before items are on the board. The Product Owner should modify items in the Backlog to fit the selected input criteria (**Board Backlog Source** option).

Best Practice: Create a *Grooming* board for the Product Owner to prepare Stories and then a *Delivery* board for the team to use that matches their cadence. Do the following:

a) In the **Backlog**, create a Custom Filter using a **Tag** or other property.

b) Create a Kanban board and set the **Board Backlog Source** as the previously created Custom filter.

c) Groom the Backlog and set the property in the relevant Stories so that they display in the appropriate board.

3. **Work Planning**

The following items are optional and only apply if you are using actual/estimated point size and recording Tasks.

a) The development team should conduct a planning meeting in order to *estimate Stories*.

b) The team members *create Tasks* for their Stories.

- Estimate the effort for each Task.
- Assign an Owner to each Task. See *Assigning and Estimating Tasks*.

> 💡 **Tip:** It is likely that many of your Stories will contain the same types of Tasks. If so, we recommend setting up a template Story with all Tasks and using that on which to base each new Story. See *Copying Tasks from Other Stories*.

4. **Kanban Execution**

a) Navigate to ⊞ **Team Room** > ▦ **<Kanban Board Name>**.

b) *Drag-drop your cards into appropriate lanes*.

c) *Edit Stories* and review Tasks within the *Kanban Card Task List*.

d) Use Rhythm's *integrated Code Review feature* to review, comment on, and accept all code changes.

e) Review your ongoing ***Cycle Time*** and ***WIP Summary***.

f) Check the *Cumulative Flow*.

g) *Add Stories explicitly to your board*.

h) *Run Filters*.

i) Use *Discussions* in your board.

5. Release Tracking (If using Releases)

   The Product Owner and other stakeholders monitor the Release in the *Tracking* perspective.

Home

The **Home** perspective ![home icon] provides access to the following:

Projects	Projects are listed and managed in the left pane. See *Working with Projects*.
System Administration ![gear icon]	Administrators use the System's **Admin Tools** to manage Users and to set Project access rights. See *Administration*.
Reset Project Cache ![reset icon]	Click to reload the Project from the database into the web server. ⚠️ **Important:** This should only be used if you suspect that there are data synchronization issues between your web server and the database. Contact customer care before using this functionality.
Activity view ![activity icon]	The **Activity** view displays in the **Home** perspective. You can see all recent Story activity. See *Using the Activity view*.

Using the Activity View

The primary use for the **Activity** view is to display a list of Stories that have recently been modified. You can do the following:

Limit Stories that display	Click the calendar ![calendar icon] to select the **Start Date** for the **Activity** view. Click ![pin icon] Pin Now to have the Calendar start at the current day-time.
See the Work Progress	Hover over a Story's Status bar ![status bar] and it displays the **Actual Work** and **Estimated Work** hours for all the Tasks in the Story. For example: Worked hours: 1.1 Estimated hours: 3.5
Open a Story	Click **Goto Story** ![goto icon] to the far right to open the Story.
Comparing Version Differences	Click **View Changes** ![view changes icon] to open the **Compare Versions** dialog box. See *Viewing Changes in a Story*.

Viewing Changes in a Story

The **Compare Versions** dialog box allows you to view two different versions of a Story. The fields appear in a grid in two columns next to each other.

1. Click ![home icon] **Home**.

2. In the **Activity** view, select a Story and click ![view changes icon] **View Changes**. The **Compare Versions** dialog box opens with two columns displaying the changes next to each other.

3. Click **Options** > **Only show differences** to limit the grid to fields that have changed.

4. Click **Options** > **Extended Attributes** to show all fields including **Extended Attributes**.

Working with Projects

Projects contain Backlogs, Releases, and Sprints. You can add or edit Projects in the **Favorite Projects** or **Available Projects** lists on the **Home** perspective.

Use the **Project Access Rights** tab in **Admin Tools** to:

• Limit which users can see the project.
• Reset permissions for the Project.
• Make the Project public.

⚠️ **Important:** Any Projects that you create in Atlas are automatically available in Rhythm so you don't need to re-create them. However, you will still need to set project access rights for each user. See *Setting Project Access Rights*.

Creating a Project

The first step to managing your Projects is to create a project to organize your Backlog and manage your Sprints.

💡 **Tip:** Watch how to do it: *Create a Project*.

1. Click 🏠 **Home**.
2. Click 🗋 **New Project**. The **Create Project** dialog box appears.
3. Type the **Name** of the Project.
4. Choose the visibility option (`Public` or `Restricted`) from the list.
5. Enter the **Project Description**.
6. Click **Save**.
7. To make the Project a *Favorite*, click the **Star** button. When a Project is a *Favorite*, it appears in the **Favorite Projects** list in the Home perspective and is also available in the **Switch Projects** list ☰.

After creating your Project, you need to set up user access to the Project. See *Setting Project Access Rights*.

Editing a Project

1. Click 🏠 **Home**.
2. Click **Edit** (✏️) in the **Favorite** or **Available Projects** list. The **Edit Project** dialog box appears.
3. Make changes to the Project **Name** or **Description**.
4. Click **Save**.
5. To modify the Project visibility (`Public` or `Restricted`), see *Setting Project Access Rights*.
6. To make the Project a *Favorite*, click the **Star** button. When a Project is a *Favorite*, it appears in the **Favorite Projects** list in the Home perspective and is also available in the **Switch Projects** list ☰.

Switching Projects

Whenever you mark a Project as a *Favorite*, you can easily switch to that Project from any perspective.

1. In the top left of the application, click the **Projects** icon ☰ to the left of the Project name. The list of your *Favorite* Projects displays.
2. Select the Project to switch to. The active Project changes.

Administration

This section contains information that is relevant to administrators. There are some tasks that only administrators can perform.

Setting Project Access Rights

Project Administrators can add/remove users from Projects based on role. The following roles are available:

Viewer
A *Viewer* can see the Project in the list of Projects, can add it as a favourite, and view all of its assets. *Viewers* can't edit or delete assets. *Viewers* can perform operations such as generating reports, but generally they can't change the data or the configuration of the Project.

Contributor
A *Contributor* can do everything a *Viewer* can do plus they can edit the Project data. They cannot modify the configuration of the Project (no access to **Admin Tools**).

Administrator
A project *Administrator* can do everything a *Contributor* can, plus they can modify Project configuration via the **Admin Tools**.

> **Note:** A *Project administrator* is not the same thing as a *Server administrator*. In order to be a *Server administrator*, another *Server administrator* needs to apply `Server Administrator` permissions to that user in the **Privileges** group of the **Users** tab. See **Managing Users**.

1. Log in with Project Administrator credentials.
2. Click **Home**.
3. Click **Admin Tools**.
4. Enter the user **Privileges**:

   Check the **Server Administrator** check box to make the user a *Server Administrator*.
5. Enter the user **Login Status**:

   Check **Suspended** to lock the user out.

   If the Atlas Hub has Active Directory configured, you can use it by checking **Active Directory**.
6. Click **Project Access Rights**.
7. In the **View** pane, there are two sections: **Atlas** and **Rhythm**. Each has **Viewers** and **Contributors** buttons. Click one of the buttons. The list of users with that role in the Project is displayed.
8. To add a user with that role, hover over a user in the **Available Users** list and then click **+**. The user is added to the Project in the role that you selected.
9. Click **Make Public** to make the Project public. If you don't see that option, click and then select **Make Public**.
10. Click (**Reset Permissions**) to remove all users from roles in the Project. Doing this will give each user access to the Project as a *Contributor*. However, only users with *Server administrator* credentials can manage the Project.

User Account Management

User management includes adding users to the system as well as managing project teams. Only administrators can add people to the system. Once a user is added, they can create Projects and project teams.

Once you add a project team to a Project, the Project will only be visible to team members. It is visible to everyone until you create the project team.

Roles

The following roles are available:

Viewer	A *Viewer* can see the Project in the list of Projects, can add it as a favourite, and view all of its assets. *Viewers* can't edit or delete assets. *Viewers* can perform operations such as generating reports, but generally they can't change the data or the configuration of the Project.
Contributor	A *Contributor* can do everything a *Viewer* can do plus they can edit the Project data. They cannot modify the configuration of the Project (no access to **Admin Tools**).
Administrator	A project *Administrator* can do everything a *Contributor* can, plus they can modify Project configuration via the **Admin Tools**.

> **Note:** A *Project administrator* is not the same thing as a *Server administrator*. In order to be a *Server administrator*, another *Server administrator* needs to apply `Server Administrator` permissions to that user in the **Privileges** group of the **Users** tab. See **Managing Users**.

> **Note:** A user can have multiple roles based on the projects they are working on. For example, a project administrator for `Project A` could be a `Project Viewer` on `Project B` while a viewer on `Project B` could be a contributor for `Project A`. The roles and permissions are project based, with the exclusion of the `Server Administrator`.

Adding Users

Only Administrators can add users to the system. Once user accounts have been created, then project managers can add them to project teams.

> **Tip:** Watch how to do it: *Add Users to Rhythm*.

> **Tip:** Click the **Gear** menu ⚙ at the top of the **User Information** pane to enable editing.

> **Note:** You can view the Server License Usage status for all users. If you have recently made a change, click **Refresh License Info** to update the status information.

1. Log in with Project Administrator credentials.

2. Click 🏠 **Home**.

3. Click **Admin Tools**.

4. Click the **Users** tab.

5. Click **New User**. (To edit an existing user, select a user from the **Available Users** list and click ⚙). Enter the following details:

Logon Name	Enter the name with which the user will use to log on.
Distinguished Name	If using **Active Directory**, enter the user's **Distinguished Name**.
Password	Enter the user's password. This can be changed by the user later.
Confirm Password	Re-type the password.
Name	Enter the name of the user as you want it to appear within the system. This is the name that will appear on assigned assets.
Email Address	Enter the user's email address.

> **Important:** In order to receive email notifications from Discussions, you must enter a valid email address.

Phone Number	Optionally enter the phone number,

6. Enter the user **Privileges**:

   Check the **Server Administrator** check box to make the user a *Server Administrator*.

7. Enter the user **Login Status**:

   Check **Suspended** to lock the user out.

   If the Atlas Hub has Active Directory configured, you can use it by checking **Active Directory**.

8. Choose the License Type.

Named License	Can be used only by the user who has been assigned that license. Users who receive the named user licenses are guaranteed access.
Concurrent License	Users share the licenses and can log on as long as there are concurrent licenses available.
No License Assigned	Users without a license assigned cannot log on.

You can add as many users as you choose, but access is granted only to users with named user licenses or to users who receive concurrent licenses as they log on. If you have named user licenses, you must assign them to specific users.

Editing Users

1. Log in with Project Administrator credentials.
2. Click 🏠 **Home**.
3. Click **Admin Tools**.
4. Click **Find User** at the bottom and enter the user's name in the search field. The matching users will appear below the search box.
5. Select the user in the list.
6. Click the **Gear** menu ⚙ at the top of the **User Information** pane to enable editing.
7. Edit the proper fields.
8. Click **Save**.

Changing User's License Type

1. Log in with Project Administrator credentials.
2. Click 🏠 **Home**.
3. Click **Admin Tools**.
4. Click the **Users** tab.
5. Select a user.
6. Click the **Edit User** icon ⚙.
7. In the **License** group, select a type from the **Assigned Type** list.

Setting the Project Type

Follow the steps below to set the project as *Scrum*/*Kanban*.

1. Log in with Project Administrator credentials.
2. Click 🏠 **Home**.
3. Click **Admin Tools**.
4. Click **Project Settings**.

5. Select one of the following options in the **Project Type** group:

`Scrum`	When selected, Kanban Boards will not show in **Planning** or **Team Room**.
`Kanban`	When selected, **Sprint List** and **Sprint Cards** will not show in **Team Room**. You can still create Sprints.
`Scrum & Kanban`	All Scrum/Kanban options display in **Planning** and **Team Room**.

6. Click **Save**.

Changing Task Status Values (Swimlanes)

Administrators can change the default Task Status Values from `Not Started`, `In Progress`, or `Complete`. You can add any values that your team needs. You must use a minimum of three. These values will be available for each Project on the server, but the ones selected will be available for the current project.

⚠️ **Important:** For Scrum Project types, each Task **Status** Value that is defined will be an individual column in the ⊞ **Team Room** > ▪▪ **Sprint Cards** view, so limiting the Values will help keep that view manageable.

💡 **Tip:** Watch how to do it: *Creating a New Task Status Value*

1. Log in with Project Administrator credentials.

2. Click 🏠 **Home**.

3. Click **Admin Tools**.

4. Click the **Project Settings** tab.

5. The available **Status Values** appear in the list in the **Task Status Management** group. The ones that are checked are the ones that are being used.

   a) Click **New Task Status** to add a new **Status** Value.
   b) Alternatively, you can double click any existing value to edit it.
   c) Click the corresponding check box to enable that **Status** Value.
   d) Drag-drop the items to place them in the order you prefer. This order represents how they will display in columns in the **Sprint Cards** view.

   The first checked item in the list will be indicated as the `Starting Task Value` while the last will state `Completed Task Value`.

Configuring Input Streams

Input Streams (🔼) enable business and quality teams to work in seamless cadence with the Agile development team. Using Micro Focus Connect, *Input Streams* allow you to pull in external Requirements and Defects into the Rhythm **Breakdown** view for you to review and discuss. If you then choose to *Accept* them, new Epics/Defects are created in Rhythm with clear traceability back to the original external asset. This link provides a mechanism for feedback to the original author.

Each Project in Rhythm can contain a single Input Stream for Requirements and a single Input Stream for Defects. After an administrator sets up the Input Stream for the Project, you actively **Accept** or **Reject** each item that you want to add to the Backlog.

⑆ Input Streams

☑ **Defects**

      Query: | All Items ▾ |

☑ **Requirements**

      Query: | Planned requirements ▾ |

✎ **Note:** Watch how to configure them: *Configuring Input Streams*.

Creating a Requirement Input Stream

For Requirements, you need to create a filter in Atlas that contains all of the conditions that you want. When you do this, a query is created in Rhythm that you can set for the project. For example, you could create an Atlas Filter with the criteria `Where Status = Planned`. This query would pull all Requirements that meet the Filter condition.

1. In Atlas, create a Filter that contains all of the criteria that you need. A resulting Query will be created in Rhythm that you can use in this step.

   💡 **Tip:** Watch how to do it: *Creating a Query in Atlas*.

2. Log in with Project Administrator credentials.
3. Click 🏠 **Home**.
4. Click **Admin Tools**.
5. Click the **Input Streams** tab.
6. Click the **Requirements** check box.
7. Click **Configure**. The **Create Input Stream** dialog box opens.
8. From the **Queries** list, select the query that you want to use.
9. Click **OK**.

Items that match the query criteria will be available for review and discussion in the 🗐 **Planning** > ⊞ **Breakdown** view.

Creating a Defect Input Stream

For Defects, an administrator can select any of the pre-existing queries that are defined in the Atlas Hub, or use default queries provided by Rhythm: `All Items`, `All Open`, **All Open and High Severity**, `All Open in the last year`, `Flagged Items`, `Not a Priority`, `Priority`, `Status = Closed`, `Deferred`, `Open`, `Resolved`, or `Verified`, `Type = Defect` or `Suggestion`, or `Unread Changes`.

To create a Defect Input Stream:

1. Log in with Project Administrator credentials.
2. Click 🏠 **Home**.
3. Click **Admin Tools**.

4. Click the **Input Streams** tab.

5. Click the **Defect** check box.

6. Click **Configure**. The **Create Input Stream** dialog box opens.

7. From the **Queries** list, select the query that you want to use for Defects.

8. Click **OK**.

Items that match the query criteria will be available for review and discussion in the ▤ **Planning** > ⊨ **Breakdown** view.

Accessing Micro Focus Connect

When you connect a third party tool to the Atlas Hub via Micro Focus Connect, you can sync data into Rhythm. For more information, access the community for documentation and supported connectors for Micro Focus Connect.

1. Click 🏠 **Home**.

2. Click **Admin Tools**.

3. Click the **Connect** tab. The **Connect** tab opens.

4. Click **Open**. Micro Focus Connect opens.

5. For additional information on Micro Focus Connect, refer to the Micro Focus Connect documentation: *http://documentation.microfocus.com/help/nav/5*

SCM Configuration

Rhythm's default Atlas Hub configuration automatically uses StarTeam as its SCM tool. This SCM configuration allows you to perform Code Reviews, pairing Rhythm Stories/Tasks with your StarTeam check-ins. If you are already using StarTeam, you can start creating and collaborating in Code Reviews immediately. Here's what you will do:

In StarTeam

- Continue your existing SCM practices (using StarTeam Views).
- As part of your StarTeam commit, select an existing Rhythm Story as a *Process Item*. This associates the Story with the specific commit.

In Rhythm

- Open the Story. Click **Actions** > **Create Review** and a Code Review icon will be under the Story.
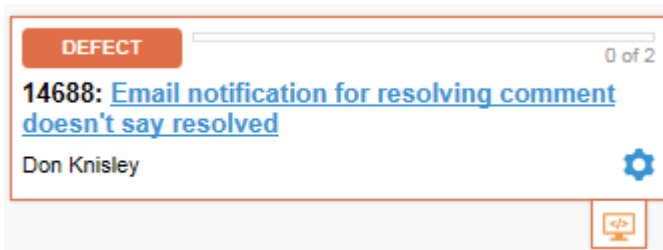
- *Invite team members to review your changes*.
- See Code Reviews that you are involved with as the *Author*, *Reviewer*, or *Watcher*.
- Mark up and *comment your reviews*, *automatically sending suggestions to the owner*.
- *Accept or reject the code reviews*.

⚠ **Important:** If you are using another supported SCM tool, such as Git, you will need to *create and configure a new SCM Configuration*.

Creating a New SCM Configuration

Rhythm is configured to use StarTeam as its SCM tool by default. If you use Git or a different, supported SCM system, you can configure it for your Rhythm project to enable Code Reviews.

1. Click **Home**.
2. Click **Admin Tools**.
3. Click **SCM Configuration**.
4. Click **New**. A new entry is added to the **SCM Configuration** list.
5. In the **Repository** group:
   a) Select the **Type**: Git.
   b) Create a unique **Name** for the SCM Configuration.
   c) Enter the full **URL** for the system. For example, all of the below would work:

   ```
   ssh://server/repositoryName.git
   http://server/repositoryName.git
   git://server/repositoryName.git
   ```

6. In the **Branches to Monitor** group, click **New** to add a new row for each branch to monitor. If no branches are added then all are included.
   a) In the **Pattern** field, use the * or a regular expression to include a branch.
   b) In the **Action** field, select Include or Exclude.
7. In the **Authentication** group, select **Basic** or **SSH**:

   For **Basic**:

   1. Enter the **Username**.
   2. Enter the **Password**.

   For **SSH**:

   1. Enter the **Private Key File Location** (on the server).

      ⚠️ **Important:** For this field, you must use the following location and file:

      `\cosmo\data\conf\ssh\<filename>.`

      `<filename>` must be named according to the following rules:

      1. It must start with and use any lower-case letter between a-z.
      2. It may contain numbers (after the initial lower-case letter).
      3. It must not have a file extension.
   2. Enter your **Private Key Pass Phrase**.

Planning

High performing agile teams organize and plan their work. The Rhythm **Planning** perspective enables teams to:

- Organize and *Breakdown* their *Stories*.
- Manage their *Backlog*.
- Define the *Timeboxes* in which *Releases* and *Sprints* are incrementally delivered.
- Create and administer a *project's Kanban Board(s)*.

