JIRA SOFTWARE

for

Universal Parks Technology

Standard Operating Procedures

Revision Log

This Revision Log tracks all revisions to this document. When a revision occurs, the details are logged below.

DOCUMENT HISTORY					
Version Number	*Date*	*CR ID#*	*Description of Change*	*Author*	*Amended Pages*
1.0	2019-03-09				
1.1	2019-04-10		Update to Overhead Time Tracking	LHoffar	
1.2	2019-06-05		Added section for QA	AJackson	

Table of Contents

1 Welcome to Jira

WELCOME TO JIRA!

We are excited to implement Jira software at Universal Orlando. Before we begin, let’s outlines roles, responsibilities and program expectations.

1.1 Roles and Responsibilities

- Release Train Engineers (RTEs)
  - Responsible for overall project administration
  - Add/remove users to UPT Jira project
  - Create/Start/Stop Sprints
  - Add/Edit/Remove Components
  - Add/Edit/Remove Releases
- NBCU Jira Administrators
  - Changing values in drop down lists for fields
  - Providing licenses to users
- Scrum Masters
  - Administrators on Team Scrum Board
  - Can configure Team Board
- Product Owners
  - Can create all work types – only people who can create Initiative and Epic work items
  - Responsible for creating Initiatives and Epics
- Everyone
  - Can create & edit Stories, Bugs, Risks, Issues, Test Cases, Sub-tasks
  - Can move work items through workflows
  - Log time

1.2 Program Expectations

1.2.1 Sprint Administration

- Sprints will be two weeks long
- Sprints will begin on a Tuesday and end on a Monday
- Sprints will be closed on Monday nights
- New sprints will begin on Tuesday evenings
- Only Project Administrators can create and start sprints
- All teams need to complete sprint planning by end of day Tuesday
  - All Stories and Bugs added to the sprint
  - All Stories and Bugs estimated with Story Points
- All sprint names will be “UPT Sprint #.#” to help differentiate from sprints in other Jira projects

1.2.2 Time Tracking

All of your time should be tracked in Jira via work logs. See the 9Time Tracking section below for how to add/edit/remove a work log.

- For DEP labor reporting
    - Full week labor reporting is mandatory – including “admin” work (work not tied to a charter) and PTO (or other time off)
    - All DEP labor reporting is mandatory for <u>Senior Manager</u> and below.
        - This is mandatory for the following:
            - Those who are funded by DEP (regardless of department)
            - Those who are not funded by DEP but ARE a member of Parks Technology
        - This is not required for
            - Those who are not funded by DEP and are NOT a member of Parks Technology
- For USJ labor reporting
    - This is a new labor tracking request. This is required for Director level and below.

