{JIRA Case: <http://jira.pentaho.com/browse/DOC-3126>}

JIRA

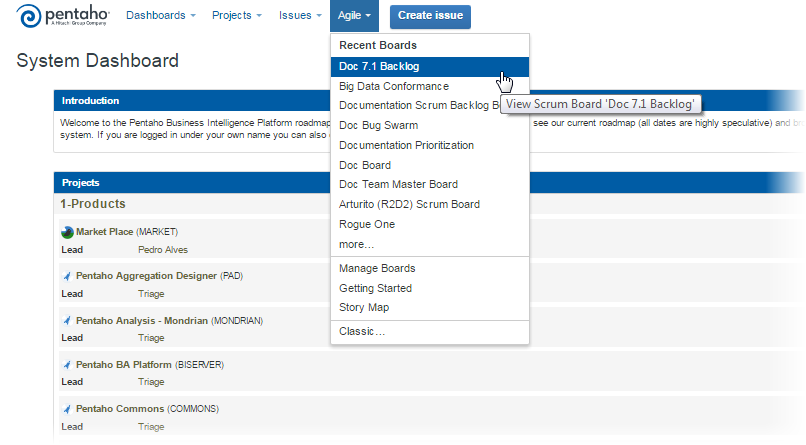
{iWiki Article: <http://iwiki.pentaho.com/display/DOC/JIRA>}

The agile scrum process at Pentaho is defined by our [Agile Playbook](http://iwiki.pentaho.com/display/ENG/Agile+Playbook). If you are not already familiar with agile scrum and JIRA, we recommend you first look through Pentaho’s Agile Playbook. This article assumes you have a basic understanding of how the scrum process works here at Pentaho.

Access JIRA for documentation work through the following URL:

* <http://jira.pentaho.com/secure/Dashboard.jspa>

This address opens a JIRA System Dashboard containing links to common boards and cases related to you. From this dashboard, use the **Agile** menu to view the current documentation backlog. The following example shows how to access the documentation backlog for 7.1:



In the Agile menu, select **more…** to view a complete list of available boards on the **Mange Boards** page.

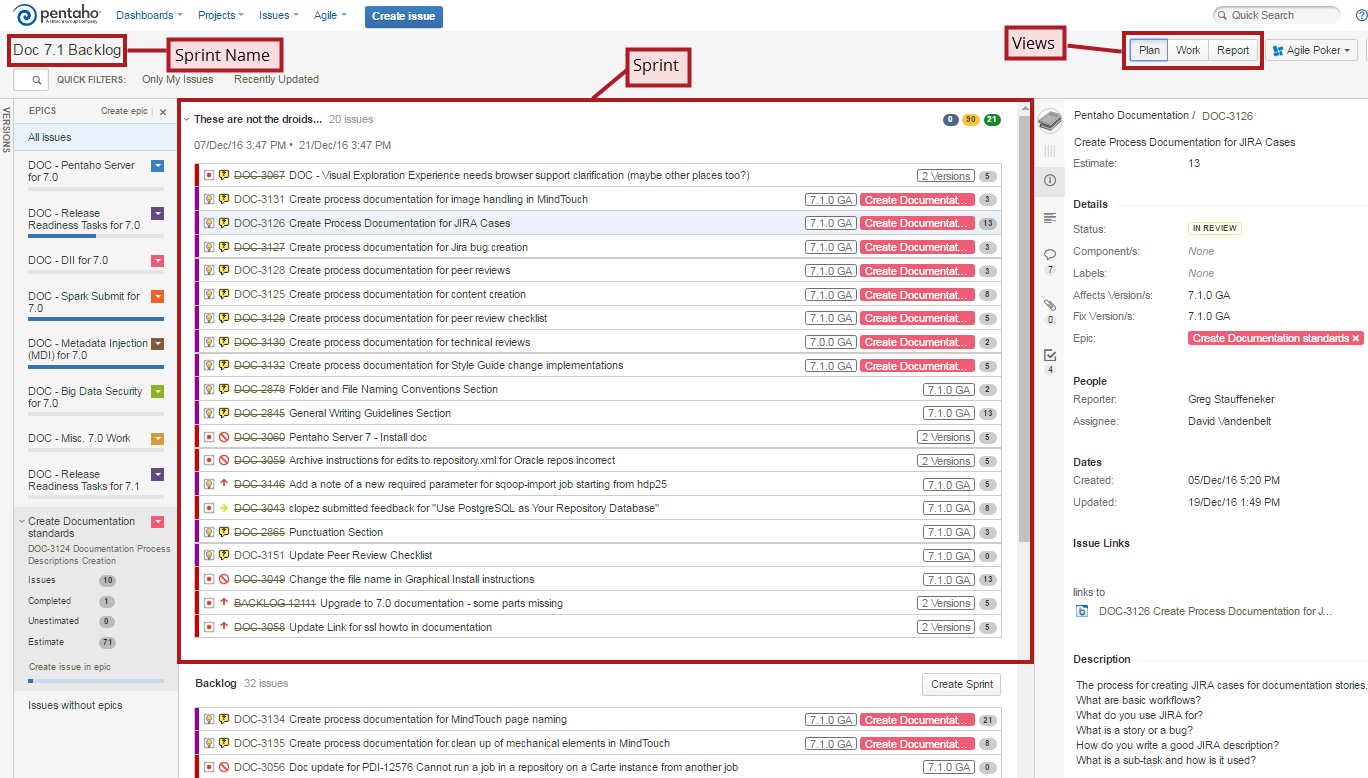
On your backlog board, you will most often use the following views:

* **Plan** is used during grooming and planning.
* **Work** is used during the sprint.

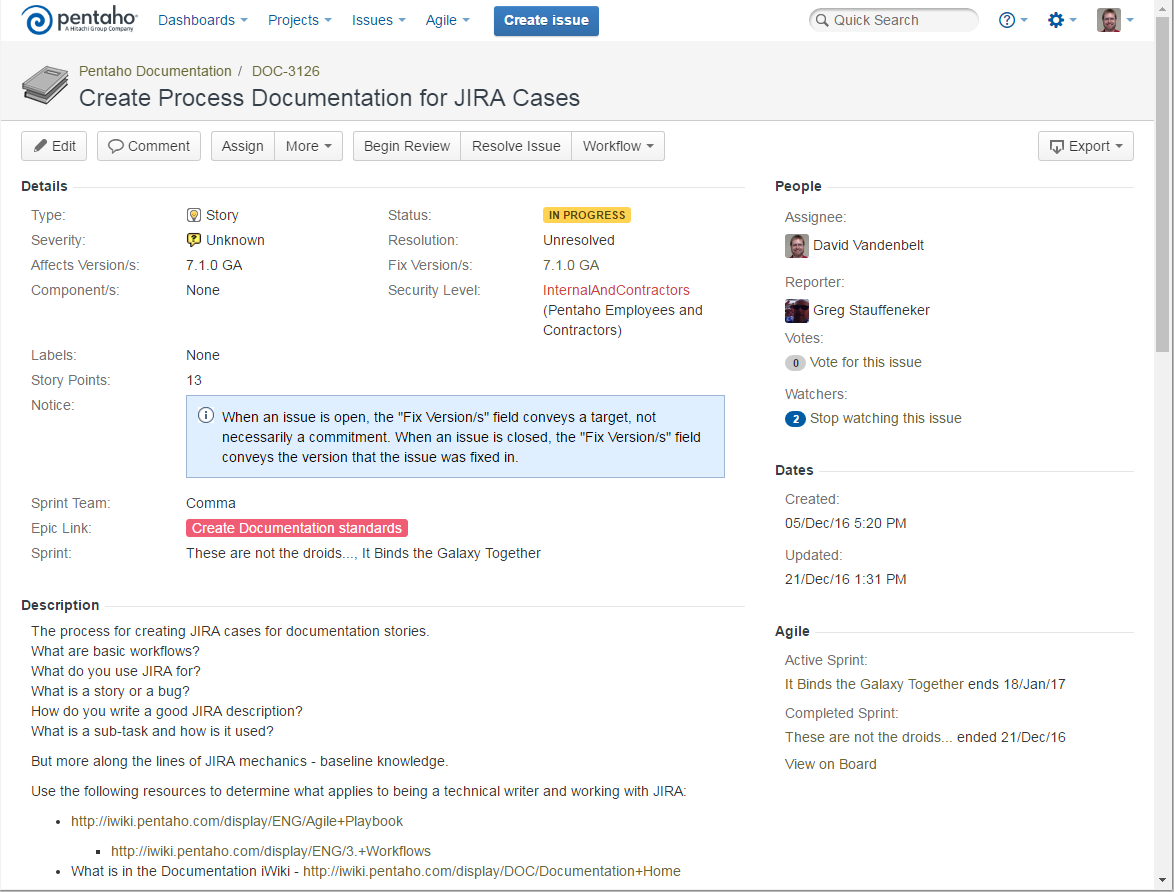
The following example shows he **Plan** and **Work** views:

|  |  |
| --- | --- |
| Plan View | Work View |
|  |  |

You should familiarize yourself with the following parts of the **Plan** view in the current documentation backlog:



The **Sprint** section shows summaries of stories, spike, and bugs being worked on by the team. JIRA represents them as individual issues because JIRA was originally a software for issue tracking. At Pentaho, we refer to these items as *cases* rather than *issues*. The following figure is an example of a case as it appears in JIRA:



As a documentation team member, you will be mainly working on cases the team selected to complete within the sprint. While working on these cases, you will be moving them through a defined JIRA workflow. This workflow is described in the [Workflow during a Sprint](#Workflow) article.