Breakdown View

Click ▤ **Planning** > ▤ **Breakdown**.

The **Breakdown** view enables you to define and group Stories into a meaningful tree organization. The organization structure is user-defined, allowing the creation of a tree structure that matches each individual project's needs. You can create new items inline, and use drag-drop to move items within the hierarchy.

Work items are broken down into Stories (using parent-child relationships) that make up the actionable **Backlog**. This is often done when an individual Story is too large and cannot be completed within the boundaries of your preferred delivery window: Sprint, Release, or Kanban Board. These individual Stories provide incremental progress toward the larger defined need. That means:

- Stories in the **Breakdown** view that do not have children and are not already scheduled in a Sprint are fed to the **Backlog** for ranking and then delivery assignment.
- These Stories (Stories that are parents) will never exist in the **Backlog** and cannot be assigned to a Sprint. They are in the **Breakdown** view allowing you to organize the smaller, actionable Stories. However, these Stories are available for use in a Kanban Board.

See the following diagram:
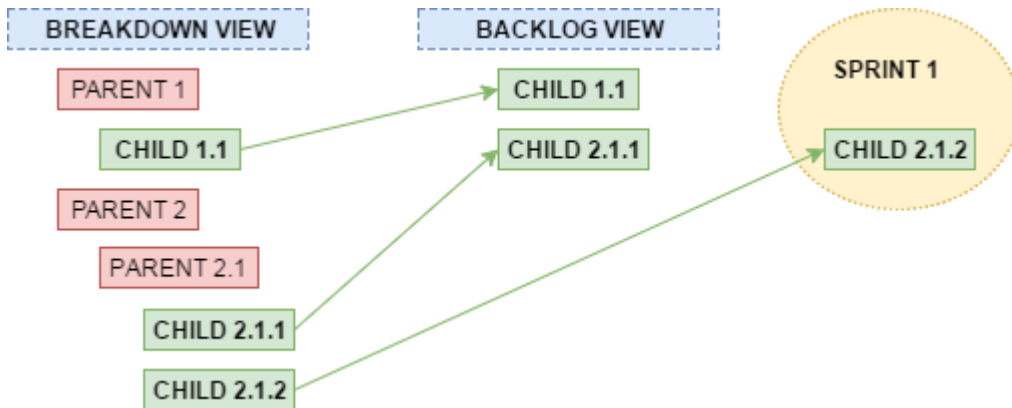
Breaking Down a Story

While the **Backlog** is used to store the ranked, actionable work queue, the **Breakdown** view allows you to define and group Stories into a meaningful hierarchy. You use the **Breakdown** action to do this.

1. Click ▤ **Planning** > ▤ **Breakdown**.
2. Click ⚙ > **Breakdown** to the left of the Story you want to break down. The **Create Story** box opens.
3. Select the **Type**.
4. Select the **Title**.
5. Click **Save & Close** if you are finished adding child Stories or click **Save & New** to add another child Story.

To edit the Stories further, see:

- *Editing a Story*
- *Adding Tasks to a Story*

Retrieving a Story URL

The following shows you how to retrieve a Story's URL so that you can link back to it from other Stories or third-party applications.

1. Click ⬛ **Planning** > ⊟ **Breakdown**.

2. Click the **ID** column to open the **Story Editor**.

3. Your browser's Address bar will contain the full URL that you can copy.

Input Streams

Input Streams (⥮) enable business and quality teams to work in seamless cadence with the Agile development team. Using Micro Focus Connect, *Input Streams* allow you to pull in external Requirements and Defects into the Rhythm **Breakdown** view for you to review and discuss. If you then choose to *Accept* them, new Epics/Defects are created in Rhythm with clear traceability back to the original external asset. This link provides a mechanism for feedback to the original author.

Each Project in Rhythm can contain a single Input Stream for Requirements and a single Input Stream for Defects. After an administrator sets up the Input Stream for the Project, you actively **Accept** or **Reject** each item that you want to add to the Backlog.

> 📝 **Note:** If you do not have privileges that are equal to or exceed those of *Viewer* in Atlas, the **Info Stream** column will not display realistic information for Stories that originate from Requirements due to the lack of permissions (to access the trace to the Requirement).

Requirements	For Requirements, you need to create a filter in Atlas that contains all of the conditions that you want. When you do this, a query is created in Rhythm that you can set for the project. For example, you could create an Atlas Filter with the criteria `Where Status = Planned`. This query would pull all Requirements that meet the Filter condition. The following columns are visible in the Requirements Input Stream grid: `ID`, `Name`, `Type`, `Hub Priority`, `Priority`, and `Status`. Only the `Name` and `Priority` properties map into a Story.
Defects	For Defects, an administrator can select any of the pre-existing queries that are defined in the Atlas Hub, or use default queries provided by Rhythm: `All Items`, `All Open`, All Open and High Severity, `All Open in the last year`, `Flagged Items`, `Not a Priority`, `Priority`, `Status = Closed`, `Deferred`, `Open`, `Resolved`, or `Verified`, `Type = Defect` or `Suggestion`, or `Unread Changes`. The following columns are visible in the Defects Input Stream grid: `ID`, `Name`, `Priority`, `Status`, and `Severity`. Only the `Name` field is used when imported into Rhythm.

Project administrators enable this functionality in **Admin Tools** and can switch the query at any time during the project. After they have been configured, click **Input Streams** on the ⬛ **Planning** > ⊟ **Breakdown** view to **Accept**/**Reject** individual items.

Stream States

Whether an item appears in an Input Stream depends on the Query as well as it's internal **Stream State**:

Pending	The default state for every item. When an item is `Pending` and matches the query, it is listed in a Stream.
Accepted	The item has been accepted and another artifact (Story) has been created
Rejected	The item has been rejected. No other artifact has been created from this one. It will not be listed in the Input Streams.

Although you don't modify these directly, you can see the values when you compare versions of the original Input Stream asset on which the Story is based. And, there is a way to move a `Rejected` item back to `Pending`. See *Rejecting an Item* in *Accepting Input Stream Items*.

Reviewing Input Stream Items

If your administrator configured Defect or Requirement Input Streams, access them in the 🖥 **Planning** > 📇 **Breakdown** view by clicking 〽 **Input Streams** in the menu bar. The **Input Streams** dialog box opens.

You can select either **Requirements** or **Defects**. When you do, all items matching the configured query will display in the list.

Your team should review each item and only add those to the Backlog that the team needs.

Discussing an Item

Before you **Accept** or **Reject** an item, you can click the **ID** column to open the external Defect or Requirement. Along with the item's normal properties, you can see the **Discussion** tab 💬 in the right hand pane. Use this to review and discuss with your team before you decide on the external items disposition. For more information, see *Discussions*.

Accepting Input Stream Items

As soon as you have reviewed and discussed the Defect or Requirement with the team, you will want to Accept the item into your Backlog. Do the following:

1. Open the **Input Streams** dialog box.
2. Select the **Defects** or **Requirements** option.
3. Click the **Gear** icon ⚙ for the item.
4. From the list, click **Accept**.

The following happens:

- The external item is removed from the **Input Streams** list.
- A new item (Defect or Requirement) is created in the Rhythm Backlog. This new item is a copy of the original, external item. It contains it's relevant properties like Title, Description, etc. However, any Discussion or Versions are not carried over since they apply only to the original external item.
- A Relationship is created from the new item in Rhythm to the external item. You can access the original external item via the **Relationships** tab in the **Story Editor**. Any change in the original, external item will change the Relationship status to `Suspect`.
- If you display the **Input Stream** column in your grids, you will see the 〽 icon. Hover to view the external item or click it to open the external item.

Rejecting an Item

If you choose to reject an item, click **Reject**, enter a reason in the **Reason for Rejection** dialog box, and a Discussion Topic is created for the external item. The item will no longer be visible in the **Input Streams** dialog box. In Atlas, you will see a Discussion Activity item added to the Home screen.

> 💡 **Tip:** If you inadvertently rejected an item or stakeholders have further defined your item such that it is now meaningful and should be in the Rhythm Backlog, you can modify that in Atlas. There is a **Reset Input Stream State** button for the Requirement in the **Define** view that will allow you to reset the item state so that it will be visible in the **Input Streams** dialog box again.

Import/Export

The Import/Export feature allows you to import Stories to and export Stories from the **Breakdown** view of the **Planning** Perspective. This feature supports both .CSV and .TXT files.

The feature allows you to:

- Create Stories in another tool and import them into Rhythm. For example, you may want to create Stories in Microsoft Excel.

- Export Stories into a file and consume that data with another tool.

This functionality is not designed for bulk Story editing since the import process creates new Stories for each row in the file.

Watch how to do it: *Importing/Exporting User Stories*.

About Import

- The Import feature will create new Stories even if a row `ID` in the file matches a Story ID in Rhythm.
- Requires a properly formatted .CSV/.TXT file.
- Import will accept many fields as defined in the Atlas Hub, but the heading names need to match exactly, otherwise, they will be ignored. Some common fields include `Name`, `StoryType`, `MOSCOW`, `EstimatedPoints`, `RichDescription`, `Sponsor`, `StoryStatus`, and `StoryValue`.

  The following represent the required values for list fields:

  - Atlas Hub `StoryType` = Rhythm `Type` and must be one of these values: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information about Types, see *Working with Stories*.
  - Atlas Hub `MOSCOW` = Rhythm `Priority` and must be one of these values: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.
  - Atlas Hub `StoryStatus` = Rhythm `Status` and must one of these values: `Not Started`, `In Progress`, `Complete`, or `Accepted`.
  - Atlas Hub `StoryValue` = Rhythm `Value` and must be `Low`, `Medium`, or `High`.
  - Atlas Hub `Sponsor` = Rhythm `Owner` and needs to be an existing User in Rhythm, otherwise it will be ignored.

    💡 **Tip:** Perform an export first, and use that file as your import file template.

- `StoryOrder` for the **Breakdown** view and `Rank` for the **Backlog** can be used.
- Additionally, the following fields can be used to create parent-child relationships: `ID` and `ParentID`.
- The `Name` field is the only required field.

About Export

- Export will adhere to any filter that is applied to the **Breakdown** view.
- Export will only export the columns that are visible. `Name`, `ID`, `ParentID`, `StoryOrder` and `Rank` are always exported and are listed first. Then, other selected columns are listed. `RichDescription` is at the end.
- Export will create a file with Story data for the following fields: `ID`, `ParentID`, `StoryType`, `MoSCoW`, `EstimatedPoints`, `StoryOrder`, `RichDescription`, `Sponsor`, `Blocked`, `Carried Over`, `CreatedTime`, `CreatedUserID`, `DotNotation`, `ModifiedTime`, `ModifiedUserID`, `Plan`, `Rank`, `Risk`, `Sprint`, `StoryStatus`, and `StoryValue`.

Importing Stories

It is important to understand the requirements before importing a file. See *Import/Export*.

1. Open the proper Project.
2. Click 🖥 **Planning** > 🗂 **Breakdown**.
3. Click 📥 Import.
4. Click **Select** on the **Import Stories** dialog box.
5. Select a .CSV/.TXT file and click **Open**.
6. Click **Import**.

Exporting Stories

1. Open the proper Project.
2. Click ▤ **Planning** > ⊞ **Breakdown**.
3. Apply any filter that you would like.
4. Show/hide columns that you want to export. See *Viewing Columns in Grids*.
5. Click ⬆ Export.
6. Depending on your browser, the file will be downloaded or you will be asked how you want to save the file.

Breakdown View Columns

Default Columns

The following columns display by default. If you modify them, your changes will be saved. For more information, see *Viewing Columns in Grids*.

ID — Unique ID of the item. Read-only.

Name — Story/Task/Item Name. In most cases, you can double click to edit. The Story and Task **Name** fields have a max length of 255 characters.

Type — Type is used as a way to categorize Stories. Available Story Types include: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information, see *Story Types*.

Priority — Priority is a list of values that includes the following: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.

Points — Indicates the number of Story Points it is estimated to take to complete the Story.

Owner — Indicates who is responsible for Story.

Status — The **Status** column for a Story contains a bubble control ▢ ▮ ▢ ▢ that allows you to click on one of the bubbles to set the status of the Story: `Not Started`, `In Progress`, `Complete`, or `Accepted`.

The **Status** column for a Task also contains a bubble control that allows you to set the Status by clicking on one of the bubbles. The number of bubbles depends on the amount of Task Status Values configured by your administrator. See *Change Task Status Values (Swimlanes)*.

Release — Currently assigned Release for the Sprint. Double click to enable the list for choosing a new one.

Sprint — The name of the current Sprint. Double-click to view a list of available Sprints. Select a new one if desired.

Tag — The **Tag** column allows you to select multiple pre-defined Tags for your Story. This is useful for filtering and organizing your Stories. Double-click the **Tag** cell for the proper row and select the Tags from the list to apply. For more information, see *Using Tags*.

> 🖉 **Note:** This column displays by default for new Users and Projects. For others, you need to explicitly show the column. See *Viewing Columns in Grids*

Available Columns

Other columns include:

Author — Name of the original author of the item. Read-only.

Blocked	Blocked indicates that there is an issue outside of the teams control preventing progress. If a Task is blocked, the parent Story will display ![BLOCKED] in the grid. Click the ⓘ icon in the grid to block a Task. After entering a comment, It will change to 🛑. Additionally, the Sprint Health & Progress area in the team Room will indicate that there are blocked Stories. For more information, see *Blocking a Task*.
Carried Over	This column in the grid will display a special image ![CARRIED OVER] representing that the Story was carried over from one sprint to the next (or more).
Created Date	Date that the item was created. Ready-only.
Code Reviews	The **Code Reviews** column will contain one of the various Code Review or Changes Set icons depending on whether there are active reviews or change sets related to the Story or Task. Click the icon to open the **Code Review** window (if a change set exists but a Code Review does not exist, you will be prompted to create one). A green icon indicates that all Code Reviews are accepted. Red indicates all reviews are rejected. Yellow indicates an indeterminate state where some are accepted/rejected or may not exist yet.
General	The **General** field is a text-based field that can be used to capture any business or development process data as needed by the project team. It is not used in Rhythm for any explicit purpose. It has a maximum length of 128 characters and is only available in grids that contain Story data.
Global ID	This is a unique identifier for all items in the Atlas Hub.
Input Stream	This column displays the **Input Stream** icon 🔀 if the item was derived from an Input Stream. Hover over the item for more information or click the column to open the external item. For more information, see *Input Streams*.
Kanban Status	If a Story is present in a Kanban Board lane, this column will display the board **Name** and lane in the following format: `<Board Name>: <Lane Name>`.
Last Edited	Date that the item was last edited. Ready-only.
Last Edited By	Name of the last author who last edited the item. Read-only.
Revision	Version number of the item. Read-only.
Rollup	The **Rollup** column displays the total planned points for all items contained within a parent item. It is available in the **Breakdown** view and **Sprints** Panel. Leaf items show the same value in both the **Points** and **Rollup** columns. Whereas the **Points** field is a manually entered value, the **Rollup** field is calculated. This allows you to compare the two values when actual (**Rollup**) may be greater than planned (**Points**).

Backlog View

Click 🗐 **Planning** > ☰ **Backlog**.

The **Backlog** view represents the ranked actionable work queue. This view is limited to schedule-able items (items without child Stories, and that are not already scheduled in a Sprint). Users come here to view/modify Project priorities, and to quickly add work items to a Sprint. The main goals of the **Backlog** view are to rank your Stories and then add them to Sprints.
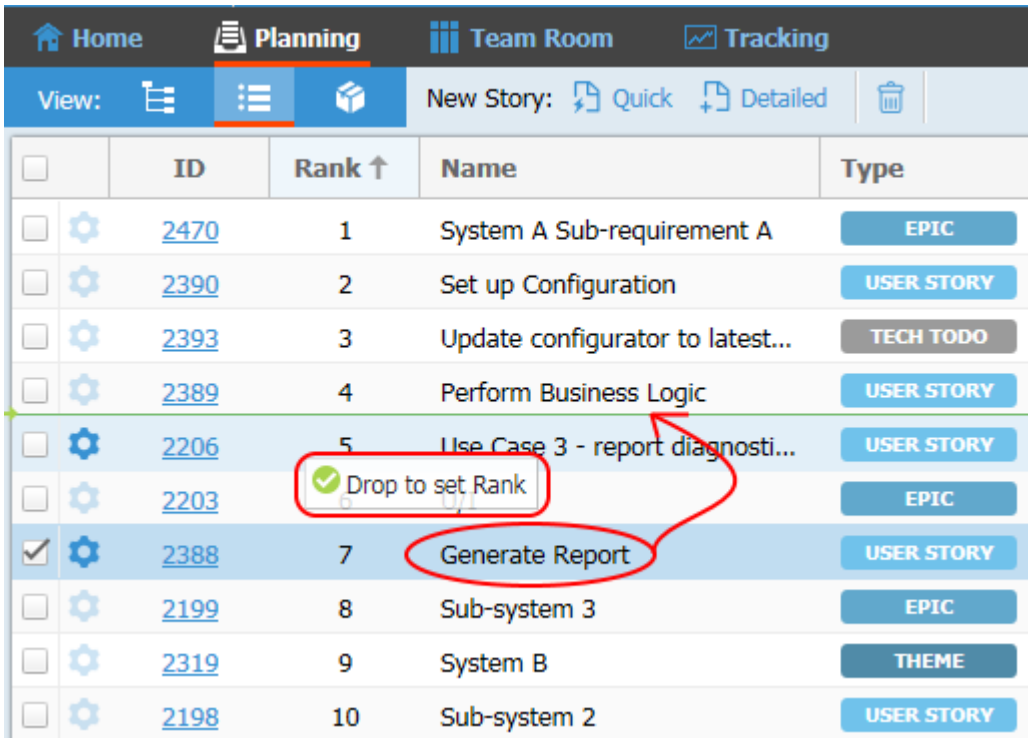
A Story's **Rank** is a numerical value used to represent it's position in the **Backlog**. Agile teams use Rank to identify relative priority and work sequence. When scheduling work (pulling items from the **Backlog** to a Sprint), Stories at the top of the **Backlog** are traditionally scheduled for work first. Story Rank can be modified via drag & drop and direct edit. When you drag-drop a Story in the **Backlog**, the Rank is automatically updated to reflect its new position/ranking. You can also double click the field and manually edit the value.

You can also: *Create a Story*, *Open a Story*, *Assign Stories to a Release*, *Rank Stories*, and *Copy Tasks from Other Stories*.

Ranking Stories

Follow the steps below to rank a Story. If you prefer, watch how to do it: *Ranking User Stories*.

1. Click ▤ **Planning** > ☰ **Backlog**.

2. Select a Story in the grid, click it and then drag and drop it into the position (rank) that you want.

When the Story is dropped, its **Rank** changes.

Sprints Panel

Click ▤ **Planning** > ☰ **Backlog** > 🏃 **Sprints Panel**. This opens the **Sprints** Panel.

You create and edit Sprints in the **Sprints** Panel in the **Backlog**. Sprints belong to a Project and can be assigned to a Release.

▶ Watch how to *Use the Sprints Panel*.

💡 **Tip:** To view a list of all of your Sprints, use ▤ **Planning** > 📦 **Timeboxes** and then select `Sprints` from the list. See *Sprints View*.