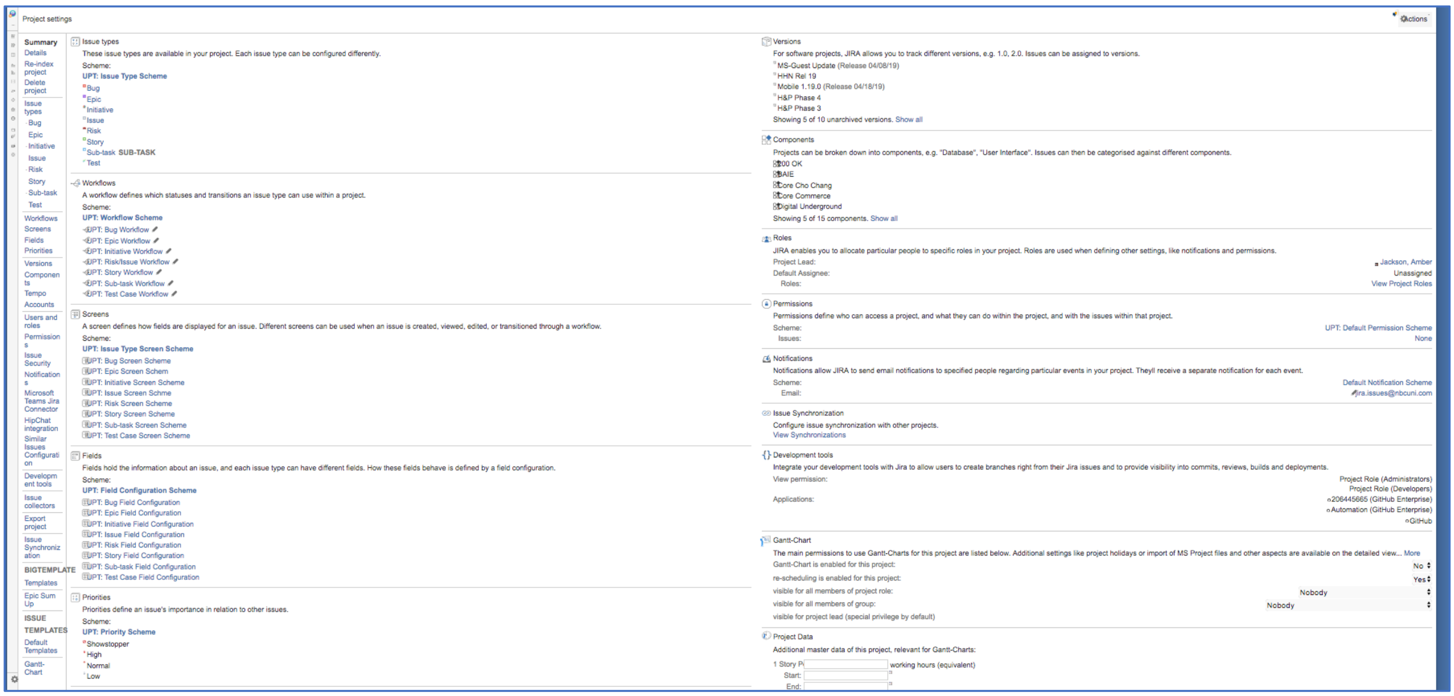
2 Jira Project

A **Jira project** is a collection of issues. A team could use a **Jira project** to coordinate the development of a product, track a **project**, manage a help desk, and more, depending on their requirements. A **Jira project** can also be configured and customized to suit the needs of you and your team.
New Jira projects can only be created by NBCU Jira Administrators (superusers).

For each Jira Project, you can decide what you want to use as:

- Issue Types
- Workflows
- System and Custom Fields
- Permission Groups
- Roles

UPT will use one unique project: **UPT Global Project**

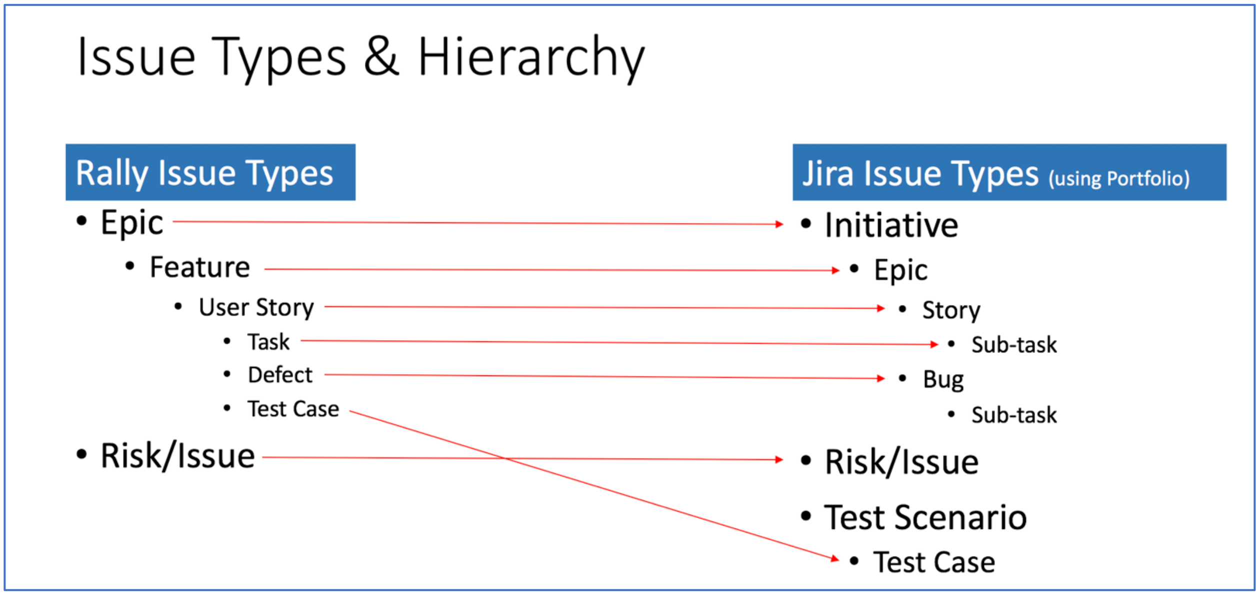


Jira is customized software. Screenshots are for training purposes only, and will vary in production based on the ongoing needs of UPT.

3 Main Notions

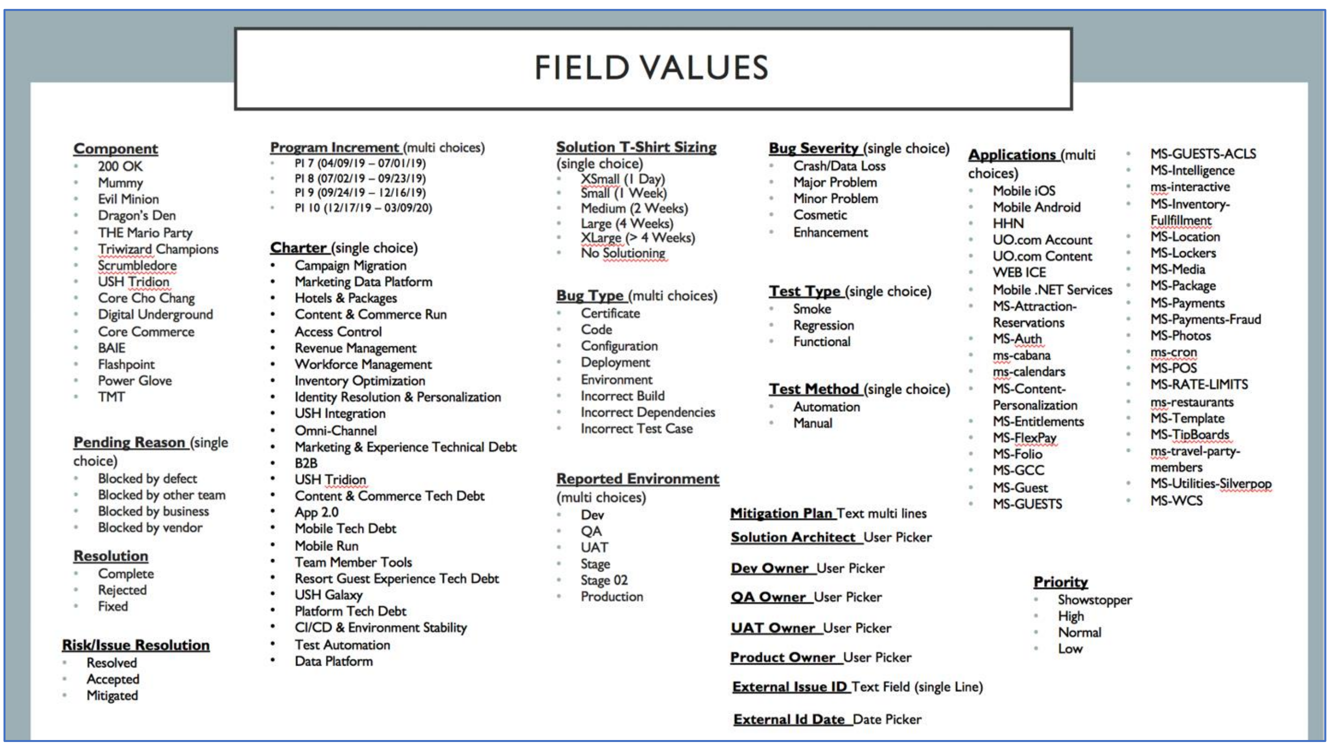
3.1 Issue Types in UPT Project

It is very important to understand the Jira hierarchy and vocabulary. The below chart displays the Rally equivalent of the new terminology. An **Initiative** spans several program increments, an **Epic** should span one program increment, and a **Story** should be completed in sprint.