Although most spikesand stories by the product ownerand groomed by the team prior to the, you may have to create these cases How to create spikes and stories is described in the [Spike and Story Creation](#EpicAndStoryCreation) article.

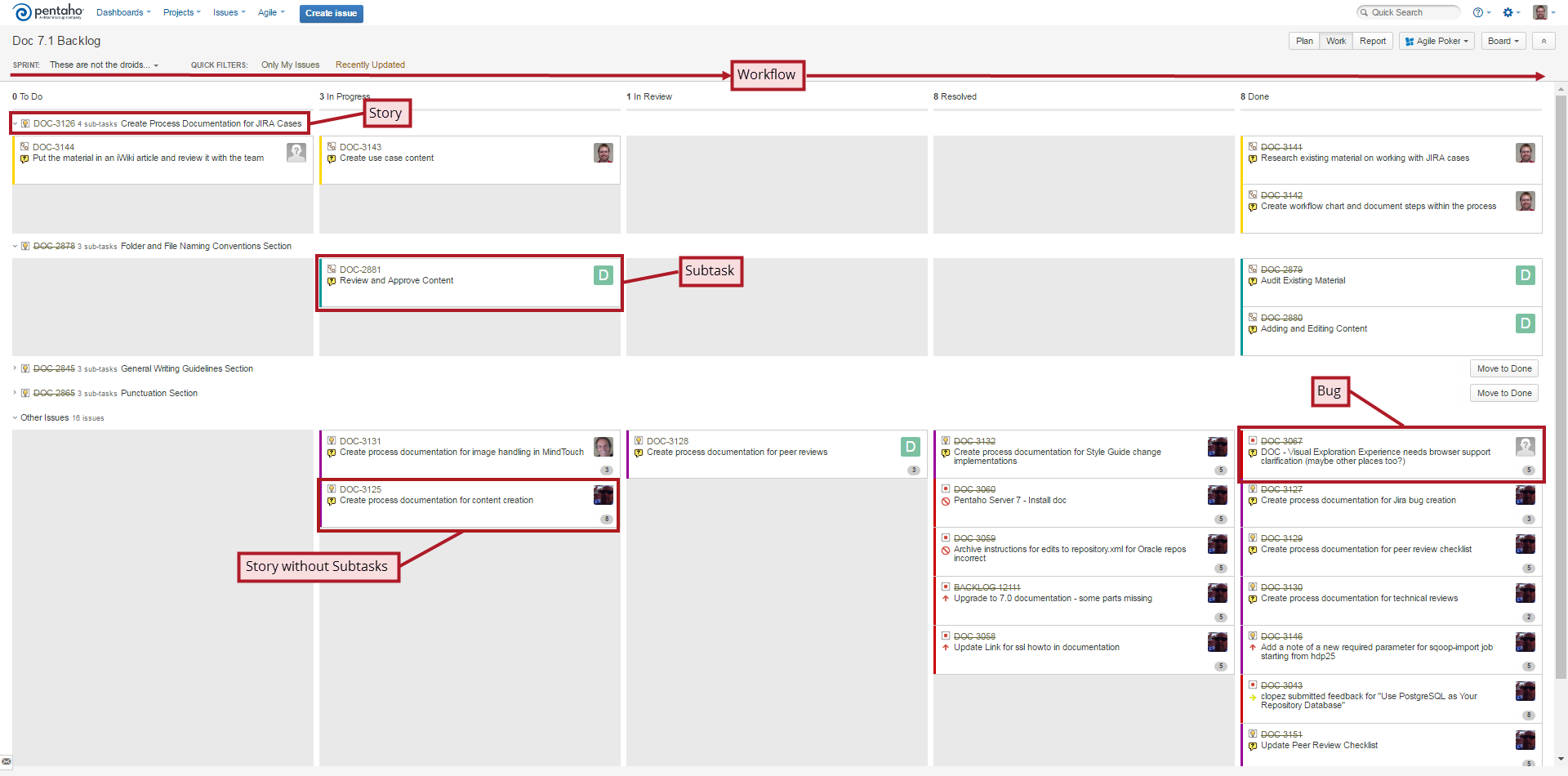
Bugs are usually created by internal and external customers. However, while working on spikes and stories in a sprint, you may find a problem within the documentation set that needs to be reported. The [Bug Creation](http://iwiki.pentaho.com/display/DOC/Bug+Creation) article describes how to create such a case.