Some of the things you can do in the this panel include:

- *Drag-and-drop items from the Backlog into a Sprint.*
- *Create a Sprint from the Backlog.*
- *Change a Sprint's Capacity.*
- *Move a Story from One Sprint to Another.*
- *Go to the Team Room.*
- *Edit a Story.*
- *Edit the Sprint.*

Sprint Status
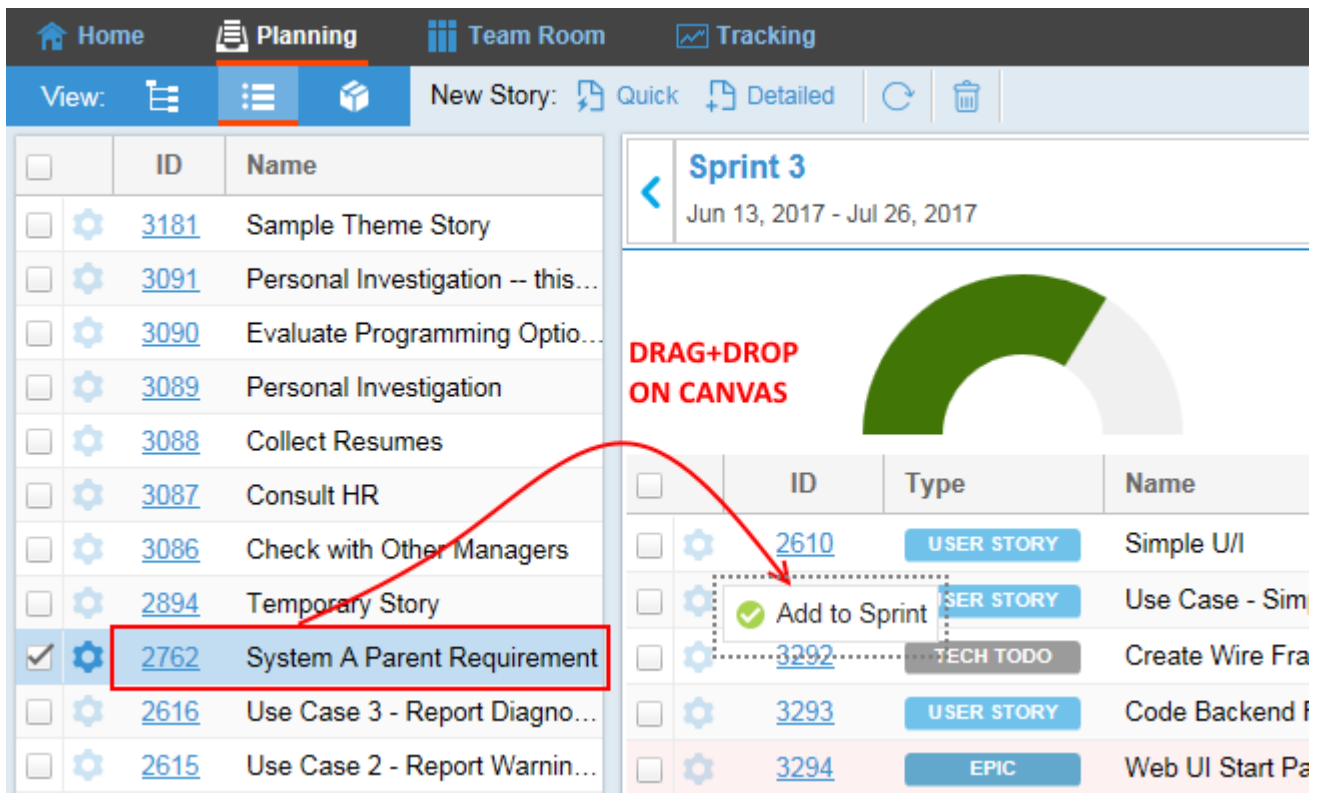
The top part of the **Sprints** Panel contains the dashboard that you can use to gauge a Sprint's health. You can review information about available/used points, items not estimated, items without owners, how many stories are true User Stories, and how many are Defects, etc.

Click **Edit Sprint** to open the **Edit Sprint** dialog box to set the available capacity for your sprint (The number of Story points that you expect to be able to complete in a Sprint.).

Adding a Story to a Sprint

Note: Parent Stories cannot be added to Sprints. Only the lowest level leaf item in your hierarchy can be added to a Sprint as a work item.

1. Click **Planning** > **Backlog**.
2. Expand the **Sprints** Panel .
3. Select a Sprint from the **Sprints** list.

4. Select a Story from the **Backlog** by clicking it and then drag-drop it into the Sprint in the **Sprints** Panel.

Creating a Sprint from the Backlog

1. Click **Planning** > **Backlog**.
2. Expand the **Sprints** Panel .
3. Click **New Sprint**. The **Create Sprint** dialog box appears.
4. Enter the **Sprint Name**.
5. Enter the **Capacity**.
6. Select the **Start Date** and **End Date**.
7. Select a **Release**.

8. Click **OK**.

You can also create a Sprint in the **Timeboxes** view. See *Creating a Sprint from Timebox View*.

Editing a Sprint

You can edit Sprints in three ways:

Click 🖥 **Planning >** ☰ **Backlog >** 🏃 **Sprints Panel.**	Click **Edit Sprint**.
Click 🖥 **Planning >** 📦 **Timeboxes**	Make sure `Sprints` are selected in the list. Click the **ID** next to the Sprint to edit.
Team Room	Click the **Edit Sprints** button on the far right.

Then, use the following steps to edit a Sprint:

1. The **Edit Sprint** dialog box appears.

2. From here, you can modify the **Name**, **Capacity**, **Dates**, and the assigned **Release**.

> 💡 **Tip:** Administrators can also modify all users' availability. See *Managing Team Capacity*.

3. Click **OK**.

Managing Team Capacity

Although each user can modify their current availability (see *Managing Your Capacity*), an administrator can also over-ride these Sprint and Project level values for each users.

When set at the Project level, the value will be used for each Sprint. When set at the Sprint level, it's only for that particular Sprint.

This value is visible as **Total** in the **Task Owner** field in the **Team Room** and **Story Editor**:
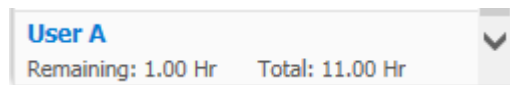
1. Open the **Edit Sprint** window (see *Editing a Sprint*).

2. In the **Team Availability** list, modify the hours available for each user for each Sprint/Project.

3. When complete, click **OK**.

Sprints Panel Columns

Default Columns

The following columns display by default. If you modify them, your changes will be saved. For more information, see *Viewing Columns in Grids*.

ID   Unique ID of the item. Read-only.

Type   Type is used as a way to categorize Stories. Available Story Types include: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information, see *Story Types*.

Name   Story/Task/Item Name. In most cases, you can double click to edit. The Story and Task **Name** fields have a max length of 255 characters.

Points   Indicates the number of Story Points it is estimated to take to complete the Story.

Priority Priority is a list of values that includes the following: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.

Status The **Status** column for a Story contains a bubble control ▢▢▢▢ that allows you to click on one of the bubbles to set the status of the Story: `Not Started`, `In Progress`, `Complete`, or `Accepted`.

The **Status** column for a Task also contains a bubble control that allows you to set the Status by clicking on one of the bubbles. The number of bubbles depends on the amount of Task Status Values configured by your administrator. See *Change Task Status Values (Swimlanes)*.

Owner Indicates who is responsible for Story.

Tag The **Tag** column allows you to select multiple pre-defined Tags for your Story. This is useful for filtering and organizing your Stories. Double-click the **Tag** cell for the proper row and select the Tags from the list to apply. For more information, see *Using Tags*.

> **Note:** This column displays by default for new Users and Projects. For others, you need to explicitly show the column. See *Viewing Columns in Grids*

Available Columns

Author Name of the original author of the item. Read-only.

Blocked Blocked indicates that there is an issue outside of the teams control preventing progress. If a Task is blocked, the parent Story will display **BLOCKED** in the grid. Click the ⓘ icon in the grid to block a Task. After entering a comment, It will change to 🛇. Additionally, the Sprint Health & Progress area in the team Room will indicate that there are blocked Stories. For more information, see *Blocking a Task*.

Carried Over This column in the grid will display a special image **CARRIED OVER** representing that the Story was carried over from one sprint to the next (or more).

Created Date Date that the item was created. Ready-only.

Code Reviews The **Code Reviews** column will contain one of the various Code Review or Changes Set icons depending on whether there are active reviews or change sets related to the Story or Task. Click the icon to open the **Code Review** window (if a change set exists but a Code Review does not exist, you will be prompted to create one). A green icon indicates that all Code Reviews are accepted. Red indicates all reviews are rejected. Yellow indicates an indeterminate state where some are accepted/rejected or may not exist yet.

Global ID This is a unique identifier for all items in the Atlas Hub.

Input Stream This column displays the **Input Stream** icon 〰 if the item was derived from an Input Stream. Hover over the item for more information or click the column to open the external item. For more information, see *Input Streams*.

Kanban Status If a Story is present in a Kanban Board lane, this column will display the board **Name** and lane in the following format: `<Board Name>: <Lane Name>`.

Last Edited Date that the item was last edited. Ready-only.

Last Edited By Name of the last author who last edited the item. Read-only.

Release This is the name of any of the Releases defined in the Project. You can normally double-click to edit and then select any of the available Releases.

Revision Version number of the item. Read-only.

Rollup The **Rollup** column displays the total planned points for all items contained within a parent item. It is available in the **Breakdown** view and **Sprints** Panel. Leaf items show the same value in both the **Points** and **Rollup** columns. Whereas the **Points** field is a manually entered value, the **Rollup** field is calculated. This allows you to compare the two values when actual (**Rollup**) may be greater than planned (**Points**).

Columns in the Backlog

Default Columns

The following columns display by default. If you modify them, your changes will be saved. For more information, see *Viewing Columns in Grids*.

ID Unique ID of the item. Read-only.

Rank A Story's **Rank** is a numerical value used to represent it's position in the **Backlog**. Agile teams use Rank to identify relative priority and work sequence. When scheduling work (pulling items from the **Backlog** to a Sprint), Stories at the top of the **Backlog** are traditionally scheduled for work first. Story Rank can be modified via drag & drop and direct edit. When you drag-drop a Story in the **Backlog**, the Rank is automatically updated to reflect its new position/ranking. You can also double click the field and manually edit the value.

Name Story/Task/Item Name. In most cases, you can double click to edit. The Story and Task **Name** fields have a max length of 255 characters.

Tag The **Tag** column allows you to select multiple pre-defined Tags for your Story. This is useful for filtering and organizing your Stories. Double-click the **Tag** cell for the proper row and select the Tags from the list to apply. For more information, see *Using Tags*.

> **Note:** This column displays by default for new Users and Projects. For others, you need to explicitly show the column. See *Viewing Columns in Grids*

Type Type is used as a way to categorize Stories. Available Story Types include: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information, see *Story Types*.

Priority Priority is a list of values that includes the following: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.

Points Indicates the number of Story Points it is estimated to take to complete the Story.

Owner Indicates who is responsible for Story.

Release This is the name of any of the Releases defined in the Project. You can normally double-click to edit and then select any of the available Releases.

Other Available Columns

Author Name of the original author of the item. Read-only.

Blocked Blocked indicates that there is an issue outside of the teams control preventing progress. If a Task is blocked, the parent Story will display **BLOCKED** in the grid. Click the ⓘ icon in the grid to block a Task. After entering a comment, It will change to ⊘. Additionally, the Sprint Health & Progress area in the team Room will indicate that there are blocked Stories. For more information, see *Blocking a Task*.

Carried Over This column in the grid will display a special image **CARRIED OVER** representing that the Story was carried over from one sprint to the next (or more).

Created Date Date that the item was created. Ready-only.

Code Reviews	The **Code Reviews** column will contain one of the various Code Review or Changes Set icons depending on whether there are active reviews or change sets related to the Story or Task. Click the icon to open the **Code Review** window (if a change set exists but a Code Review does not exist, you will be prompted to create one). A green icon indicates that all Code Reviews are accepted. Red indicates all reviews are rejected. Yellow indicates an indeterminate state where some are accepted/rejected or may not exist yet.
General	The **General** field is a text-based field that can be used to capture any business or development process data as needed by the project team. It is not used in Rhythm for any explicit purpose. It has a maximum length of 128 characters and is only available in grids that contain Story data.
Global ID	This is a unique identifier for all items in the Atlas Hub.
Input Stream	This column displays the **Input Stream** icon 🔃 if the item was derived from an Input Stream. Hover over the item for more information or click the column to open the external item. For more information, see *Input Streams*.
Kanban Status	If a Story is present in a Kanban Board lane, this column will display the board **Name** and lane in the following format: `<Board Name>: <Lane Name>`.
Last Edited	Date that the item was last edited. Ready-only.
Last Edited By	Name of the last author who last edited the item. Read-only.
Revision	Version number of the item. Read-only.
Status	The **Status** column for a Story contains a bubble control 🟩🟩⬜⬜ that allows you to click on one of the bubbles to set the status of the Story: `Not Started`, `In Progress`, `Complete`, or `Accepted`.

The **Status** column for a Task also contains a bubble control that allows you to set the Status by clicking on one of the bubbles. The number of bubbles depends on the amount of Task Status Values configured by your administrator. See *Change Task Status Values (Swimlanes)*. |

Timeboxes View

Click 📋 **Planning** > 📦 **Timeboxes**

Agile projects require both Sprints and Releases to incrementally delivery value. The **Timeboxes** view allows the creation and management of both. Users can view existing, modify, and fully manage Sprints and Releases.

Sprints Grid

Click 📋 **Planning** > 📦 **Timeboxes** and then select `Sprints` from the list.

The **Sprints** grid shows you a list of all of the Sprints in your project.

▶ Watch how to *Create a Sprint*.

Creating a Sprint from Timeboxes View

The following describes how to add a Sprint while you are in the **Timeboxes** view looking at all of your Sprints. You can also add a Sprint when you are in the **Backlog**. See *Creating a Sprint from the Backlog*.

1. Click 📋 **Planning** > 📦 **Timeboxes**
2. Select `Sprints` from the list.

3. Click ⊕ **New**.

4. Enter the **Sprint Name**.

5. Enter the **Capacity**.

6. Select the **Start Date** and **End Date**.

7. Select a **Release**.

8. Click **OK**.

Columns in Sprints View

The following columns are available for viewing in the grid on this page. Show/hide columns: *Viewing Columns in Grids*.

ID	Unique ID of the item. Read-only.
Name	Story/Task/Item Name. In most cases, you can double click to edit. The Story and Task **Name** fields have a max length of 255 characters.
Start Date	Planned Start Date of the Sprint.
End Date	Planned End Date of the Sprint.
Stories	Total number of Stories in the Sprint or Release.
Assigned	This value is the total number of Story points of all Stories assigned to the Sprint.
Delivered	This value is the total number of Story points of all Stories that are in the `Accepted` or `Complete` states.
Release	Currently assigned Release for the Sprint. Double click to enable the list for choosing a new one.
Capacity	The number of Story points that you expect to be able to complete in a Sprint.
Author	Name of the original author of the item. Read-only.
Created Date	Date that the item was created. Ready-only.
Global ID	This is a unique identifier for all items in the Atlas Hub.
Last Edited	Date that the item was last edited. Ready-only.
Last Edited By	Name of the last author who last edited the item. Read-only.
Revision	Version number of the item. Read-only.

Releases Grid

Click 🖥 **Planning** > 📦 **Timeboxes** and then select `Releases` from the list.

The Releases grid shows you a list of all of the Releases in your project.

▶ Watch how to *Create a Release*.

Creating a Release

Use the **Timeboxes** view of the **Planning** perspective to create/edit a Release.

💡 **Tip:** To edit an existing Release, click the number in the **ID** column of the grid to open the **Edit Release** box.

1. Click 🖥 **Planning** > 📦 **Timeboxes**.

2. Select `Releases` from the list in the toolbar.

3. Click ⊕ **New**. The **Create Release** dialog box appears.

4. Edit the **Release Name**.

5. Edit the **Weekly Capacity** (in Story Points).

6. Select the **Start Date**.

7. Select the **End Date**.

8. Click **OK**.

Changing the Weekly Capacity of a Release

This release-level value is the weekly capacity available for each Sprint (in Story Points). Do the following to change the default weekly capacity of a release.

1. Click 📋 **Planning** > 📦 **Timeboxes**.

2. Select `Releases` from the list in the toolbar.

3. Click the number in the **ID** column of the grid to open the **Edit Release** box. The **Edit Release** box appears.

4. Edit the **Weekly Capacity** (in Story Points).

5. Click **OK**.

Columns in Releases Grid

The following columns are available for viewing in the grid on this page. Show/hide columns: *Viewing Columns in Grids*.

ID	Unique ID of the item. Read-only.
Name	Story/Task/Item Name. In most cases, you can double click to edit. The Story and Task **Name** fields have a max length of 255 characters.
Start Date	Planned Start Date of the Sprint.
End Date	Planned End Date of the Sprint.
Stories	Total number of Stories in the Sprint or Release.
Weekly Capacity	This release-level value is the weekly capacity available for each Sprint (in Story Points). You can modify this on the **Edit Release** dialog box. Each new Sprint that is created will use this as the **Capacity** value.
Total Capacity	This is the Release's total capacity (in Story Points). It is calculated by **Weekly Capacity** * Weeks in Release (**Start Date** to **End Date**). This is a read-only value.
Author	Name of the original author of the item. Read-only.
Created Date	Date that the item was created. Ready-only.
Last Edited	Date that the item was last edited. Ready-only.
Last Edited By	Name of the last author who last edited the item. Read-only.
Revision	Version number of the item. Read-only.
Global ID	This is a unique identifier for all items in the Atlas Hub.

Kanban Boards View

Click 📋 **Planning** > 🗔 **Kanban Boards**.

This view allows Project or Server Administrators to manage all of the boards in a Project. You can create, delete, edit and sort all boards.

Each board displays in a row with the gear icon ⚙ allowing you to edit the board.

Once you create your board, follow the steps in *this sequence of topics* to help you to configure your board and to *make it available in the Team Room*.

Creating a Board

1. Click **Planning** > **Kanban Boards**

2. Click **New Board**.

3. Enter the **Name** of the board.

> **Note:** Names cannot be duplicates or empty.

4. Click **OK**.

Kanban Board Administration

For Project and Server Administrators, you can access the **Kanban Board Administrator** using one of the following methods:

1. From the **Planning** perspective:

   a. Click **Planning** > **Kanban Boards**.

   b. Click > **Administrator** on the board you want to edit.

2. From the **Team Room**:

   a. Click **Team Room** > **<Kanban Board Name>**.

   b. Click the **Administrator** button in the top right corner.

The **Kanban Board Administrator** window is comprised of four sections allowing you to administer different Kanban features.

Name

The top section of the panel contains the **Name** field. Click in the field to rename the board.

Board Backlog Source

Contains options from where you want to pull your Stories. See *Getting Stories Into Your Board*.

Other

Contains options for your board's availability in the **Team Room** and how your Board handles **WIP Limit** and **Age**. See *Setting Kanban Board Options*.

Swimlanes

Allows you to customize your lanes. See *Managing Swimlanes*.

Getting Stories Into Your Board

Use the steps below to determine how Stories are added to your board. No matter which option you choose, all Stories that match the criteria that you select are initially displayed in the **Board Backlog** lane.