3.2 Fields used in UPT Project

Several custom fields have been created or updated specifically for the UPT Global Project.



Jira Name	Rally Equivalent	Definition/Purpose
Initiative	Epic	An issue type (work item type) that represents a large effort, typically spanning multiple program increments. This work item represents the top level of the work item hierarchy used by UPT. Children of Initiatives are Epics.
Epic	Feature	A work item type that represents a medium effort, typically spanning multiple sprints but fitting within a single program increment. This work item represents the second level of the work item hierarchy. Children of Epics are Stories and Bugs.
Story	User Story	A work item type that represents a small effort, typically spanning a few days and fitting within a single sprint. This work item represents the third level of the work item hierarchy. Children of Stories are Sub-tasks.
Bug	Defect	A work item type that represents a problem found in previously completed work. Children of Bugs are Sub-tasks.
Sub-task	Task	A work item type that represents the smallest effort, typically spanning a few days. This work item represents the fourth and lowest level of the work item hierarchy.
Risk	Risk	A work item type that represents something that could happen that will negatively impact the program or team's ability to complete their work on time. If this "something" happens, then the Risk has been realized and is now considered an Issue.
Issue		A work item type that represents a Risk that has been realized.
Test Case	Test Case	A work item type that represents the steps to be taken to validate a requirement has been implemented correctly. It includes any preconditions necessary to set up the procedure and any expected results for each step in the procedure.
Project	Project	A container for work items. Unlike Rally, projects cannot be nested in Jira, so we have a single project for all of UPT. Project is no longer a way to assign a scrum team to a piece of work.
Component		Field that exists on all work item types to represent the scrum team(s) that will be working on the item. Multiple values can be selected. All team scrum boards are built off this field.
Charter		Field that represents the charter that the work item falls under. This field exists on all work item types so that logged time can be mapped to the corresponding charter for finance reports.
Reporter	Author	A field that represents the person who entered the work item into Jira.
Summary	Title/Name	A field in which to name the work item
Acceptance Criteria		Free text field to enter acceptance criteria for the work item
Labels	Tags	A field for adding additional attributes to a work item. Acts kind of like a free text field – if you type in a new label that doesn't already exist, you will create that new label. Acts kind of like a list field – if you type in a label that already exists, you'll see the existing option. HOWEVER – if you enter a label with a typo, or use a capital letter when the existing version is lower case, you will end up with a new label.

Jira Name	Rally Equivalent	Definition/Purpose
Linked Issues		Kind of like Dependencies in Rally, but not limited to this type of relationship. Consists of two fields: "Linked Issues" is a drop down list where you select the link type; ex: blocks, is dependent on. "Issue" is the field where you enter the issue key that goes with the link type. Need to pay attention to direction of the link – "Linked Issues" drop down will have both directions for a given pair.
Assignee	Owner	This field represents the person responsible for ensuring the work item is completed through the workflow.
Program Increment	Release	This field represents the Program Increment in which the team intends to do the work item. Larger items, like Initiatives and possibly Epics, could span multiple PIs, so this field is a multi-select.
Release	Milestone	In Jira, Releases are created at the project level by the Project Lead, and are then available for selection in related fields: Fix Version/s, Affects Version/s
Fix Version/s	Milestone Fixed in Build	This field represents the Release that contains the work item. You can select multiple Releases for this field.
Affects Version/s	Found in Build	This field represents the Release in which a Bug was found. You can select multiple Releases for this field. This field is only used for Bugs.
Applications	Location Platform Tags (for Test Cases)	This field represents the different applications/products that DEP builds. Examples: Mobile iOS, Mobile Android, HHN, UO.com Account, UO.com Content, WEB ICE, Mobile .NET Services, MS-Attraction-Reservations, MS-Auth, ms-cabana, ms-calendars. You can select multiple options for this field.
Epic Name		When creating a new Epic, the Epic Name custom field is a short name field used to identify the Epic. By default it is available only to Epic Issue Type. It is intended to be used to label any issues that belong to the Epic.
Epic Link		When creating a Story, Bug, Risk, Issue, or Test Case, the Epic Link field will be available (required in some cases). This field is where you select the Epic Name for the parent epic that "owns" the story, bug, risk, issue, or test case.
Parent Link	Parent	In Jira, this field is activated for Epics only, and displays the Initiative that "owns" the epic.
Solution Architect	Solution Architect	The main Architect assigned to complete the solutioning work for the epic. Required when moving an epic into the "Solution Design" state.
URL	Solution Document	The URL for the design page in Confluence for the solution documentation.
T-Shirt Sizing	Solution Sizing	The estimate for the effort needed to complete the solutioning process. Required when moving an epic into the "Solution Design" state.
Dev Owner	Dev Owner	Field for Stories and Bugs only. Represents the developer responsible for the code work of the story/bug.
QA Owner	QA Owner	Field for Stories and Bugs only. Represents the QA person responsible for generating the Test Cases for the story/bug.
UAT Owner	Biz Owner	Field for Stories and Bugs only. Represents the person responsible for ensuring Business testing in UAT is completed for the story/bug.
Product Owner	Owner (for Epics and Features)	Field for Stories and Bugs only. Represents the Product Owner for the Epic that the story/bug is linked to.