Workflow during a Sprint

{iWiki Article: <http://iwiki.pentaho.com/display/DOC/Workflow+during+a+Sprint>}

The documentation team uses JIRA cases to track our progress during a sprint through the **Work** view. The workflow of a case moves from left (the **To Do** column) to right (the **Done** column). When you sign up for a case, you will move the case through the columns (**In Progress** and **In Review**). Once in **Resolved**, the story is inspected by the product owner who moves it to the **Done** column.

You should familiarize yourself with the following parts of the **Work** view of the current documentation backlog:



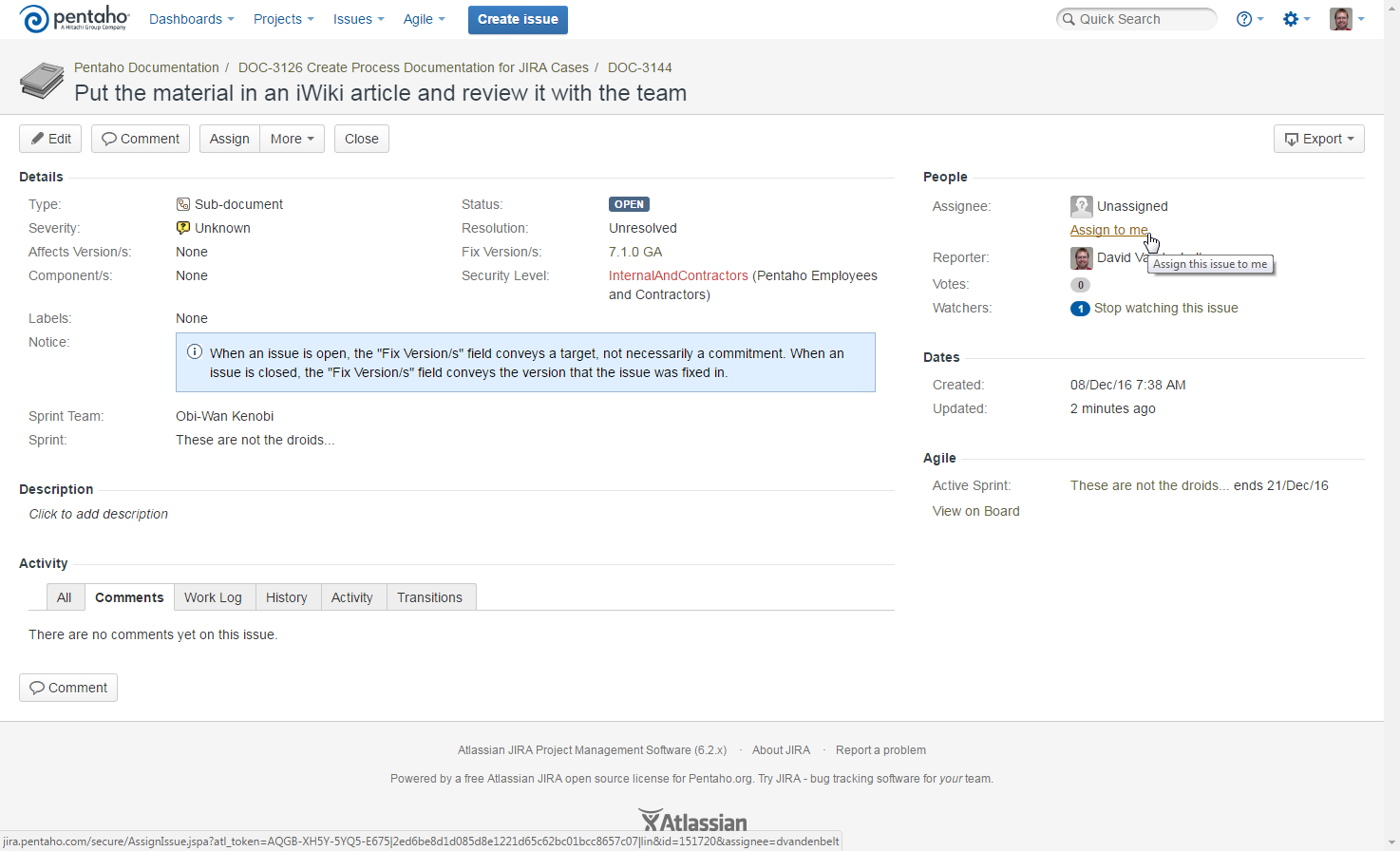
The workflow (lifecycle) of a case during a sprint is defined by the following steps in JIRA:

1. [Choosing a JIRA Case](#_Step_2:_Assigning)
2. [Adding Subtasks to a JIRA Case](#_Step_2:_Adding)
3. [Working a JIRA Case](#_Step_4:_Starting)
4. [Reviewing a JIRA Case](#_Step_4:_Reviewing_1)
5. [Resolving a JIRA Case](#_Step_5:_Resolving)