1. Click ![] **Planning** > ![] **Kanban Boards**.

2. Click ![] > **Administrator** for the board to configure. The **Kanban Board Administrator** window appears.

3. In the **Board Backlog Source** group, select one of the following:

Sprint	Select this option to use your Sprints as the source for your board's Stories. Typically only used if you are moving from Scrum to Kanban or if you plan to use both together.
Release	Select this option to use Releases as the source for your board's Stories. This option allows you to define a target payload for a specific release and use a board to manage its delivery. Using this option allows you to use Rhythm's **Tracking** perspective.
Custom Filter	Select this option to use a pre-defined custom filter as the source for your Board's Stories. This option provides the ultimate flexibility. You can include anything that you can filter on, including Tags. For more information, see *Working with Filters*.

> **Note:** The Custom Filter must be a Project-level filter and cannot be changed to public or shared.

4. Click **OK**.

Setting Kanban Board Options

Administrators can use the steps below to add a board to the **Team Room** menu, to enforce WIP, and to show Age Indicators in cards.

1. Click ![] **Planning** > ![] **Kanban Boards**.

2. Click ![] > **Administrator** for the board to configure. The **Kanban Board Administrator** window appears.

3. In the **Other** group:

   a) Click **Make available in Team Room** to add the board as a **Team Room** menu item.

   b) Check **Enforce WIP** to prevent Stories from being added to lanes that have reached their **WIP Limit**.

   c) Check **Show Age Indicators** to highlight stories that have been in their current lane for too long. An indicator similar to the following displays in the card: ![4 days].

4. Click **OK**.

Managing Swimlanes

Use the steps below to manage your board's Swimlanes. The default Swimlanes include `In Progress` and `Done`. An example of a full list of lanes would be the following:

	Name	WIP Limit	Age Indicator	Story Status Mapping
✓	Define	∞	1 day	Not Started
✓	Dev in Progress	6	4 days	In Progress
✓	QA in Progress	3	2 days	In Progress
✓	Functional Review	∞	No	Completed
✓	Done	∞	No	Accepted

> **Note:** There is also an initial lane named **Board Backlog**. Even though you cannot configure it or see it in the list above, all boards have this lane. The **Board Backlog** lane is the initial lane where all

Stories begin, at least those that meet the **Board Backlog Source** option (you can also manually add Stories to a board which may or may not meet this criteria). To configure which Stories go there, see the *Board Backlog Source option*.

The goal of the **Swimlanes** list is to add the proper lanes to meet your organization's needs. So, within the list you will:

- Click **New** to create a lane.
- Double click the **Name** column to rename a lane.
- `Drag+drop` lanes to arrange the order.
- Click **Delete** 🗑 to delete a lane.
- Define the **WIP limit** (how many Stories can be in the lane).
- Set the **Age Indicator** (how long they can be in the lane before they are flagged).
- Map the lane to an existing Rhythm status in the **Story Status Mapping** column.
- Uncheck a lane if you don't want to use it but want to save for later.

To add a new lane:

1. Click **New** and then double click to enter the lane **Name**.
2. Double click the **WIP Limit** cell. Enter the number of Stories that can be in the lane at one time. The default value is the infinity symbol which means no limit.
3. Double click the **Age Indicator** cell. Enter the numbers of days at which an item will be flagged if the limit is reached.
4. Double click the **Story Status Mapping** cell. Select a value that the **Status** field will change to when a Story enters the lane.

Do Not Change	The **Status** value does not change. A Story will retain its previous Status when entering this lane.
Not Started	**Status** is changed to `Not Started`.
In Progress	**Status** is changed to `In Progress`.
Completed	**Status** is changed to `Completed`.
Accepted	**Status** is changed to `Accepted`.
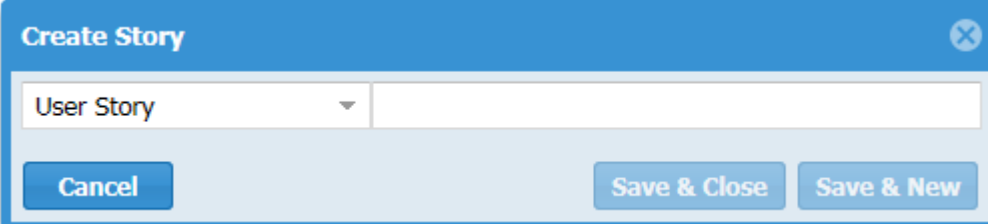
5. Click **OK**.

Creating a Story

You can create *Quick* or *Detailed* Stories in the **Planning** > **Breakdown** and **Planning** > **Backlog** views.

Use *Quick Story* when you are adding lots of high level Stories.

In each perspective, look for **New Story**: **Quick** | **Detailed**.

For more information, see *Using the Story Editor*.

To create a detailed Story:

1. Click ![icon] **New Story: Detailed**.

Create Backlog Item

Not created yet.

Owner: Not Assigned ▾

Type: User Story ▾

Points: 0 ▲▼

Status: ▢ ▢ ▢ ▢

Priority: Could Have ▾

Sprint: Not Assigned ▾

Release: Not Assigned ▾

Parent: No Parent ✎ ⊖

Title...

B *I* U S x₂ x² I̲ₓ | ≟ ∷ | ≣ ≣ | ≣

Format ▾ | Font ▾ | Size ▾ | **A**▾ **A**

✂ ▢ ▢ ▢ ▢ | ↩ ↪ | 🔍 ᵇₐ | 📄 | 🖼 | ⊞

🔗 🔗

Save   **Cancel**

2. Enter the details for the Story:

Title	Enter a descriptive name for the Story.
Owner	Select the resource responsible for the Story.
Type	Select the Type of Story: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information about Types, see *Working with Stories*.
Points	Enter the estimated number of Story Points required to complete the Story.
Status	Click one of the boxes indicating the status: `Not Started`, `In Progress`, `Complete`, or `Accepted`.
Priority	Select a Priority from the list: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.
Sprint	Select the Sprint for the Story, if needed.
Release	Select the Release for the Story, if needed.
Parent	The Parent Story is displayed. Click the link to display the Parent or click the **Set Parent** icon to change the parent.
Description	Add or edit the Story description.

3. Click **Save**.

Editing a Story

Use the steps below when you want to add more detail to a base Story or to begin adding tasks to a Story. You can edit a Story in the ⬛ **Planning** > ▦ **Breakdown** and ⬛ **Planning** > ☰ **Backlog** views.

For more information, see *Using the Story Editor*.

1. Click ⚙ > **Open** to the left of a Story.

> 💡 **Tip:** If you display the **ID** column, you can simply click the number in that column to edit the Story.

The **Edit Story** dialog box opens.

2. Enter the details for the Story:

Title	Enter a descriptive name for the Story.
Owner	Select the resource responsible for the Story.

Type	Select the Type of Story: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information about Types, see *Working with Stories*.
Points	Enter the estimated number of Story Points required to complete the Story.
Status	Click one of the boxes indicating the status: `Not Started`, `In Progress`, `Complete`, or `Accepted`.
Priority	Select a Priority from the list: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.
Sprint	Select the Sprint for the Story, if needed.
Release	Select the Release for the Story, if needed.
Parent	The Parent Story is displayed. Click the link to display the Parent or click the **Set Parent** icon to change the parent.
Description	Add or edit the Story description.

3. To add Tasks to the Story, see *Adding Tasks to a Story*.

Targeting Stories for a Release

During release planning, product owners and managers review the Backlog and choose which Stories need to be included in the Release. The progress data for the Stories will be used to calculate the **Release Progress** and **Release Burnup** charts in the **Tracking** Perspective.

1. From the **Planning** perspective, select the Story you want to target for a Release.
2. Click ⚙ > **Assign to Release**. The **Assign to Release** dialog box opens.
3. Select a Release from the list and click **OK**.

💡 **Tip:** To change multiple rows simultaneously:

1. Show the **Release** column in the grid.
2. Select multiple rows.
3. Double click any **Release** cell and select the new value.
4. Click **Save**.

Quick Find

When you want to quickly search for a string in the **Planning** perspective and don't need to create a filter:

1. Click either 🗒 **Planning** > 🗂 **Breakdown** or 🗒 **Planning** > ☰ **Backlog**.
2. Put your cursor in the **Quick Filter** field.
3. Type the text to search for and click `Enter`. Your results show in the grid. Click **Clear Filter** to remove the results.

Working with Filters

Filtering your Backlog simplifies the process of locating Stories that match a certain criteria.

▶ Watch how to do it: *Creating and Applying Filters*.

✏ **Note:** If a Kanban **Board Backlog Source** is set to a filter, you cannot share or delete that filter.

Creating a Filter

Use the **Filters** dialog box to create simple or complex multi-field, multi-level filters.

Note: You also use this dialog box to duplicate existing filters or delete filters.

1. Open the **Breakdown** or **Backlog** views in the **Planning** Perspective.
2. Click ▼ (**Filter**). The **Filter** dialog box appears.
3. Click 🗂 (**New Filter**). The editable filter field appears in the right panel.
4. Type a name for the filter.
5. Select **Project** if the filter is for the project only or select **Shared** if the filter is for all projects.
6. In the initial **Where** clause group, select criteria from the **Criteria** list. The appropriate field for the specified criteria appears.
7. Select an operator from the **Conditions** list (for example., `begins with`, `>=`, `>`, etc.).
8. Select or type the criteria for the operator in the last field. This field changes based on the criteria selected from the **Criteria** list.
9. Optional: Click **Add Query Rule** ⊕ to add a new row to the current **Where** clause group.
10. Optional: Click **[Add Group]** to add another level to the filter. **And/Or** options appear.

    You can add additional groups by repeating these steps.
11. Optional: Click **And** to change it to **Or**.
12. Click **Save**. The filter appears in the filter list.

Applying a Filter

To apply an existing filter:

1. Open the **Breakdown** or **Backlog** views in the **Planning** Perspective.
2. In the toolbar, select a filter from the **Filter** list. The items that match the filter criteria display below.

Bulk Editing in a Grid

You can perform bulk edits on multiple rows for any visible column in the grid by:

1. Click the check box in the far left column to toggle selection of a row. Or, click the body of any row and that row will be selected (and others de-selected).
2. `Ctrl+click` to select another row (or click another check box).
3. Double-click any cell in the row and change the value.
4. Click Save.

Team Room

The **Team Room** provides a team-centric view for recording/reviewing Sprint activities. The design is focused on enabling recurring team activities, both for updating work status and running the daily standup. You have several options on how you want to work with your Stories/Tasks including using Scrum or Kanban:

Scrum
- 🗒 *Sprint List*
- ▦ *Sprint Cards*

Kanban
▥ *Kanban Boards*

Scrum

Use the topics in this section if you use Scrum (as opposed to *Kanban*).

Sprint List

The **Team Room Sprint List** ⊞ contains a list of the Stories contained in the selected Sprint. You can expand them to see the Tasks. You can also double click for in-line editing.

💡 **Tip:** Use the browser **Address** field to copy the link to the currently selected Sprint, Story, or Task.

Information Pane

Sprint List	Select a Sprint from the list.
Remaining Days	Shows how many days remaining in the Sprint.
Remaining Hours	Shows how many hours of actual hours work left compared with planned.
Accepted	Provides the percentage of the number of accepted stories that are accepted.
Operating Capacity	Point-based metric comparing how many points are being completed compared to planned.
Edit Sprint	Click to Edit Sprint.
Code Review	Click to view the Code Review Summary window which provides metrics on code reviews in the Sprint. For more information, see *Code Review*.
Sprint Activity	Click to view the Sprint Activity window which provides a date-based listing of all changes to your Stories.
Discussions	In addition to each Story supporting Discussions, each Sprint supports Discussions. Click the **Discussion** icon 💬 to open the **Sprint Discussions** box. For more information, see *Discussions*.

Team Overview

The **Team Overview** pane provides graphs and charts and informative data about your sprint. The pane can be expanded or collapsed using the arrow to the right ⌄.

💡 **Tip:** When you apply a filter in the **Team Room**, the data in the **Team Overview** represents only that of the applied filter.

Burndown	The **Burndown** chart is a run-chart of outstanding work that shows the total number of hours on the vertical access with number of days on the horizontal access. This chart is used to help predict when the work will finish.

  1. The horizontal dashed line at the top represents the team capacity.
  2. The solid line represents the ideal Burndown rate.
  3. The blue line with boxes represents the actual hours remaining and should be trending down.
  4. The solid orange line (Burnup) indicates the time recorded by the team and should be trending up, ideally an inverse of the boxed line.

Cumulative Flow

The **Team Room Cumulative Flow** chart represents the amount of work in the Sprint in different states for each day of the Sprint. The vertical axis represents the points in the Sprint. The horizontal axis represents each day of the sprint. It includes **Points Not Started** (light green), **Points In Progress** (green), **Points Completed** (dark green), and **Points Accepted** (blue). The height of the chart defines the scope. If something is added or the estimate is increased, then the whole chart gets taller. If something is removed or the estimate decreased, then the chart gets smaller. It provides the ability to see changes in scope in the sprint.

Points by Type

The **Points by Type** graph provides a pie chart of the distribution of percentage of work based on the types of Stories. Click one of the types to apply a Quick Filter to see just those items.

Health & Progress

The information in this list contains data points of the Sprint of which you need to be aware. There are positive indicators (✔) and trouble indicators (⚠):

- Number of blocked Stories.
- Stories not estimated.
- Stories without Tasks.
- Tasks not estimated.

- Tasks with actual hours greater than estimated hours.

Click any of the items in the list with a ⚠ to apply a filter to only show those items.

When a filter is applied, you can click the clear icon (❌) to clear any filter.

Sprint Cards

The **Team Room Sprint Cards** view ▦ shows the Sprint's Stories and Tasks arranged in the default status columns: **Not Started**, **In Progress**, **Complete**. Drag and drop each story's card to change the status. Expand a Story to see and status the Tasks using the arrow on the right ⌄. You can *change the available status columns*.

💡 **Tip:** Use the browser **Address** field to copy the link to the currently selected Sprint, Story, or Task.

Information Pane

Sprint List	Select a Sprint from the list.
Remaining Days	Shows how many days remaining in the Sprint.
Remaining Hours	Shows how many hours of actual hours work left compared with planned.
Accepted	Provides the percentage of the number of accepted stories that are accepted.
Operating Capacity	Point-based metric comparing how many points are being completed compared to planned.
Edit Sprint	Click to Edit Sprint.
Code Review	Click to view the Code Review Summary window which provides metrics on code reviews in the Sprint. For more information, see *Code Review*.
Sprint Activity	Click to view the Sprint Activity window which provides a date-based listing of all changes to your Stories.
Discussions	In addition to each Story supporting Discussions, each Sprint supports Discussions. Click the **Discussion** icon 💬 to open the **Sprint Discussions** box. For more information, see *Discussions*.

Team Overview

The **Team Overview** pane provides graphs and charts and informative data about your sprint. The pane can be expanded or collapsed using the arrow to the right ⌄.

💡 **Tip:** When you apply a filter in the **Team Room**, the data in the **Team Overview** represents only that of the applied filter.

Burndown	The **Burndown** chart is a run-chart of outstanding work that shows the total number of hours on the vertical access with number of days on the horizontal access. This chart is used to help predict when the work will finish.

1. The horizontal dashed line at the top represents the team capacity.
2. The solid line represents the ideal Burndown rate.
3. The blue line with boxes represents the actual hours remaining and should be trending down.
4. The solid orange line (Burnup) indicates the time recorded by the team and should be trending up, ideally an inverse of the boxed line.

Cumulative Flow

The **Team Room Cumulative Flow** chart represents the amount of work in the Sprint in different states for each day of the Sprint. The vertical axis represents the points in the Sprint. The horizontal axis represents each day of the sprint. It includes **Points Not Started** (light green), **Points In Progress** (green), **Points Completed** (dark green), and **Points Accepted** (blue). The height of the chart defines the scope. If something is added or the estimate is increased, then the whole chart gets taller. If something is removed or the estimate decreased, then the chart gets smaller. It provides the ability to see changes in scope in the sprint.

Points by Type

The **Points by Type** graph provides a pie chart of the distribution of percentage of work based on the types of Stories. Click one of the types to apply a Quick Filter to see just those items.

Health & Progress

The information in this list contains data points of the Sprint of which you need to be aware. There are positive indicators (✔) and trouble indicators (⚠):

- Number of blocked Stories.
- Stories not estimated.
- Stories without Tasks.
- Tasks not estimated.

- Tasks with actual hours greater than estimated hours.

Click any of the items in the list with a ⚠ to apply a filter to only show those items.

When a filter is applied, you can click the clear icon (⊗) to clear any filter.

Creating a Story

You can create *Quick* or *Detailed* Stories in the ⊞ **Team Room** > ☰ **Sprint List** and ⊞ **Team Room** > ⊞ **Sprint Cards** views.

Use *Quick Story* when you are adding lots of high level Stories.

In each perspective, look for **New Story**: **Quick** | **Detailed**.

For more information, see *Working with Stories*.

💡 **Tip:** Watch how to do it: *Creating a Story*.

To create a detailed story:

1. Click ➕🗋 **New Story: Detailed**.

2. Enter the details for the Story:

Title	Enter a descriptive name for the Story.
Owner	Select the resource responsible for the Story.
Type	Select the Type of Story: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information about Types, see *Working with Stories*.
Points	Enter the estimated number of Story Points required to complete the Story.
Status	Click one of the boxes indicating the status: `Not Started`, `In Progress`, `Complete`, or `Accepted`.
Priority	Select a Priority from the list: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.
Sprint	Select the Sprint for the Story, if needed.
Release	Select the Release for the Story, if needed.
Parent	The Parent Story is displayed. Click the link to display the Parent or click the **Set Parent** icon to change the parent.
Description	Add or edit the Story description.

3. Click **Save**.

Editing a Story

Use the steps below when you want to add more detail to a *Quick Story* or to begin adding tasks to a Story.

You can edit a Story in the ▦ **Team Room** > ▤ **Sprint List** and ▦ **Team Room** > ▩ **Sprint Cards** views.

For more information, see *Working with Stories* or watch how to do it: *Editing Stories*.

1. Click ⚙ > **Open** to the left of a Story.

💡 **Tip:** If you display the **ID** column, you can simply click the number in that column to edit the Story.

The **Edit Story** dialog box opens.

2. Enter the details for the Story:

Title	Enter a descriptive name for the Story.
Owner	Select the resource responsible for the Story.
Type	Select the Type of Story: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information about Types, see *Working with Stories*.
Points	Enter the estimated number of Story Points required to complete the Story.
Status	Click one of the boxes indicating the status: `Not Started`, `In Progress`, `Complete`, or `Accepted`.

Priority	Select a Priority from the list: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.
Sprint	Select the Sprint for the Story, if needed.
Release	Select the Release for the Story, if needed.
Parent	The Parent Story is displayed. Click the link to display the Parent or click the **Set Parent** icon to change the parent.
Description	Add or edit the Story description.

3. To add Tasks to the Story, see *Adding Tasks to a Story*.

Sprint Activity Window

The **Sprint Activity** window is available in the **Team Room** for the **Sprint List** and **Sprint Cards** views. It provides activity that occurs on all Tasks, Stories, and Discussions for all users. You can filter Tasks by **Date** (From, To), **Type**, **User** and many more (see below). You can use this information to:

- Provide detailed support or explanation for data in the **Team Room** Charts (status changes, work record hours, etc).
- Filter by **Type** to see what was `Added/Created` or `Removed/Deleted` and by whom. You can also isolate Tasks by: `Edited`, `Worked`, `Discussed`, `Carry Over`, `Blocked/Unblocked`.
- Use the data however you need to as the basis for your daily stand-ups or Sprint retrospective.

From the top-right of the **Team Room**, click the **Sprint Activity** button: .