# Step 1: Choosing a JIRA Case

If you want to work on a case within the sprint, assign it to yourself by performing the following steps:

1. From either the **Plan** or **Work** JIRA view, right-click the JIRA ID link for your case.
2. From the context menu that displays, select to open the case in a new tab or window. The case appears in the new tab or window, as shown below:



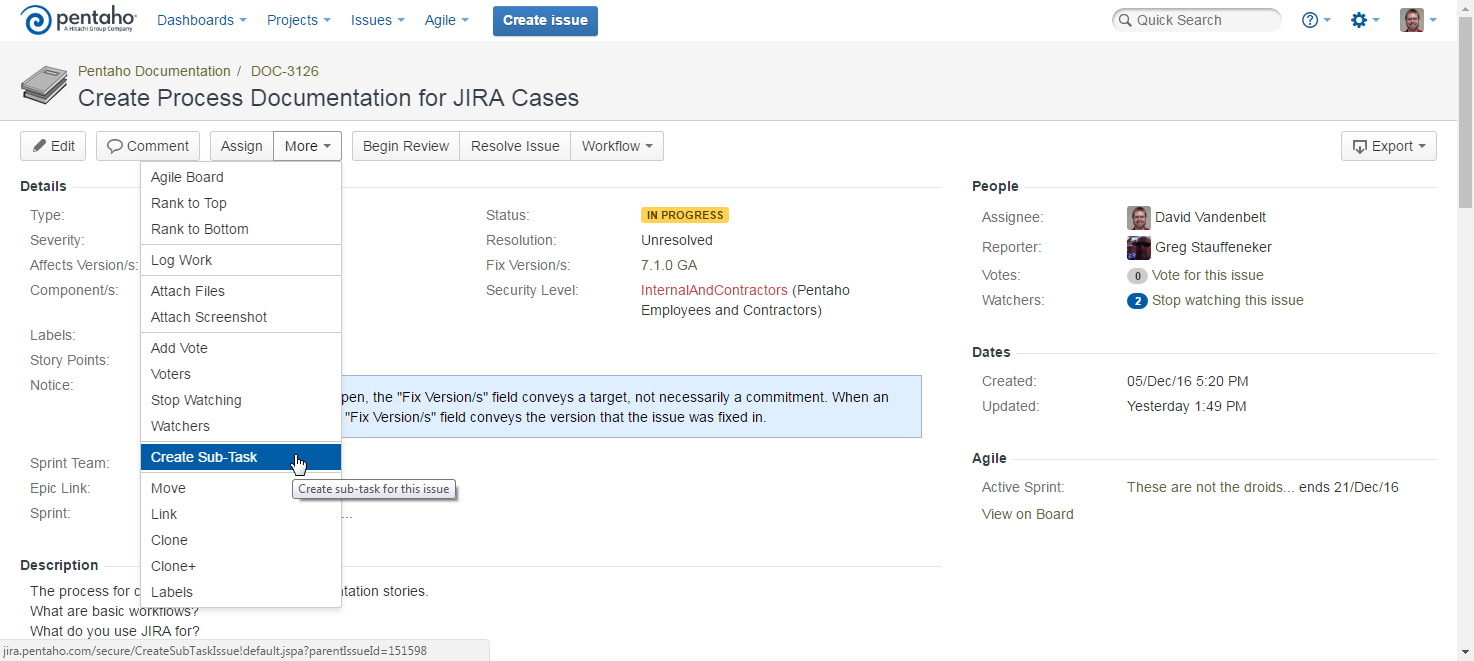
1. Click the **Assign to me** link under the People section near the upper right corner to assign yourself to the case. The **Assignee** will change from **Unassigned** to your name.

You are now the “owner” of the case. Typically, you would only assign yourself to one case at a time.

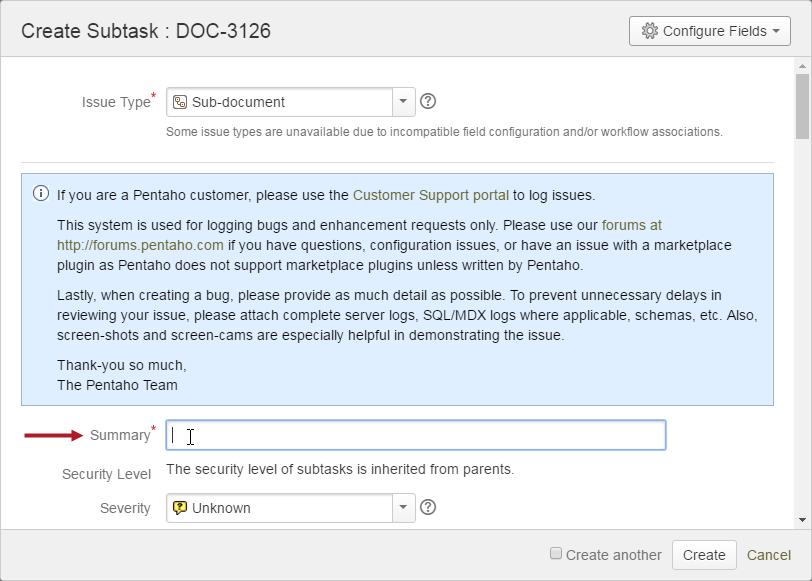
# Step 2: Adding Subtasks to a JIRA Case

Some stories may be broken down into subtasks. This is useful for working a story in clearly defined stages, or for multiple writers to work on a large story. You can also add subtasks to a story as you deem necessary.

1. Open the story, spike, or bug to which you are adding subtask.
2. Select **Create Sub-Task** under the **More** menu as shown in the following example:

The **Create Subtask** dialog box will appear.

1. Enter the title of your case in the **Summary** field, which is shown in the following figure:



Subtasks are also individual cases within JIRA. The **Summary** text will appear at the top when anyone opens the case. This text is also the description of the subtask in the **Work** view.

1. Set **Fix Version/s** to the release you are currently working towards and **Sprint Team** to **Obi-Wan Kenobi**. For example, if the next release is 7.0, set **Fix Version/s** to **7.1.0 GA**. The documentation backlog for sprinting displays cases based on a filter. The two major parameters of this filter are the **Fix Version/s** and **Sprint Team** fields.
2. Enter any useful information you might have about the subtask in the **Description** field. While the Description field is not required, it is useful for other writers and reviewers involved in the case.
3. Click **Create**. The subtasks will appear in a new Sub-Tasks section of parent case, which is the original story or bug. Optionally, if you are creating more than one subtask, click **Create another**. Selecting this option will automatically fill the **Fix Version/s** and **Sprint Team** fields with the values you specified for this instance.

Subtasks can be tracked through the parent case or the **Work** view. Each subtask has subtask its own JIRA ID. You can assign yourself to a subtask and work it the same way you would with any other JIRA case.

# Step 3: Working a JIRA Case

After assigning yourself to a case, click **Begin Work** for a story or subtask, or click **Start Progress** for a bug. Your case is now **In Progress**. If the case is a subtask, it will move from the **To Do** column to the **In Progress** column within the **Work** view as shown in the following example:

|  |  |
| --- | --- |
|  |  |

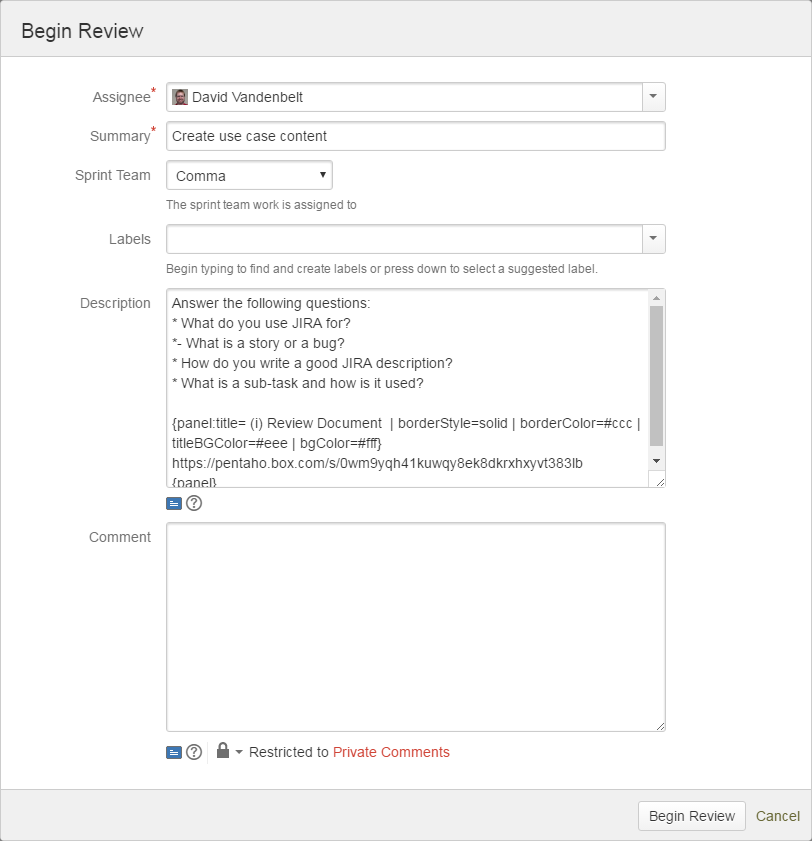
The same will occur for a parent case if it does not contain any subtasks.