Moving a Story from One Sprint to Another

You can move a Story from one Sprint to another, or even back to the **Backlog** by using the gear menu in the **Sprints** Panel of the **Backlog** or in either view of the **Team Room**.

> **Note:** You can only move a Story to a current a future Sprint, not one in the past.

> **Tip:** Watch how to do it: *Moving Stories*.

1. Click > **Move...**.
2. Select the Sprint to move the Story to, or select **Backlog**.

Adding Stories to the Team Room from the Backlog

> **Note:** Parent Stories cannot be added to Sprints. Only the lowest level leaf item in your hierarchy can be added to a Sprint as a work item.

1. In the **Team Room**, click **Add Existing** . The **Add Stories from the Backlog** window appears.
2. Select the row or rows for the Stories to add and then click **Add to Sprint**.

Accepting a Story

The option to accept a Story is only available after all Tasks associated with the Story have been completed. When all the Tasks in a Story have been completed, the status of the Story changes to `Completed`.

At the end of the Sprint, the completed Stories are reviewed and accepted by the Product Owner.

Watch how to do it: *Accepting User Stories*.

1. Open the **Team Room** in either the **Sprint List** or the **Sprint Cards** view.
2. Click the last box in the **Status** column ▢ ▇ ▢ ▢ to **Accept** the Story.

Changing a Sprint's Capacity

A Sprint's capacity is measured in Story Points. You can modify capacity in the following ways:

1. Click ▤ **Planning** > ◈ **Timeboxes**, select `Sprints` from the **Type** list and double click the **Capacity** cell in the list and modify the value.
2. Click ▤ **Planning** > ☰ **Backlog**, open the **Sprints** Panel ⛹, click **Edit Sprint** and adjust the value in the **Capacity** field.
3. In the **Team Room**, click **Edit Sprint** and then modify the **Capacity** field.

Carrying Over Stories

Sometimes Stories don't get finished during a Sprint. You can carry over any incomplete Story into another Sprint or to the **Backlog**. Carrying over a Story performs the following actions:

- Creates a new Story with the same amount of points as the original.
- Carries over any Tasks that are not complete.
- Creates a relationship of type `Carry Over` to the original Story.
- The new Story title is prefixed with `[Carried Over]`.
- Discussions, Attachments, etc are not carried over to the new Story.
- The original Story **Status** control is yellow to indicate the carry over action.

1. Go to the **Team Room**.
2. Click ⚙ > **Carry Over** to the left of the Story.
3. Select where the Story should go: to another Sprint, or to the **Backlog**.

The carried over Story will appear first in the list and the **CARRIED OVER** image will display in the **Carried Over** column in grids.

Retrieving a Story URL

The following shows you how to retrieve a Story's URL so that you can link back to it from other Stories or third-party applications.

1. Click ▦ **Team Room** > ▤ **Sprint List** or ▦ **Team Room** > ▦ **Sprint Cards**.
2. Click the **ID** column to open the **Story Editor**.
3. Your browser's Address bar will contain the full URL that you can copy.

Using Quick Filters

In the **Team Room**, you can use a Quick Filter to display only certain information. The Quick Filter is only a single level filter, it does not apply compound conditions.

💡 **Tip:** When you apply a filter in the **Team Room**, the data in the **Team Overview** represents only that of the applied filter.

1. Click the **Filter** icon ▼ in the toolbar.
2. Select how you want to filter the data:

By Owner	Select the owner that you want to filter by.
By Type	Select the Story Type to filter by.

By Status	Select the Status value to filter by.
Stories	Contains several options for Stories in various states or by various owners.
Tasks	Contains the following options to filter Tasks by: `To Do`, `Not Estimated`, and `Actual > Estimate`.
Code Reviews	Contains various options for how to filter for Stories with Code Reviews.

After being applied, the filter icon changes to ▼.

Note: When a filter is applied, you can click the clear icon (⊗) to clear any filter.

Team Room Columns

Default Columns

The following columns display by default. If you modify them, your changes will be saved. For more information, see *Viewing Columns in Grids*.

Order	The **Order** column represents the numerical position of the item in the list. Double click to enable manually editing the value.
ID	Unique ID of the item. Read-only.
Type	Type is used as a way to categorize Stories. Available Story Types include: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information, see *Story Types*.
Name	Story/Task/Item Name. In most cases, you can double click to edit. The Story and Task **Name** fields have a max length of 255 characters.
Owner	Indicates who is responsible for Story.
Status	The **Status** column for a Story contains a bubble control ▢▢▢▢ that allows you to click on one of the bubbles to set the status of the Story: `Not Started`, `In Progress`, `Complete`, or `Accepted`. The **Status** column for a Task also contains a bubble control that allows you to set the Status by clicking on one of the bubbles. The number of bubbles depends on the amount of Task Status Values configured by your administrator. See *Change Task Status Values (Swimlanes)*.
Points	Indicates the number of Story Points it is estimated to take to complete the Story.
Tag	The **Tag** column allows you to select multiple pre-defined Tags for your Story. This is useful for filtering and organizing your Stories. Double-click the **Tag** cell for the proper row and select the Tags from the list to apply. For more information, see *Using Tags*. **Note:** This column displays by default for new Users and Projects. For others, you need to explicitly show the column. See *Viewing Columns in Grids*
Task Est.	**Task Estimate** indicates the estimated number of hours it will take to complete the Task. For Stories, this is a roll-up value. For more information, see *Adding Work Hours to a Task*.
Actual	Indicates the actual amount of hours worked on the Task. For Stories, this is a roll-up value. For more information, see *Adding Work Hours to a Task*.
To Do	Indicates the remaining hours left to work on the Task. For Stories, this is a roll-up value. For more information, see *Adding Work Hours to a Task*.
Blocked	Blocked indicates that there is an issue outside of the teams control preventing progress. If a Task is blocked, the parent Story will display **BLOCKED** in the grid. Click the ⓘ icon in the grid to block a Task. After entering a comment, It will change to ⊘. Additionally, the Sprint Health &

Progress area in the team Room will indicate that there are blocked Stories. For more information, see *Blocking a Task*.

Priority Priority is a list of values that includes the following: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.

Available Columns

Author	Name of the original author of the item. Read-only.
Carried Over	This column in the grid will display a special image **CARRIED OVER** representing that the Story was carried over from one sprint to the next (or more).
Created Date	Date that the item was created. Ready-only.
Code Reviews	The **Code Reviews** column will contain one of the various Code Review or Changes Set icons depending on whether there are active reviews or change sets related to the Story or Task. Click the icon to open the **Code Review** window (if a change set exists but a Code Review does not exist, you will be prompted to create one). A green icon indicates that all Code Reviews are accepted. Red indicates all reviews are rejected. Yellow indicates an indeterminate state where some are accepted/rejected or may not exist yet.
General	The **General** field is a text-based field that can be used to capture any business or development process data as needed by the project team. It is not used in Rhythm for any explicit purpose. It has a maximum length of 128 characters and is only available in grids that contain Story data.
Global ID	This is a unique identifier for all items in the Atlas Hub.
Impact	This column displays the Change Packages icon ⟨⟩ if the Story or Task is related to a Change Package. Additionally, the icon will be decorated with the number of items contained in the Change Package. Click the icon to open the **Change Packages Viewer**.
Input Stream	This column displays the **Input Stream** icon ⥮ if the item was derived from an Input Stream. Hover over the item for more information or click the column to open the external item. For more information, see *Input Streams*.
Kanban Status	If a Story is present in a Kanban Board lane, this column will display the board **Name** and lane in the following format: `<Board Name>: <Lane Name>`.
Last Edited	Date that the item was last edited. Ready-only.
Last Edited By	Name of the last author who last edited the item. Read-only.
Revision	Version number of the item. Read-only.

Kanban

When a Rhythm *Project Type is Kanban or Scrum & Kanban*, the **Team Room** menu will display a list of your Kanban Boards. Select a board to view it.

Click ⊞ **Team Room** > 🔲 **<Kanban Board Name>**.

The **Kanban Board** contains three sections: Information Pane, Filters, and the Swimlanes.

Use the topics in this section to learn how to work in the Kanban Board.

Kanban Board Information Pane

The Information pane in ⊞ **Team Room** > 🔲 **<Kanban Board Name>** contains the following information:

Board Selection

This list contains all of the Project's Kanban Boards that have been made available to the **Team Room**. See *Setting Kanban Board Options*.

Cycle Time Label

Displays the average amount of time it takes for a card to progress from the first lane to the last lane.

WIP Summary Label

Number of Stories in all lanes not including the **Board Backlog** or the last lane.

Card Blocked Icon

If a card is blocked, you will see this indicator:

 Cumulative Flow Button

Click to view the Kanban **Cumulative Flow** Diagram.

The Kanban Cumulative Flow diagram represents the number of Stories (y-axis) on any given day (x-axis) and the lane in which they were in. The colors in the chart represent the different lanes. If more than eight weeks of data exists in the lanes, it is broken into two week increments.

If a `Custom Filter` is used as the board input, the start date is the date that the first item was added to the board and the end date is the current day (today).

> **Note:** To zoom in on a region, right click and drag the region. To go back one zoom level, double click the graph or click the **Undo Zoom** button.

 Discussions Button

Each **Kanban Board** contains its own Discussion. For more information, see *Discussions*.

 Administrator Button

Visible only for System or Project Administrators. Click to open the **Kanban Board Administrator** window. From here, administrators can customize their board. For more information, see *Kanban Board Administration*.

Kanban Board Filters

The filter pane contains four filters: **User**, **Type**, **Tag**, **Quick Filter**.

- Quick Filter contains options for filtering Stories with Code Reviews.
- Select any filter to apply to the **Kanban Board**. Note that this is a single-level filter.
- When a filter is applied, any lanes without cards are collapsed by default.

- After a filter is applied click the **Clear Filter** button ⊗ to remove the filter.

Working in the Swimlanes

The **Kanban Board** is divided into Administrator-defined swimlanes with your Stories represented as cards.

All Cards start in the initial **Board Backlog** lane. Use the information in this topic to help you understand how to use the swimlanes.

Note: For administrative information, see: *Managing Swimlanes*.

Board Backlog Lane

All boards contain the **Board Backlog** lane. This lane is the starting point for all Stories that match the *Board Backlog Source*.
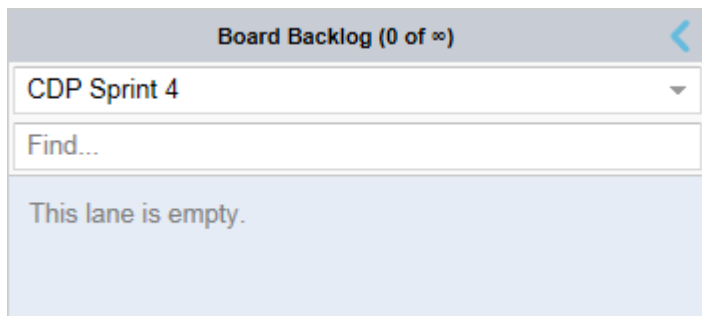
In this lane you can:

- Drag-drop Cards into other lanes.
- Use the **Sprint** list to switch the active Sprint.
- Type text in the quick **Find** field to filter the Cards.

Using the Lanes

The lanes appear like this:
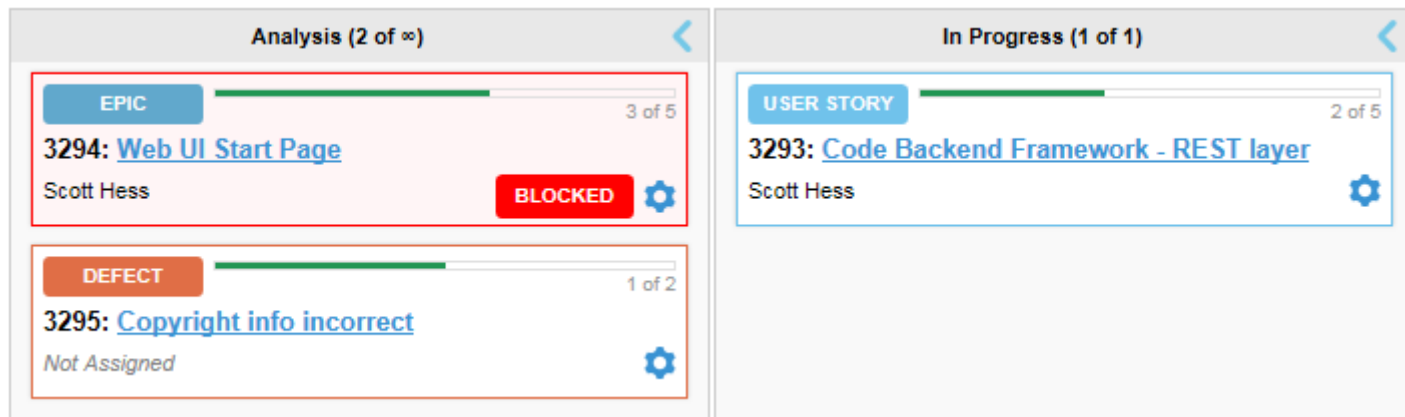
The lane headings include name and indicate WIP and Limit, if those options are turned on.

Click the arrow to expand or collapse a lane.

Lanes are expanded by default unless a filter is applied.

Using the Cards

The Story card in the lane is interactive and you can do the following with it:
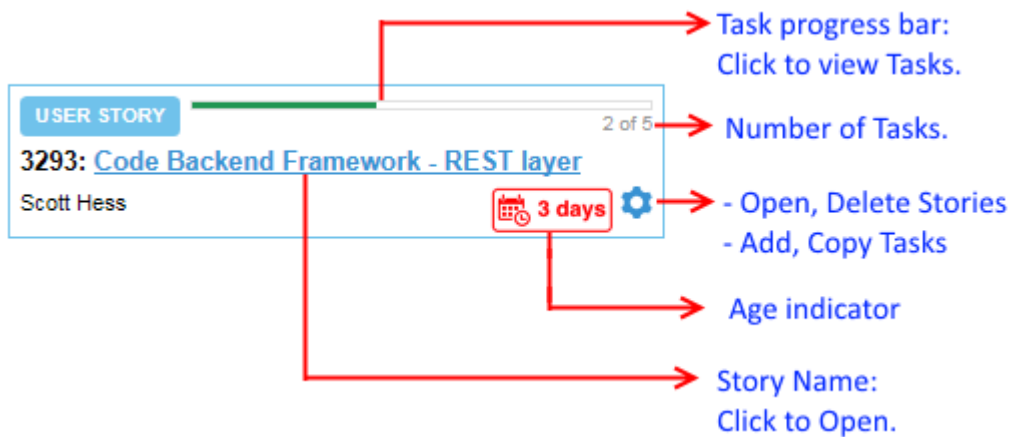
Use the **Gear** icon 🔧 in each card to:

Open	Opens the **Story Editor**. See *Working with Stories*.
Add Task	Adds a Task to the Story. See *Working with Tasks*.
Copy Tasks From...	Opens the **Copy Tasks from** dialog box. See *Copying Tasks from Other Stories*.
Remove	Removes the Card from the current lane. If the Story still meets the Board's *Board Backlog Source criteria*, the Story returns to the **Board Backlog** lane. Otherwise, it will only be available in the **Planning** perspective.
Delete	Deletes the Story.

Using Tasks

When you click the Task progress bar, you get a preview of the Story's Tasks:

ID ↑	Type	Name	Owner	Estim...
5019	DEV	API Calls	Administrator	5.00 Hr
5020	DEV	Database Model	Scott Hess	10.00 Hr
5021	DEV	Schema Editor	Scott Hess	6.00 Hr
5022	QA	Test	Administrator	0.00 Hr
5023	DOC	DOC	Administrator	0.00 Hr

From here you can:

- Click the **ID** column to open the Task.
- Manage which columns to view in the window and to take action upon. See *Viewing Columns in Grids*.

Drag-Drop

Use `drag+drop` to move Stories around lanes.

Moving to a lane provides popup feedback as to lane availability:

✅ Move to this lane	Indicates a valid move.

⊘ WIP Limit Reached	Indicates that the **WIP Limit** has been reached and the Story cannot enter the lane.
⊘ Already in this lane	Indicates that you are attempting to move the item to the lane it is already in.
⊘ Saving. Please Wait.	Indicates that the item is saving and cannot be moved.

Adding Stories to a Kanban Board

Stories are automatically added to a board based on the *Board Backlog Source*. However, you can explicitly create a Story on a board. When you create a Story, it is assigned to the first real lane of the board after the **Board Backlog** lane. The Story does not inherit any data (Sprint/Release/Filter) based on the **Board Backlog Source**. Even if the board **Backlog Source** changes, the Story will remain on the board.

You can create 🗋 *Quick* or 🗋 *Detailed* Stories.

To create a 🗋 Detailed Story:

1. Click ▦ **Team Room** > ᗰ **<Kanban Board Name>**

2. Click 🗋 **New Story: Detailed**.

Create Backlog Item

Not created yet.

Title...

Owner:	Not Assigned
Type:	User Story
Points:	0
Status:	🟩 ⬜ ⬜ ⬜
Priority:	Could Have
Sprint:	Not Assigned
Release:	Not Assigned
Parent:	No Parent ✏ ⊖

B *I* U S̶ x₂ x² Iₓ ⅈ☰ ⅈ☰ ⅈ☰ ⅈ☰ ☰

Format ▾ | Font ▾ | Size ▾ | **A**▾ **A**

✂ 🗋 📋 📋 📋 ↶ ↷ 🔍 ᵇ⁺ₐ 📋 🖼 ⊞

🔗 🔗̷

Save   Cancel

3. Enter the details for the Story:

Title	Enter a descriptive name for the Story.
Owner	Select the resource responsible for the Story.
Type	Select the Type of Story: `User Story`, `Technical To Do`, `Epic`, `Theme`, or `Defect`. For more information about Types, see *Working with Stories*.
Points	Enter the estimated number of Story Points required to complete the Story.
Status	Click one of the boxes indicating the status: `Not Started`, `In Progress`, `Complete`, or `Accepted`.
Priority	Select a Priority from the list: `Must Have`, `Should Have`, `Could Have`, or `Won't Have`.
Sprint	Select the Sprint for the Story, if needed.
Release	Select the Release for the Story, if needed.
Parent	The Parent Story is displayed. Click the link to display the Parent or click the **Set Parent** icon to change the parent.
Description	Add or edit the Story description.

4. Click **Save**. The Story is added to the first lane of the board.

Tracking

The **Tracking** perspective is a resource for project managers and product owners who want an overview of how the release is progressing. It includes charts to show progress as well as a list of all stories targeted for the current release organized by type. This view includes easily understandable charts for **Progress**, **Health**, **Cumulative Flow** and **Breakdown**.

In *Agile* development, a product can be released to a customer after every iteration or after several iterations or Sprints. Rhythm uses the *Release* functionality to organize Sprints. You must create a *Release* before you can start creating and planning for Sprints.

When you are planning for a Release, the **Target Release** function allows you to add Stories to target them for a Release during release planning. These Stories will be part of the calculations for *Release Tracking*.

⚠ **Important:** To make the best use of this feature, it is important to target Stories in your **Backlog** for release. Only Stories that are targeted for the current Release will be included in the *Release Tracking* calculations.

Release Progress

The summary information at the top of the page provides a snapshot of the current status of the release to allow the team and the product owner to determine if any changes need to be made in order to meet the project deadlines. Information in the summary includes:

- Release start and end dates.
- Remaining Days.
- Remaining Work (Points or Stories).
- Operating Velocity (Number of Points/Stories per week).
- Operating Capacity (Percentage of Points/Stories).