While working on the case, use the **Comment** button to **Add** comments about the progress of this case. If you have any correspondence (via e-mail or Slack) with someone regarding the case, copy it into a comment.

If work stops on a case for an extended period of time (more than a day away from the case), **Stop Progress** on the case to move it back to **Open** status.

# Step 4: Reviewing a JIRA Case

When you are ready to review the content of a case, perform the following steps:

1. Open your case.
2. If the case is a story, click **Begin Review**. If it is a bug or a subtask, click **Workflow** > **Begin Review**. The **Begin Review** dialog box will appears as shown in the following example:
3. Explain how the review process is going to occur within the **Comment** field of the **Begin Review** dialog box will appear. If you have sent an e-mail, invitation, or a Slack message to a reviewer, copy that text into the **Comment** field. This text will appear under the **Comments** tab in the Activity section of the case.
4. Click **Begin Review**.

The case moves from the **In Progress** column to the **In Review** column within the **Work** view.

Subtasks may not require a review. See [Peer Review](http://iwiki.pentaho.com/display/DOC/Peer+Reviews) and [Technical Review](http://iwiki.pentaho.com/display/DOC/Technical+Reviews) process articles to determine if your subtask requires a review. For completed subtasks not requiring a review, click **Close** instead of **Workflow** > **Begin Review**.

# Step 5: Resolving a JIRA Case

When reviews are complete, perform the following steps:

1. Open your case.
2. If the case is a story or bug, resolve it by clicking **Resolve Issue**. If the case is a subtask, resolve it by clicking **Close Issue**. The **Resolve Issue** dialog box will appear. This dialog box is very similar to the **Begin Review** dialog box.
3. Explain how the case was resolved within the **Comment** field of the **Resolve Issue** dialog box. If you have received an e-mail or a Slack message about the resolution of the case, copy that text into the **Comment** field.
4. Click **Resolve Issue**.

If the case is a subtask, the **Status** will change to **Closed** and the case will move in the **Work** view to the **Done** column.

If the case is a story or a bug, the **Status** will change to **Resolved** and the **Assignee** will change to **Unassigned**. It will also move from the **In Review** column to the **Resolved** column within the **Work** view. The product owner closes a story or bug after a final review. Assign the case to the product owner (the technical communications manager) for the final review.

# JIRA Workflow Buttons

The following table is a quick reference of the most commonly-used workflow buttons:

| Button | Description |
| --- | --- |
| Begin Work/Start Progress | Moves a case from the **ToDo** column to the **In Progress** column within the **Work** view indicating Work has [started](#_Step_4:_Starting) the case. |
| Begin Review | Moves a case from the **In Progress** column to the **In Review** column within the **Work** view indicating the [review](#_Step_5:_Reviewing) of content associated with the case has started. |
| Resolve Issue | Moves a case from the **In Review** column to the **Resolved** column with the **Work** view indicating the case has been [resolved](#_Step_6:_Resolving) and is ready for final review. |
| Close/Close Issue | Moves a case to the **Done** column within the **Work** view indicating the case is complete. |

The following table is a quick reference of other possible workflow buttons you might encounter:

| Button | Description |
| --- | --- |
| Reopen Issue | Returns the closed case to an **Open** or **Reopened** state. Either state can be treated as if the case was in its original **Open** status. At this point, you can [assign](#_Step_2:_Assigning) yourself to the case again and [start your progress](#_Step_4:_Starting). |
| Blocked | Indicates progress on the case is blocked and work cannot proceed. |
| Stop Progress | Returns the case back to an **Open** state. The case was in progress, but you had to switch to something else. If you do not plan to return to the case within a day, stop its progress to return it to an **Open** state for someone else to sign up for it. |
| Check In | Checks any code into the source tree to resolve this case. It is for developers to indicate they have modified the source code. It is not intended for use by the documentation team. |
| De-dupe to me | Do not use. It is not intended for use by the documentation team. |

Spike and Story Creation

{iWiki Article: <http://iwiki.pentaho.com/display/DOC/Epic+and+Story+Creation>}

Before a sprint begins, the product owner creates epics. Epics define overall features or functionality. Based on these epics, the product owner and team create stories describing how the team will go about doing the work needed to provide these features or functionality.