✏ **Note:** If a Story is not estimated, it is not included in the release tracking calculations and a warning is displayed in the **Health** section indicating that there are Stories without estimates.

The **Progress** bar indicates the number of Story Points `Accepted`, `Completed`, `Started`, `Blocked`, and `Not Started` in a release.

Project Tracking using the Tracking Perspective

You can adjust the **Tracking** perspective to show Point or Story count-based Project metrics. Projects that use Story Points for planning and executing will find the point-based tracking metrics useful. For Projects that do not use Point-based planning, the **Use Number of Stories** option may provide more value.

To do this, select one of the following options:

> Use Story Points	Tracking metrics are based on Stories and assigned Story Point values.
> Use number of Stories	Tracking metrics are based on Story counts (any provided Points are not considered).

Release Health

The following information is contained in the **Health** section on the **Tracking** perspective:

Estimated Stories	Lists the number of Stories not estimated or says that all Stories are estimated.
Projected Completion Date	A calculation is performed that is based on the **Operating Velocity**. If it's after the defined Release boundaries, it is red and provides the *remaining days*.
Velocity Suggestion	Displays the weekly velocity required to meet the Release end date
Payload Suggestion	Suggests the necessary changes to meet the Release end date.

Release Cumulative Flow

The **Tracking Cumulative Flow** chart represents the amount of work in the Release in different states for each Sprint of the Release. The vertical axis represents the total number of Points or Stories in the Sprint (depending on whether the release was planned and tracked with Points or Stories). The horizontal axis represents each Sprint of the Release. It includes **Points Not Started** (light green), **Points In Progress** (green), **Points Completed** (dark green), and **Points Accepted** (blue). The height of the chart defines the scope. If something is added or the estimate is increased, then the whole chart gets taller. If something is removed or the estimate decreased, then the chart gets smaller. It provides the ability to see changes in scope in the Release.

Tip: To change the vertical axis setting between Points or Stories, see *Release Progress*.

Release Breakdown

The **Release Breakdown** includes all of the stories targeted for the release separated by Story Type and in hierarchical order.

Working with Stories

A *User Story* is a definition of a discreet unit of work to perform on your product. Stories exist in the **Backlog**, in a Sprint, or in a **Kanban Board**. They are modified with the **Story Editor**.

Rhythm provides many ways for you to create, to edit, and to manage your Stories:

- *Breakdown View*
- *Backlog View*
- *Team Room*
    - *Sprint List*
    - *Sprint Cards*
    - *Kanban Boards*

You can also automate the creation of Stories using other tools in the Atlas Planning and Tracking Suite:

- Use Input Streams to pull in Requirements and Defects as Stories. See *Configuring Input Streams*.
- Use Atlas to push full Plans or individual Requirements into Rhythm. See *Using Rhythm with Atlas*.

Story Types

Stories can be any of the following types:

User Story	Used to capture the description of a software feature from an end-user perspective. It helps create a simple description of a Requirement.
Epic	An *Epic* encapsulates a large body of work or numerous Stories. It should be broken down into numerous smaller Stories.
Theme	A theme is used to track of group of Stories that share common traits and can be grouped together.
Tech To Do	Describes a Story that usually contains technical debt.
Defect	Used to track actual product Defects

Using the Story Editor

Use the **Story Editor** to create your Story. The **Story Editor** contains four sections:

Story Details	This pane in the top-left contains the following fields: **Owner**, **Type**, **Points**, **Status**, **Priority**, **Sprint**, **Release**, and **Parent**.
Title and Description	The **Title** is a standard field but **Description** uses a rich-text field with all of the editing capabilities that you would expect.
	Note: **Title** has a maximum size of 255 characters and **Description** has a maximum size of 2 GB including images.
Tasks Pane	Define your Tasks for the Story in the **Tasks** pane. For more information, see *Working with Tasks*.
Extensions Pane	The expandable right-hand pane allows you to do the following with Stories:
	Discussions Create a Discussion to share with the team. See *Discussions*.

📎 **Attachments**	Add files to your Story. See *Using Attachments with Stories*.
🔺 **Relationships**	Create relationships to or from your Story to other assets. See *Relationships*.
	💡 **Tip:** The **Relationships** tab contains links to Code Reviews and Change Packages related to the Story.
🏷 **Tags**	Create Tags to label similar Stories. See *Using Tags*.
⚗ **Tests**	Click to view test results associated with the Story. See *Viewing Test Results*.
📄 **Versions**	Review every change made to your Story. Compare as needed. See *Viewing and Comparing Versions of a Story*.

Discussions

Discussions allow you to share ideas and have in-depth conversations using *Topics*. These Topics allow team members to provide feedback via **Agree/Disagree** icons as well as responding directly to the Topic via messages. When the all of the team's feedback is provided, the Topic can be `Accepted` or `Rejected`. Until then, it is in the *Open* state.

Discussions are available for each Story and for each Sprint or in a Kanban Board. Click the 💬 icon to access them. For the Story Editor, it's a tab in the right pane. For the Sprint or Kanban Board, it's an icon in the top right of the **Team Room**.

When a Discussion exists in a Story, the icon on the tab in the **Story Editor** is decorated like this: 💬

✏ **Note:** The highlighted number next to the Discussion icon represents the number of Open Topics in the Discussion.

- A Discussion can contain multiple Topics.
- A Topic contains feedback in the form of:

  - Textual messages via **Reply** ↩.
  - **Agree/Disagree** (👍 and 👎) click responses.
- A Topic is in the `Open` state until you `Close` it via **Accept** or **Reject**. You could re-open it later.
- When replying to a Topic, if you want to notify someone, use the @ symbol and select a user from the list provided. Note the following:

  - The @ symbol can be used multiple times to address a message to multiple users.
  - The @ symbol causes emails to be sent to all listed users (if email integration has been enabled in the Atlas Hub).
  - A reply will message the user that started the topic, all of the users that have posted a reply to the topic, and any users mentioned anywhere in the discussion thread.
- When you Block a Task, you are prompted with the **Provide Blocking Reason** dialog box. When you enter the reason and click **OK**, a new Discussion Topic is added to the Story and all team members with Tasks in the Story are notified. When you unblock a Task and provide a reason, likewise, all team members receive a reply.
- Watch how to use them: *Using Discussions*.

Topic Life Cycle Diagram
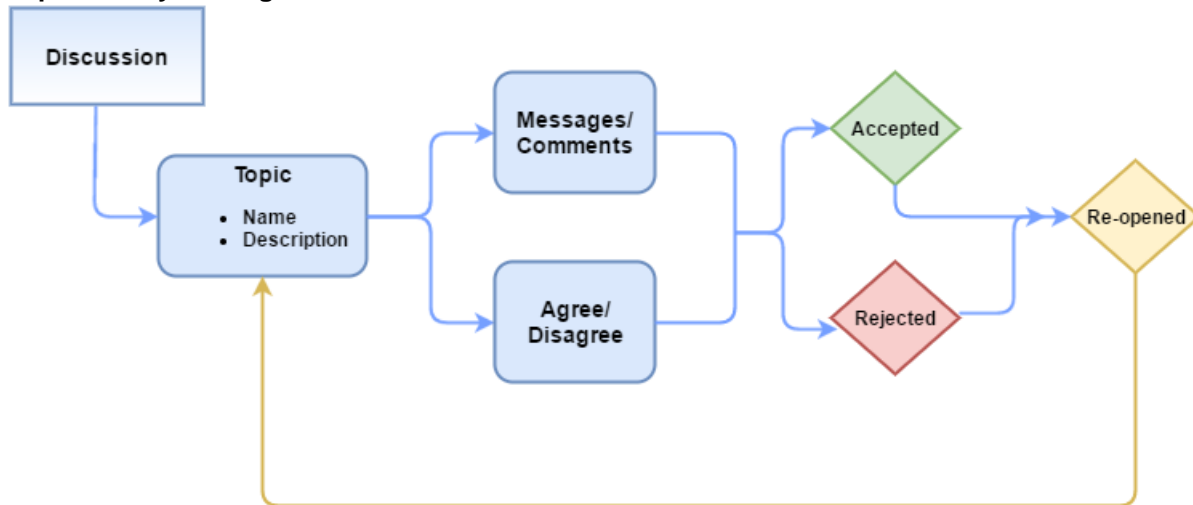
Using Attachments with Stories

You can add attachments to your Story to include as related content. Users can upload, download, and replace attachments as needed. As long as users have rights, they can also delete an attachment. There is no limit as to the size of the attachment, however, the larger the attachment, the longer the upload/download time. There is a limit of 1024 attachments per Story.

1. Open the **Story Editor**.
2. Click the **Attachments** 🔗 tab. When there are attachments in the Story, the tab is decorated like this: 🔗.

   To Add an Attachment:

   1. Click 🔗 Attach File. The **Attach Files** dialog box opens.
   2. Select one or more files to attach.
   3. Click **Add**.

   To Open an Attachment:

   1. Hover over the row of the attachment.
   2. Click **Download File** 📂 to the far right.

   To Replace an Attachment:

   1. Hover over the row of the attachment.
   2. Click **Replace with new file** 🔲. The **Select File** window opens.
   3. Click **Select** to browse and select a file.
   4. Click **Update**.

   To Delete an Attachment:

   1. Hover over the row of the attachment.
   2. Click **Delete** 🗑 .

   To Sort Files:

   1. Click **Sort** 🔤.
   2. Select how you want the attachments sorted: `Time Added`, `Filename`, or `User`.

Relationships

Rhythm uses Relationships to help you understand how different items are connected. Changing an object may have implications on other elements of the Project. For example, changing a Requirement that has already been pushed as a Story in your Agile planning tool might change the scope or complexity of the Story. Relationships are supported to allow you to see the connection between Stories and other related assets. Linking related objects together helps to ensure that changes are implemented correctly at all levels.

You can create relationships manually in Rhythm through the **Relationships** tab in the **Story Editor**. When a Relationship exists in a Story, the icon on the tab is decorated to indicate that a Relationship exists, like this:

Rhythm also creates some Relationships automatically for you. Whenever you push Requirements to the Rhythm **Backlog** from Atlas, a Relationship between the Requirement and the Story is created. When you pull in Requirements or Defects using Input Streams, a relationship is automatically created. Additionally, Code Reviews are listed in the **Relationships** tab. Click the **Goto** icon to open the Review.
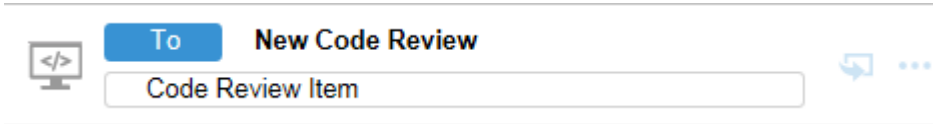
The following lists the types of relationships used within the Atlas Planning and Tracking Suite:

User Managed/Created	`Basic`, `Copy`, `Dependent`, `Requirement Stream`, and `Requirement Synchronization`.
Application Managed/ Created	`Carry Over`, `Defect Stream`, `Execution` (relationship between a Story and a Sprint, for example) , `Parent-Child`, and `Requirement Stream`.

Watch how to do it: *Creating a Relationship*.

Relationship Status

Each relationship type may have various statuses. When a change is made to an item with a Relationship, the relationship status changes to **Suspect**. You can view the change and update the status to bring it up-to-date, or you can **Defer** the update. If you defer, the status of the Relationship changes to **Defer**.

⚠️ **Important:** For all Relationships, except for synchronizations, changing the status does not change the items. The status acts as a label to help you find what you've reviewed and what you need to review. For example, updating the status changes the status to **up-to-date** but does not make any changes to the target item details.

Creating a Relationship

1. Open the **Story Editor**.
2. Click the **Relationship** tab 🔗 . The **Relationship** pane appears.
3. Click 🔗 **New Relationship**.
4. Select the Project from the **Project** list.
5. Select one or more items with which to create a relationship.
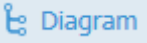
    If you can't find the item(s), use the **Find** field.

    If you want to create a relationship to a new Story, click **New Story**.
6. Click **Add**. The **Relationship Type & Direction** dialog box opens.

7. Select `Basic Relationship`.

8. Click **OK**.

Visualizing Relationships

Relationships are accessed from the **Relationships** tab and appear in a list. In certain perspectives, you can view your relationships in a diagram. When you see ⠿ Diagram above the **Relationships** list, click it to open the **Relationships Diagram** window. Here's how it works:

- Items to the left of the Story link *to* the Story. Items to the right of the selected Story link *from* the Story.
- The diagram displays an item's first level relationships. Items with child relationships contain a solid blue dot. You can click and navigate to all relationship levels. Cross-project relationships links (where the Story is in another project than the current one) are indicated with a dashed line.
- Select an item and you will see summary information on the top of the window.
- Click 🔙 (**Goto**) to go to the item in a new browser window/tab.
- The diagram is an infinite scrolling canvas that can be panned via click and drag, and zoomed via the scroll wheel.
- Click **Reset** to restore the default pan/zoom.
- Click **Reload** to refresh the diagram.
- Although item names are fixed-length on the diagram, hovering over an item displays a tool-tip with an asset's full Name, ID, Type, and Project Name.
- Select a **Relationship Status**, **Relationship Type**, or **Item Type** to filter the diagram based on your selections.
- If an item is a Change Package, you can open the **Change Package Viewer** directly from the diagram. Click the button [</>] in the viewer.

Relationship Diagrams have indicators to help determine relationship and status at a glance:

- Green lines indicate an `Up-to-Date` relationship.
- Red lines indicate a `Suspect` relationship.
- Orange lines indicate a `Deferred` relationship.
- Gray, solid lines indicate a parent-child relationship.
- Dashed lines indicate a relationship with an external project (cross-project).
- Solid lines indicate a relationship within the current project.

Additionally, if you hover over the connecting lines, status and details appear.

Relationship Diagram Example
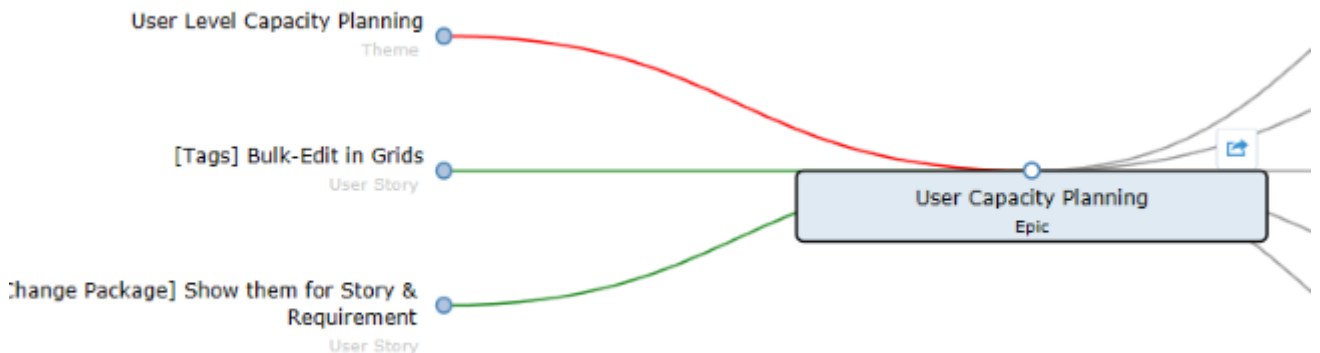
Viewing Status and Details of Relationship Items

You can view the status or detail of an item with a Relationship. Rhythm items display status of Requirements or Stories. Atlas Hub items display according to the defaults or, in the case of custom Atlas Hub items, according to what the administrator created. Atlas Hub items are displayed in the **Other** tab and are organized in folders and files in a file explorer format.

1. Click the **Relationship** tab 🔧. The **Relationship** pane appears.
2. Scroll through the **Existing Relationships**. Each Relationship shows the status or details.

   🖉 **Note:** Select the status from the **Relationship Status** list to filter the list based on status or select Project, Relationship Type, Artifact Type, or Direction from the **Group By** list to group relationships.

Table 1: Relationship Items

Icon	Type	Description
`</>`	Change Package	A collection of changes from your SCM tool related to a commit.
⇄	Change Request	An Atlas Hub generic Request object.
🖥	Code Review	A Rhythm Code Review.
N/A	Concept	An Atlas Concept
📄	File	A reference to a file.
🗂	Kanban Board	A Rhythm Kanban Board.
N/A	Plan	An Atlas Plan.
◈	Requirement	An Atlas Requirement.
N/A	Release	A Rhythm Release.
🏃	Sprint	A Rhythm Sprint.
💲	Story	A Rhythm Story.
☑	Task	A Rhythm Task.
N/A	Test	An Atlas Hub Test object.
？	User Defined Type	A custom type defined in the Atlas Hub.
▢	Whiteboard	An Atlas Whiteboard.

Updating Relationship Status

1. Open the **Story Editor**.
2. Click the **Relationship** tab 🔧. The **Relationship** pane appears.
3. Scroll through the **Existing Relationships**. Each Relationship shows the status or details.

   🖉 **Note:** Select the status from the **Relationship Status** list to filter the list based on status or select Project, Relationship Type, Artifact Type, or Direction from the **Group By** list to group relationships.

4. Click **...** and choose to **Update** or **Defer** suspect relationships.

Status Actions

Deferred When a relationship is in this status, the available actions are `Compare`, `Update` and `Remove Relationship`.

Suspect When a relationship is in this status, the available actions are `Compare`, `Update`, `Defer` and `Remove Relationship`.

Using Tags in the Story Editor

In Rhythm, a *Tag* is a label that you define and then apply to Stories. What you need to know about Tags:

- They can be Project-specific or Server-specific:

  - A **Server Tag** is available to all Projects on the server.
  - A **Project Tag** is only available for the current Project.
- They can be nested up to one level deep.
- They can contain spaces, numbers, and special characters.
- They can be applied in grids by double clicking the cell in the **Tag** column to display the Tag list.
- Watch how to use them: *Tagging Stories*.

To manage tags, click the **Tags** tab 🏷️ in the right pane of the **Story Editor**. If Tags exist, the tab icon is

decorated like this: 🏷️. There are two panels:

Assigned Tags Contains any Tags already assigned to the Story.

Available Tags Contains all available **Project** and **Server** Tags, in a tree.

To Create a Tag

1. Click 📄 **New Tag**.
2. Enter the **Name**.
3. If the Tag is to be available for all Projects on the server, click **Server Tag**.
4. Click **Save**.

To Create a Nested Tag

1. Select a Tag.
2. Click the **Gear** menu ⚙️ on the right.
3. Select **Add Child Tag...**
4. Enter the **Name**.
5. Click **Save**.

To Remove a Tag from a Story

1. Select a Tag in the **Assigned Tags** list.
2. Click the **Remove** icon ⛔.

To Delete a Tag

1. Select a Tag in the **Available Tags** list.
2. Click 🗑️ **Delete**.

Viewing Test Results

When you synchronize Requirements or Stories between the Atlas Hub and a testing tool, Rhythm can display the test results returned by your testing tool. In order to do this, your system administrator needs to use Micro Focus Connect to connect the two systems.

For more information, administrators should see *Accessing Connect*.

To view test results for a Story:

1. Open the **Story Editor**.

2. Click the **Tests** tab 🧪.

There are two graphs:

Latest Test Results	This graph provides an aggregation of the statuses of all the Tests assigned to a Story , as they appear the last time they were executed. **N/A** indicates Tests that are assigned but they have **never** been executed (for example, a brand new test).
Test Results Per Day	Contains the following: • The **Duration** on the X-axis of the graph correlates to the duration supplied in the **Admin Tools**. The number of tests in the Y-axis is that of the selected item. • Click the legend (`Passed`/`Failed`/`Not Run`) at the top of the graph to turn each one on/off. • The **Goto** button takes you to the equivalent artifact in the integrated test tool. • Hovering over an item gives you a tool tip with details.

Viewing and Comparing Versions of a Story

Every time a Story is modified, a new version is created. You can use the **Story Editor** to review and to compare two versions of a Story.

Watch how to do it: *Review Version Information*.

1. Open the **Story Editor**.

2. Click the **Versions** tab 📑 in the right pane.

3. Select a version and then `ctrl+click` another version and then click 🔲 Compare. The **Compare Versions** dialog box opens with columns detailing each version of the Story. The fields that you can compare are grouped by various types (**System**, **Description**, **Attachments**, **Extended Attributes**, etc) and displayed in a tree.

4. Click **Options** > **Only show differences** to hide all items that haven't changed.

5. Click **Options** > **Extended Attributes** to show various other fields that may have changed.

Accessing Stories in Native SCM Environments

When your native SCM environment is connected to the Atlas Hub, Rhythm provides links to those Stories when those Stories have files attached. To access your Story in your native SCM environment to check in/check out files against a Story:

1. Open the **Story Editor**

2. If the Story contains files in your SCM environment, the **ID** field will contain a clickable link. Click it.

Story Details

ID: 2361
Revision: 1.3
Modified: January 31, 2017, 3:20 pm
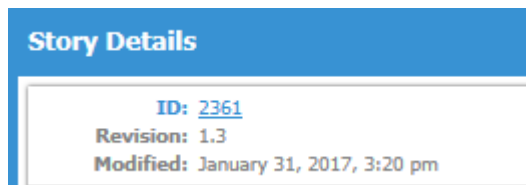
Your native SCM environment opens to the Story.

✏️ **Note:** Depending on your environment, you may need to have certain permissions for the project to view it.

Using the Change Packages Viewer

When you are in the **Team Room** or the **Story Editor**, you have the option of adding the **Impact** column to your grid. When you do this, you will see the number of Change Packages related to your Story/Task.

In the **Story Editor**, the Task grid will display the number of related Change Packages for each task. In the **Team Room** grid, the count of all Change Packages related to your Story/Task are displayed. Story Change Package counts represent the number of all related Change Packages (directly or indirectly related through Tasks).

Click the row in the **Impact** column to open the **Change Packages Viewer**.

Change Packages for CP Story

Name	Item Type	Change Type	Commit Action
− **Initial Checkin**			
Your Doc.txt	File	New in workspace	Add
Source.txt	File	New in workspace	Add
My Doc.txt	File	New in workspace	Add
More Source.txt	File	New in workspace	Add
Write Code	Task	Unchanged Process Item	Mark Resolved
− **Documentation Changes**			
My Doc.txt	File	Modified in workspace	Checkin
Write Documentation	Task	Unchanged Process Item	Mark Resolved
− **Dev WIP**			
Source.txt	File	Modified in workspace	Checkin
More Source.txt	File	Modified in workspace	Checkin
Edit Code	Task	Unchanged Process Item	Mark Resolved

The Change Packages viewer displays:

- All items contained in the Change Package, grouped by **Name**. (You can group by any of the columns).
- The **Item Type**.
- The **Change Type**.
- The **Commit Action**.

If you click an item, such as a File, it will attempt to open the native application from which the item came. Rhythm assets, such as Tasks, will be opened in Rhythm.

Tip: You can also open the **Change Package Viewer** from within the **Relationship Diagram** window. Click the button to open the window:
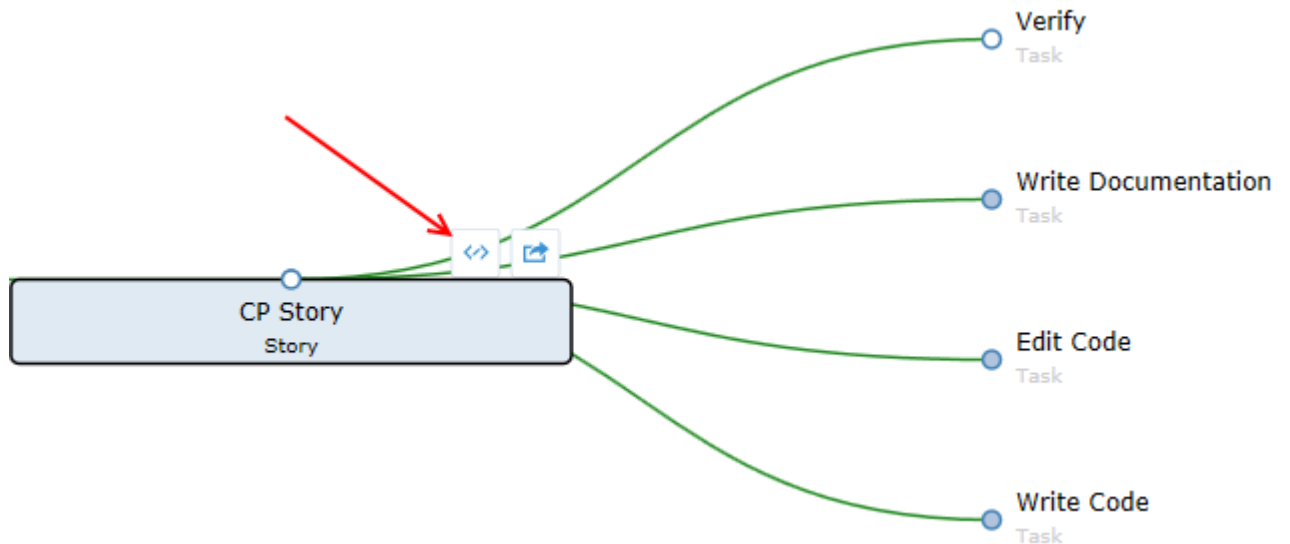
Working with Tasks

Tasks are the units of work used to define a Story. Tasks contain work records to track individual hours.

You can create a *Quick Task* or a *Detailed Task*.

A Task can be in the default **Status** value of `Not Started`, `In Progress`, or `Complete`. However, the available Task **Status** values can be customized to suit your organization's needs. For more information, see *Changing Task Status Values (Swimlanes)*.

▶ Watch how to *Update Task Status*.

💡 **Tip:** You can copy all Tasks from an existing Story by clicking 📄 **Copy From** on the toolbar. You'll navigate the list of Stories and select the one from which you want to import the Tasks. For more information, see *Copying Tasks from Other Stories*.

💡 **Tip:** You can click the gear menu ⚙ and select **Add Task** to add a Quick Task (`Type`, `Title`).

Creating a Detailed Task

Follow the steps below to create a detailed Task:

▶ You can also watch how to do it: *Create a Detailed Task*.

1. Open the **Story Editor**.
2. Click **Detailed Task**.

The **Task Editor** appears.

3. Enter the details for the Task:

Title	Enter a descriptive name for the Task.
Owner	Select who will performing the Task.
Type	Select the type of Task it is: `Development`, `QA`, or `Documentation`.
Estimate	Enter the estimate of the number of work hours required to complete the Task.
Status	Click one of the Status boxes: `Not Started`, `In Progress`, or `Complete`. These are the system default values. Your administrator can change these values. See *Change Task Status Values (Swimlanes)*.
Blocked	Click ⊙ to block the Task. For more information, see *Blocking a Task*.
Actual	Displays the actual number of hours that have been worked on the Task.
To Do	Displays the number of remaining hours to complete the Task. See *Adding Work Hours to a Task*.
Description	Enter the information required for the task **Owner** to complete the Task. You can also include links to documents that may be stored elsewhere.

4. Click **Save**.

Creating a Quick Task

Follow the steps below to create a quick Task.

⊡ Optionally, watch how to do it: *Create a Quick Task*.

1. Open the **Story Editor**.
2. Click **New Task: Quick**.
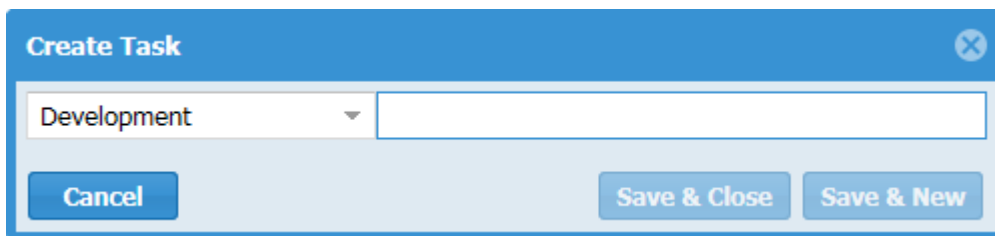
The **Create Task** dialog box appears.

3. Select the **Type**.
4. Type in the **Name**.
5. Click **Save & Close** or **Save & New**.

Task Assignment and Estimation

The following should be considered and noted when planning or tracking Task estimates, Actual Hours worked, and individual User Capacity:

- Each User should enter their Project or Sprint capacity. See *Managing User Capacity*.
- When assigning an Owner to a Task, a User's **Total** and **Remaining** hours are indicated in the list.
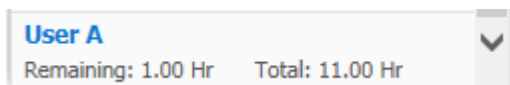
- To see these values, the following is required:

  1. The Project must be marked `Private`.
  2. The Story must be in a Sprint.
  3. They are only available in the **Team Room** and **Story Editor**.

- If you edit the **Task Estimate** field, the capacities are calculated and updated for the current Owner if there are no work records, otherwise it will calculate based on the remaining work (**To Do**).
- If you change a User, it will update both the old and new User to reflect the newly added (for the new user) and removed (for the old user) estimated hours of work remaining on a task.

Copying Tasks from Other Stories

If you find that you often use the same set of Tasks in a Story, you can set up a generic Story with all the Tasks you need and then import the Tasks into any new Story you create.

⊡ Watch how to do it: *Copy Tasks*.

1. Open an existing Story in the **Story Editor** and click **Copy From** 📄.

   💡 **Tip:** Alternatively, in the **Backlog** or **Team Room**, select a Story and then click ⚙ > **Copy Tasks From...**.

   The **Copy Tasks From** dialog box appears.

2. Navigate to the Story that contains the Tasks you want and select it.

   💡 **Tip:** Click the **Preview Tasks** icon 🔍 to display a popup list of the Tasks in the Story. Click outside the window to make it go away.

3. Click **Copy**. The Tasks are added to the Story.

Work Record Management

The **Task Editor** contains a **Work Records** pane at the bottom. Work Records are sorted and calculated based on **Work Date** and then **Created Date**. Each work record is listed as a row in the list. You can:

- Create a new record by clicking ⏱ **New**.
- Edit existing records by double clicking the cell and entering/selecting a new value.
- Delete a record using the **Delete** button.

Work Record Columns

User	Editable. Can be changed to any contributor in the project/any user on public projects.
Worked	Editable. Enter hours worked.
To Do	Editable. Enter remaining hours.
Work Date	Editable. Date the work was performed.
Last Edited	Not editable. Last edited date.
Last Edited By	User who last edited the item.

Adding Work Hours to a Task

Work hours are the units of time used to plan and to complete a Task. You can add your time to your tasks from several areas:

1. Open the **Task Editor**. The **Work Records** pane is at the bottom of the editor. Click ⏱ **New** to open the **Record Time** box.
2. In ▦ **Team Room** > ▪▪ **Sprint Cards**, on a card, click the ⏱ button to open the **Record Time** box.
3. In the ▦ **Team Room** > ▤ **Sprint List**, double click the ⏱ button in the **To Do** column to open the **Record Time** box.

For more information, see *Status and Work Hours*.

About Status and Work Hours

The following information describes how changing work hours on a Task or changing the Task **Status** impact the Story/Task.

- When the first work record is entered, the Task **Status** is changed to `In Progress`.
- The amount of time entered in the **Worked** field is automatically deducted from the **To Do** field.
- When a Task is marked `Complete`, **To Do** goes to 0. Likewise, when you enter `0` in **To Do**, the Task **Status** is set to `Complete`.
- When **Worked** entered is greater than remaining **To Do**, **To Do** goes to 0.
- When a Task is estimated or work records are added, these values are saved and are available to see when you select the **Owner** field of the Task. For more information, see *Assigning and Estimating Tasks*.

Blocking a Task

Any time an issue outside of the teams control prevents progress, you should *Block* the Task .

⚠ **Important:** When a Task is blocked, the Story is also *Blocked*.

💡 **Tip:** If you are in the **Task Editor**, you can click the Blocked icon ⓘ in the left pane to block a Task.

▶ Watch how to do it: *Block a Task*.

1. In the **Team Room**, make sure that the **Blocked** column is visible. See *Viewing Columns in Grids*.

Type	Name	Blocked	Status
USER STORY	▼ Very Important User Story		▢▢▢▢
DOC	Create Documentation	⚠	▢▢▢

2. Hover over the Block icon ⚠ in the **Blocked** column for the Task you want to block.

3. Click the icon when it enables ⚠. The **Provide Blocking Reason** dialog box opens.

4. Enter a reason in the field and click **OK**. This automatically creates a Discussion Topic in the Story and notifies the Story Owner, all team members with Tasks in the Story, and anyone else called out with the @ functionality. For more information, see *Discussions*.

5. The following also happens when a Task is Blocked:

   - The **Title** is modified to include the text `Story Blocked`.
   - The icon in the **Blocked** column changes to 🛑.
   - The **Story** row will display **BLOCKED**.
   - The Task row is highlighted.
   - The boxes in the **Status** column turn red, depending on the Status.
   - When you hover over the Blocked icon, the Task that is blocking the Story will display.

   Here is what it will look like after it is blocked:

Type	Name	Blocked	Status
USER STORY	▼ Very Important User Story	BLOCKED	▣▢▢▢
DOC	Create Documentation	🛑	▣▢▢

6. When you unblock the Task, a reply is added to the original Discussion Topic, also informing team members.

The **Health and Progress** pane of the **Team Room** will also display a blocked Story.

Health & Progress

💡 **Tip:** Click on the blocked icon to filter the **Team Room** to only see the blocked Task.

Ordering Tasks

When you create a Task, it is added to the top of the list in the Story. To change the order:

1. Click ▦ **Team Room** > ☰ **Sprint List**.

2. Make sure the **Order** column is visible. See *Viewing Columns in Grids*.

3. Double-click the **Order** column in the row for the Task to enable editing.
4. Enter the new numerical position (integer) by typing it in or use the control to increase/decrease the position.
5. Click **Enter** and then **Save** (at the bottom).

Code Review

Rhythm's native Code Review functionality helps you:

- Understand code changes in context of defined Stories and Tasks.
- Find bugs early in the cycle before they reach production code.
- Keep code clean by supporting your team's coding standards.
- Review code changes from multiple SCM repositories from within a single Rhythm Story/Task.
- Compare differences in-line or side-by-side in a modern editor
- Quickly find historic code changes related to previously delivered Stories and Tasks.
- Keep team members up-to-date on changes in the code and ensure that the team is in agreement with changes.
- Organize and route reviews to specific reviewers.
- Have line-by-line discussions with mark-down notation.

Code Review Process

Use the following high-level steps (along with the *Code Review workflow diagram*) to use Rhythm's Code Review features:

Create
1. *Configure Rhythm Projects to your Git branches* ◆ *or StarTeam Views* ⭐.
2. Associate changes to a Story during a commit: *Git* or *StarTeam*.
3. Create *a Code Review* 🖥 for a Story/Task in Rhythm.
4. *Open the Code Review* in Rhythm.

Execute
1. Add *Reviewers or Watchers* as needed ⊕.
2. *Review the Code Review **Timeline*** 🕲.
3. *Observe differences in the changes* between the files in **Unified** 📄 or **Side-by-Side** ⬛ mode.
4. Have *line-by-line Discussions* as needed.
5. Receive *email notifications* for all comments or changes to the Code Review.
6. *Accept/Reject a Code Review* 👍 👎.

View Review Code Review Summary in **Team Room**.

⚠ **Important:** Before performing Code Reviews, ensure that your Rhythm system administrator has configured your SCM system to your Rhythm Project. For more information, see *SCM Configuration*.

Code Review Workflow Diagram

The following diagram depicts the best practice workflow for using the Rhythm Code Review feature.
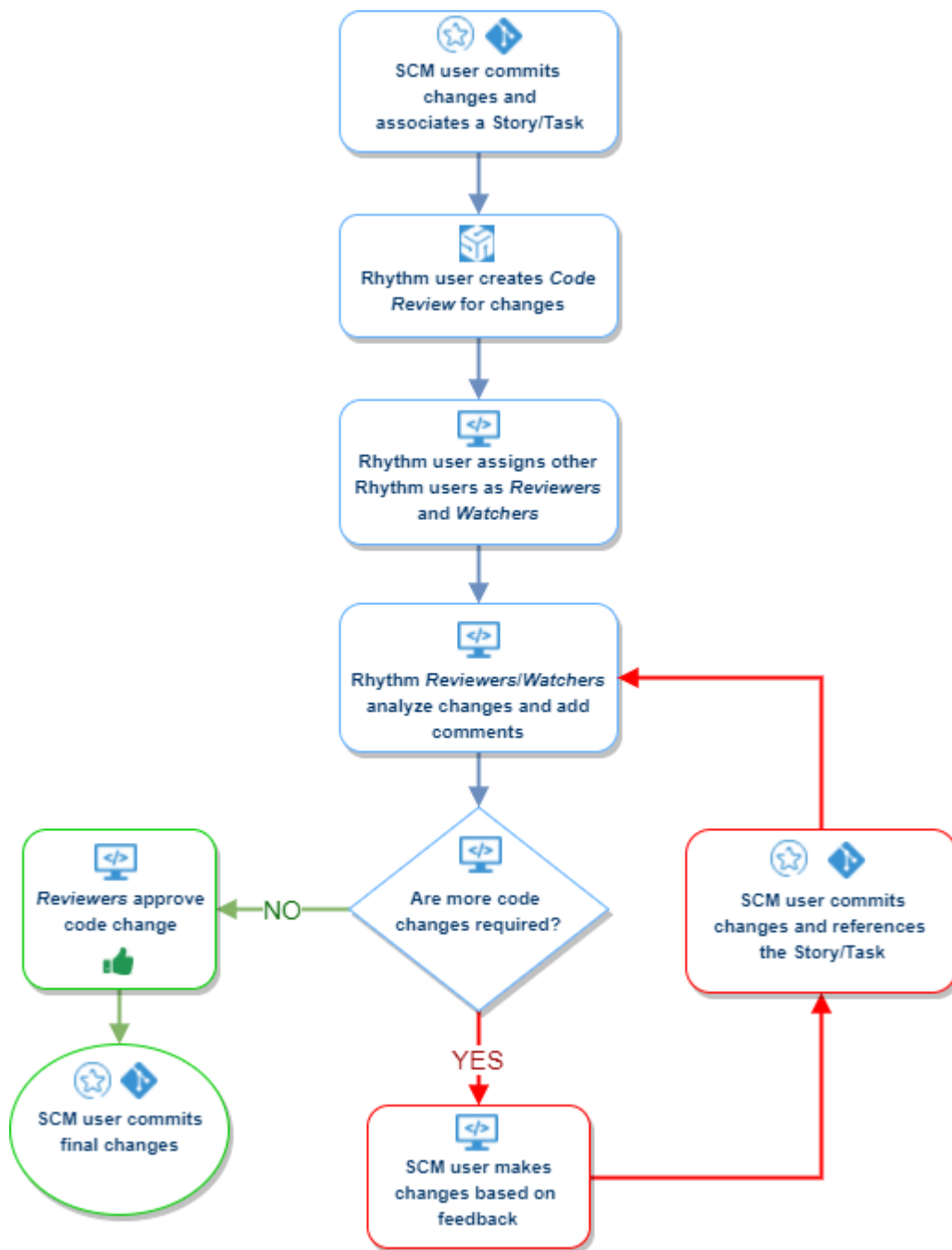
Associating StarTeam Changes to a Rhythm Story/Task

Although you can manually add change sets to a code review using the **Code Review** editor, the best practice is to associate the Story/Task with the change set during the SCM commit.