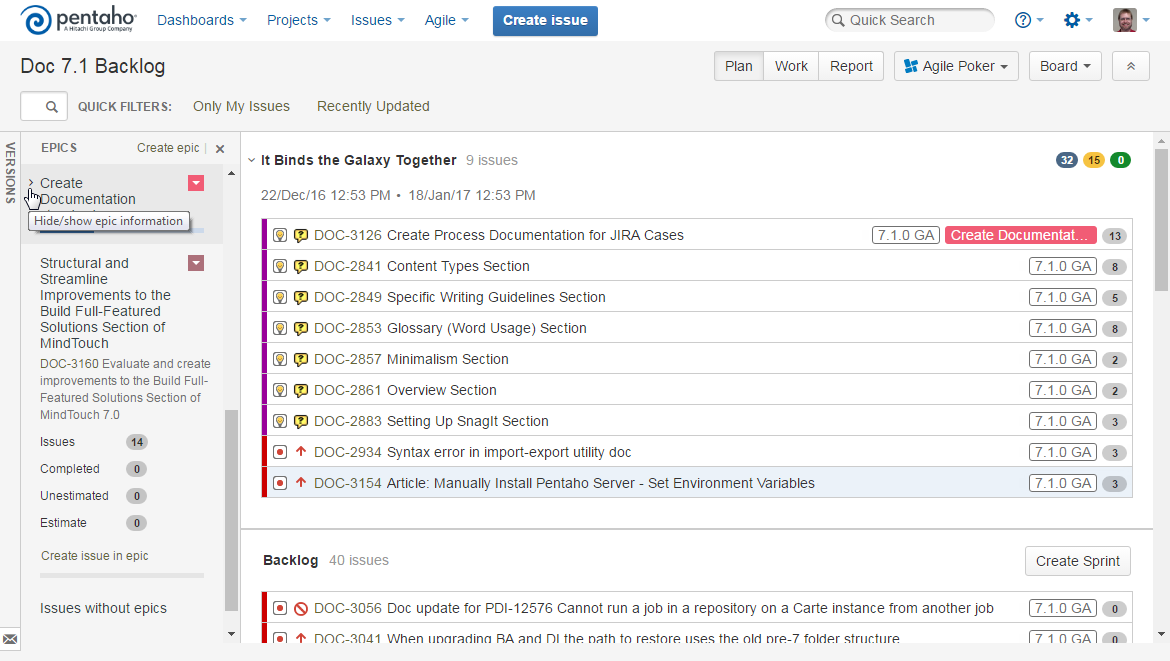
While grooming an epic, the product owner and team may agree to further research what is needed to complete that work. This research is defined in a spike. The result of a spike may be stories added to the backlog. As a documentation team member, you might be tasked with establishing a spike and creating its related stories

Spikes and stories are created through either an epic or the **Create issue** button in the JIRA toolbar. If you want to associate your spikes and stories with an epic, create them through the epic and not the **Create issue** button.



Perform the following steps to create a spike or story in JIRA:

1. If you are creating a spike or story through an epic, move your cursor over your epic listed in the EPICS panel. A **Hide/show epic information** arrow will appear.
2. Use the arrow to show more information about the epic as shown in the following example:



1. Click **Create issue in epic**. After clicking the link, the **Create Issue** dialog box appears.
2. Fill in the following fields:

| Field | Description |
| --- | --- |
| Issue Type | Whether the case you are creating as a **Spike** or a **Story**. |
| Summary | The title of the spike or story as it appears at the top of the case. Also, how it appears in the **Plan** and **Work** views. |
| Sprint Team | Set **Sprint Team** to **Obi-Wan Kenobi**. The documentation backlog uses a filter to display cases. The one parameter of this filter is the **Sprint Team** field.  ***Note****: You select values for this field through a closed dropdown menu. You can start typing to find a specific value, but you must click on it in the menu to select it.* |
| Fix Version/s | Set **Fix Version/s** to the release you are currently working towards. For example, if the next release is 8.0, set **Fix Version/s** to **8.0.0 GA**. The documentation backlog uses a filter to display cases. The other parameter of this filter is the **Fix Version/s** field.  ***Note****: You select values for this field through a closed dropdown menu. You can start typing to find a specific value, but you must click on it in the menu to select it.* |
| Use Case | A description of how the reader would use the feature associated with this spike or story. |
| Description | A definition of what needs to be done to prove the case has been completed. At Pentaho, we refer to this definition as acceptance criteria. If you do not fully know the acceptance criteria when you originally create the case, provide your best estimate. The specifics of the acceptance criteria will be fleshed out during [grooming](http://iwiki.pentaho.com/display/ENG/3.+Grooming+the+Backlog).  ***Note****: Always provide at least your best guess at acceptance criteria. Do not leave the* ***Description*** *field blank. A spike and story without a description is useless.* |

1. If you are creating more than one case, click **Create another**.
2. Click **Create**. The new case will appear in the backlog section of the **Plan** view in JIRA.

With these cases now in the backlog, the team can go about doing [this work through their sprints](http://iwiki.pentaho.com/display/DOC/Workflow+during+a+Sprint).