1. Create your Rhythm Story/Task.
2. Begin committing to your SCM system as you normally would.
3. When your code is ready for review, during the commit, in StarTeam select the Story or Task as a *Process Item*.
4. Perform the commit. This action will create an Atlas Hub Relationship between the change set and the Rhythm Story/Task.
5. Proceed to *Creating a Code Review.*

Associating Git Changes to a Rhythm Story/Task

Although you can manually add change sets to a code review using the **Code Review** editor, the best practice is to associate the Story/Task with the change set during the SCM commit.

1. Create your Rhythm Story/Task.

2. Begin committing to your SCM system as you normally would.

3. When your code is ready for review, during the commit, enter the following in the commit message:

Story	`Story:ID`
	For example: `Story:22356`.
Task	`Task:ID`
	For example: `Task:124456`

4. Perform the commit. This action will create an Atlas Hub Relationship between the change set and the Rhythm Story/Task.

5. Proceed to *Creating a Code Review.*

Creating a Code Review

Typically, you will want to create a Code Review after you have created a Story/Task in Rhythm and you have associated that Story/Task with a specific file or change set during your SCM commit. How to associate your commit with the Story depends on your SCM tool. To associate a Story/Task with files during a commit, depending on your SCM tool, see *StarTeam* or *Git*. If you already performed a commit and didn't associate it with a Story during the commit, see *Add Existing Commit to a Code Review*.

⚠ **Important:** Each Story/Task can contain only one Code Review.

To create a Code Review:

1. For a Story/Task in Rhythm that does not have change sets associated, click ⚙ **Action** > **Create Review**. In Scum, this menu is available on the Story or Task row in the Grid. For Kanban, this menu is available in the Story card. The following resulting card is for a Kanban Story:
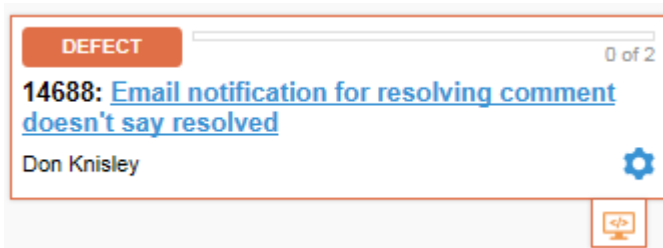
2. For a Story that already has a change set associated, click the **Change Set** icon 🔶 for the Story. The following example is from the **Sprint List** view.
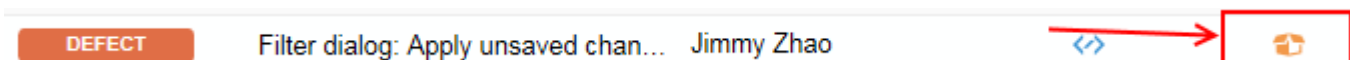
3. After you click the icon or agree to create a Code review, the **Code Review** window opens:

If you have already associated a commit with the Story, the files in the commit will appear in the **View by** list.

4. Use the *Code Review Editor* to complete the Code Review process.

💡 **Tip:** When you create a Code Review, it will appear in the *Relationships* tab of the **Story Editor**.

Understanding Code Review Icon States

The Code Review state icon provides summary Code Review information making it easy to identify when an item might warrant further action. Icon content and color change depending on related data. Here are some tips to help you understand what you will see:

- **Change Set** icon 📙: The Story/Task has related code changes, but a Code Review does not exist. Clicking on this icon will prompt the user to create a Code Review.
- **Code Review** icon: The Code Review icon is visible when a related Code Review exists. The icon color provides insight into summary status.

    - **Code Review** icon (green) 🖥: The Story has related code changes and a Code Review(s) with an accepted status.
    - **Code Review** icon (orange) 🖥: The summary status is indeterminate. This situation will arise when there are multiple related Code Reviews that are in different states. For example, this will occur when there are two related Code Reviews: one related Code Review is `Accepted`, and the other in a `Rejected` state. An amber icon will also be presented when a Code Review has been created, but there are no associated Change Sets.
    - **Code Review** icon (red) 🖥 : The story or task has a Code Review(s) with a rejected status.

The Code Review icon can be made visible in most Story and Task grids by *manually adding the Code Review* column. The Code Review icon is also visible by default on all Kanban Board cards.

Code Review Editor

The **Code Review** editor is opened from a Story/Task and contains all the features you need to perform your Code Review. You can create Code Reviews several ways:

- Click ⚙ **Action** > **Create Review** on a Story/Task.
- If a Story/Task has an associated change set but no Code Review, click the change set icon 📙 and you will be prompted if you want to start a Code Review.

Kanban	In a Kanban Board, this icon is attached to the bottom of the Story.
Scrum	For Scrum, the icon is in the **Code Review** column.

The Code Review window contains everything you need to perform a complete code review:
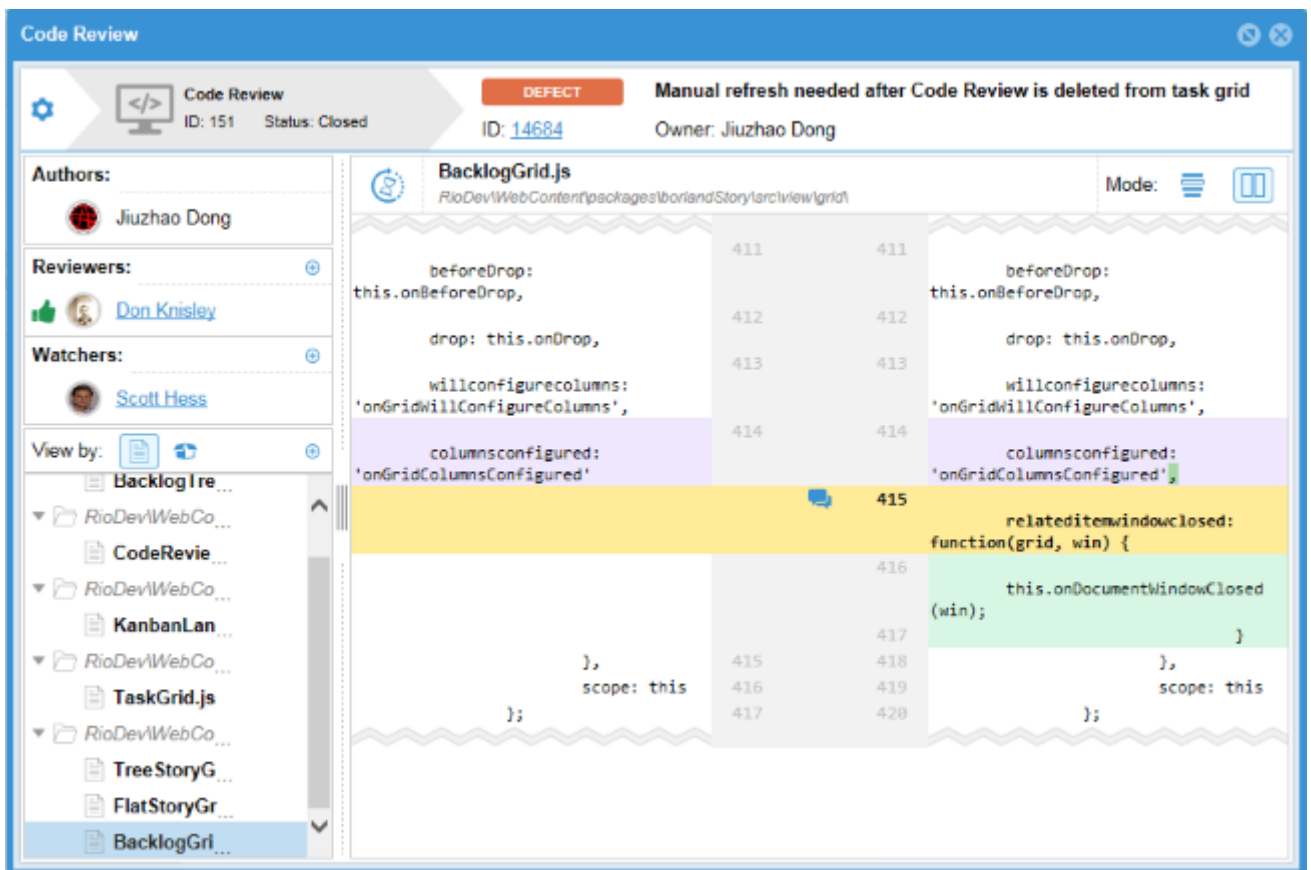
Information Pane

The top **Information Pane** of the Code Review contains the Code Review **ID** and current **Status** (`Active`, `Closed`, `Cancelled`) as well as Story's **Name**, **Type** and **Owner**.

The **Actions** icon ⚙ allows you to *change the Status* or to delete the review.

Authors Group

This list contains the user(s) who made the commit(s) in the SCM tool that are exposed in the Code Review.

Reviewers Group

Reviewers are any users that are added to the Code Review that will *comment on* and must *Approve/ Reject a Code Review*.

Click **Add** ⊕ *to add a Reviewer*.

Watchers Group

Watchers are any users that are added to the Code Review that are interested in the progress of the Code Review. Watchers are able *to add comments* to a review.

Click **Add** ⊕ *to add a Watcher*.

View By Group

The **View By** group is the list of all files contained in the Code Review. You can use either the **Files** or

Change Sets mode to select a file.

Click **Add** ⊕ *to add a Change Set or File to the Code Review*.

Selecting a file opens the **Diff** window allowing you *to review the code changes*.

Files that contain **Comment** threads are decorated like this:

Code Review Timeline

The *Timeline* shows all actions that have taken place for the Code Review since it began. The **Timeline** contains filters for **From/To**, **Type**, and **User**.

Code Review Diff Pane

When you select a file in the **View By** group, the main pane displays the file in **Diff** mode. You can use the

Unified or **Side By Side** view *to review* and *comment on the code*.

Click to return to the Code Review **Timeline**.

Assigning Reviewers and Watchers

Use the Code Review window to assign *Reviewers* and *Watchers* to a review.

Reviewers Reviewers are any users that are added to the Code Review that will *comment on* and must *Approve/Reject a Code Review*.

Watchers Watchers are any users that are added to the Code Review that are interested in the progress of the Code Review. Watchers are able *to add comments* to a review.

1. Open the **Code Review** window for the desired Story/Task.
2. In either the **Reviewers** or **Watchers** groups, click ⊕. A list of available users is displayed.
3. Select a user from the list. They are added to the group.

   💡 **Tip:** To remove a user from either list, click the user's icon.

Reviewers and *Watchers* will be notified via email when various actions occur during a Code Review. Examples: when a discussion is started within a review or when the state of the Code Review changes.

Using the Code Review Timeline

The *Timeline* shows all actions that have taken place for the Code Review since it began. The **Timeline** contains filters for **From/To**, **Type**, and **User**.

The **Timeline** displays actions within the Code Review in ascending order. You can scroll up to see previous actions, such as when the review was created.

1. To open a file that is listed in the **Timeline**, click the file name. This will close the **Timeline** pane and open the **Diff** pane.

2. To filter the **Timeline**, click:

   From Filters the **Timeline** starting on the selected date.

   To Filters the **Timeline** until the selected date.

   Type Select one of the following:

All	Removes any filters applied and shows the entire Ti.
Content Change	Displays only those entries in the Code Review that deal with file content.
User Assignment	Displays all entries dealing with users.
Status Change	Shows entries that deal with status, such as accepting changes or closing a review.
Discussion	Shows all Discussions that happened within the review.

   User Filters by the selected user.

Viewing the Files Contained within a Code Review

Use the **View By** group in the **Code Review** window to view the files contained within a Code Review.

The **View By** group is the list of all files contained in the Code Review. You can use either the **Files** or **Change Sets** mode to select a file.

1. Open the **Code Review** window for the desired Story/Task.

2. By default, the Code Review window opens in the **Timeline** mode, displaying a chronology of events within the review. To see the changes within a file, click either the **Files** or **Change Sets** buttons to view all the files within the review.

3. Select a file.

   When you select a file in the **View By** group, the main pane displays the file in **Diff** mode. You can use the **Unified** or **Side By Side** view *to review* and *comment on the code*.

Adding or Modifying a Change Set to an Existing Code Review

If you have a already committed a change to your SCM system and did not associate it to a Rhythm Story/Task, there are only a few steps to associate it with a Code Review:

1. Open the **Code Review** window for the desired Story/Task.

2. In the **View By** group in the left pane, click the **Add a Change Set** Icon ⊕. The **Add Change Set** dialog box opens.

3. Select the change set and click **OK**.

Code Review Diff Window

The **Code Review** Diff window offers two views to review the code: **Unified** and **Side-by-side**. The difference between the two modes is that **Unified** is the combined code and **Side-by-Side** has the old (*from*) on the left and has the new (*to*) on the right. Which mode you choose depends on the complexity of the changes.

Code Reviews have the following statuses: `Active`, `Closed`, `Cancelled`, and `Rejected`.

The goal of the code review is to have your Reviewers `Accept` your code changes and then you will be able to `Close` the review.

Understanding the Changes

The following color conventions are used to help you to understand the code changes:

Red   Code that was removed.

Green   Code that was added.

Purple This is an in-line change. The Code Review feature compares differences down to a single word.

If the code changes are good as-is and do not require any discussion, click **Accept** (thumb's up) icon

 in the top-right pane of the **Code Review** window.

⚠ **Important:** If any *Reviewer* `Rejects` a Code Review, the review is put into the `Rejected` state. The number of Approvals does not matter.

If there are comments required for the review, continue to *Commenting on the Code*.

Adding Comments to a Review

Rhythm Code Review allows you to create **Comments** for each line of code. Each line of code may contain multiple **Comment** threads. Each **Comment** thread can be one level deep. To create a **Comment**:

1.
Place your cursor in the *gutter* (left hand side for **Unified** or in the middle for **Side-by-Side**) until you see the blue edit icon.

2. Click the icon. A **Comment** window opens allowing you to begin your Code Review **Comment**:

3. Type your comment in the **Comment** field. Each entry must be one of the following types:

- `Comment`
- `Clean Up`
- `Code Style`
- `Bug`
- `Question`

4. During your comments, feel free to use the available *markdown syntax*.

5. When you feel that the issues in the **Comment** are understood/complete, click **Resolve** (top-right in **Comment** window) to end the comment thread.

Code Review Markdown Syntax

Rhythm uses the following markdown syntax for Code Review Comments:

Headings	`# heading 1` `## heading 2`
Bold	`**bold**`
Italic	`_italic_`
List	`* bullet 1`
Inline Code	`` `inline code` ``
Multi code	``` ``` [Enter key] your text [Enter key] ``` ```

Changing the Status of a Code Review

A Code Review can have one of the following statuses:

Active The Code Review is new or open/active and _Reviewers_/_Watchers_ are actively commenting on it.

Closed All _Reviewers_ have `Accepted` the Code Review and the _Author_ clicked ⚙ > **Status** > **Closed**. No further action is required.

Cancelled Something has gone wrong in either the check-ins or code changes and the intent of the team is to ignore this review. You can either keep it in the `Cancelled` state for historic purposes or delete it.

Rejected

When a _Reviewer_ clicks the thumb's down icon , the Code Review has a **Status** of `Rejected`. More code changes/comments/discussions are required to get the _Reviewer_ to `Approve` the review.

1. Open the **Code Review** window for the desired Story/Task.

2. Click ⚙ > **Status**.

3. Select `Active`, `Closed`, or `Cancelled`.

   📝 **Note:** As noted above, if a _Reviewer_ Rejects the Code Review, it is automatically put into the `Rejected` **Status**.

Code Review Email Notifications

During the Code Review process, _Authors_, _Reviewers_, and _Watchers_ will receive emails indicating various changes that occurred during the Code Review. These events will also be added to the Code Review's **Timeline**.

Email notifications will be sent to the Author, Reviewers and Watchers when:

- A **Comment** is added or replied to.
- A **Comment** thread was marked `Resolved`.
- A Code Review is `Accepted` or `Rejected`.
- Review **Status** is changed (`Open`, `Closed`, or `Cancelled`).
- A _Reviewer_ or _Watcher_ is added or removed.

- A Code Review is deleted.

Managing Your Capacity

For each Project in which you are a *Contributor*, you can specify your Project and Sprint level availability.

When set at the Project level, the value will be used for each Sprint. When set at the Sprint level, it's only for that particular Sprint.

This value is visible as **Total** in the **Task Owner** field in the **Team Room** and **Story Editor**:
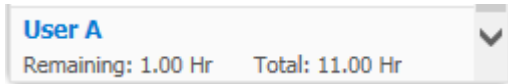
> **Note:** Administrator's can set each user's availability using the Edit Sprint dialog. For more information, see *Managing Team Capacity*.

1. In the top right corner of the application, click your **<User Name>** and select **Availability...** The **Availability** dialog box opens with a list of Projects and Sprints.
2. To set a Project-level availability, double click in the **Availability** column and type in the number of hours that will be used for each Sprint in the Project.
3. To enter hours for a certain Sprint, expand the Project node and enter the hours in the proper row.
4. Click **OK**.

When filtering on this user in the **Team Room**, the **Burndown** will contain a horizontal orange dashed line to represent the availability of the individual. For more information, see *Assigning and Estimating Tasks*.

Search

Click 🔍 (**Search**) to open the search page.

Select current project, all projects, or a specific project from the **Only show results from** list.

Type the string you want to search, and click **Search**.

Rhythm does not search common words, such as "and", "if", "else", "then", etc. These common words are referred to as *stop* words because they are used so often that the search would return too many non-specific results. To view the full list of stop words or to add your company-specific stop words, open `c:\Program Files\Micro Focus\Hub\Search\config\en_stopwords.txt`.

Rhythm gives priority to title, content, description, and snippets when matching the query to items it will display in results. All other field items are lower priority. The Item Type column displays the type of item that matched the search criteria.

To the right of each search result, there are icons to use:

Quick Look	Opens the item in an editor for you to review.
Go To	Jump to an item in your search results. If an item is in a different project, Rhythm switches your context to the other project as long as you have permission to view it.

Editing Your Profile
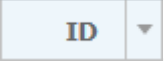
To edit your profile:

1. In the top right corner of the application, click your **<User Name>** and select **Edit Profile**. The **Edit your profile details** dialog appears.

2. Change your **Logon Name**, **Password**, **Name**, **Email Address**, or **Phone Number**.

Viewing Columns in Grids

Each perspective/view contains different grids with relevant information. There are additional columns available.

1.
   Hover over any column heading until you see an arrow .

2.
   Click the arrow to see .

3. Hover over **Columns** and you will see a list of all available columns.

4. Click each column that you want to see in the grid.

Your selections are saved for you when you return to that view.

Index