Jira Name	Rally Equivalent	Definition/Purpose
Story Points	Story Points	Story points are an abstract estimate of a work item's complexity relative to the simplest work item that that team. Story points are required when beginning work on a story. This field needs to be non-zero prior to a sprint being started so that the sprint burndown chart is useful.
Original Estimate	Task Est.	This field is intended to be an estimate for the amount of time a work item is expected to take. If a team chooses to use this field, it should be filled out at the lowest level of the hierarchy – ex: if the scrum team breaks stories down into subtasks, then the original estimate should be filled out at the subtask level only.
Remaining Estimate	To Do	This field is intended to be an estimate for the amount of remaining time a work item might take. When a work item is initially estimated, the "Remaining Estimate" field will automatically match the "Original Estimate" unless the user enters different values. As a user logs time against the work item, the "Remaining Estimate" will decrement by the amount of time logged. The user can still adjust the "Remaining Estimate" to a different value if they need to. If a team chooses to use this field, it should be filled out at the lowest level of the hierarchy.
Sprint	Iteration	This field represents the Sprint in which the team intends to do the work item. If the sprint is closed with the work item still in progress, the now-closed sprint becomes a permanent value on the work item. As a work item rolls over from sprint to sprint, you will see all the sprints it was in.
External Issue ID	ServiceNow Ticket Number Workamajig Number	A free text field for pasting a ticket number from a different system that generated the story or bug in Jira. Goes with "External ID Date" field.
External ID Date	ServiceNow Ticket Submitted	A date picker field to reflect when the External Issue was opened. Goes with "External Issue ID" field.
Pending Reason	Blocked Reason	Field for indicating why the work item is Blocked. In Jira, there are two fields: "Pending Reason" – a drop down list with high level reasons; "Reason" – a free text field to add notes or details.
Resolution		Field represents why the Initiative, Epic, Story, Bug, or Subtask is being closed. Ex: Fixed, Duplicate, Rejected, Will Not Fix
Bug Type	Defect Category	Field represents the general type of bug that was found. This field should be updated if the initial assessment ends up being wrong, ex: Bug was initially written as an Environment issue, but further triage reveals that it's a code issue.
Environment	Environment	Field represents the environment in which the bug was found, e.g. Dev, QA, etc.
Bug Severity	Severity	Field to represent how bad the issue is to the system functionality or to the end user's experience. Field is only applicable to Bugs.
Priority	Priority	Field to represent how soon the bug needs to be fixed. Field is only applicable to Bugs, but since this is a system field in Jira it will appear on all work items. It is only editable for Bugs, though.
Probability	Probability	Field is applicable to only Risks. Field represents the probability that the risk will be realized.
Impact	Impact	Field is applicable to only Risks and Issues. Field represents the impact to the program/team that is expected if the Risk is realized (or if the Risk has been realized and is now an Issue).

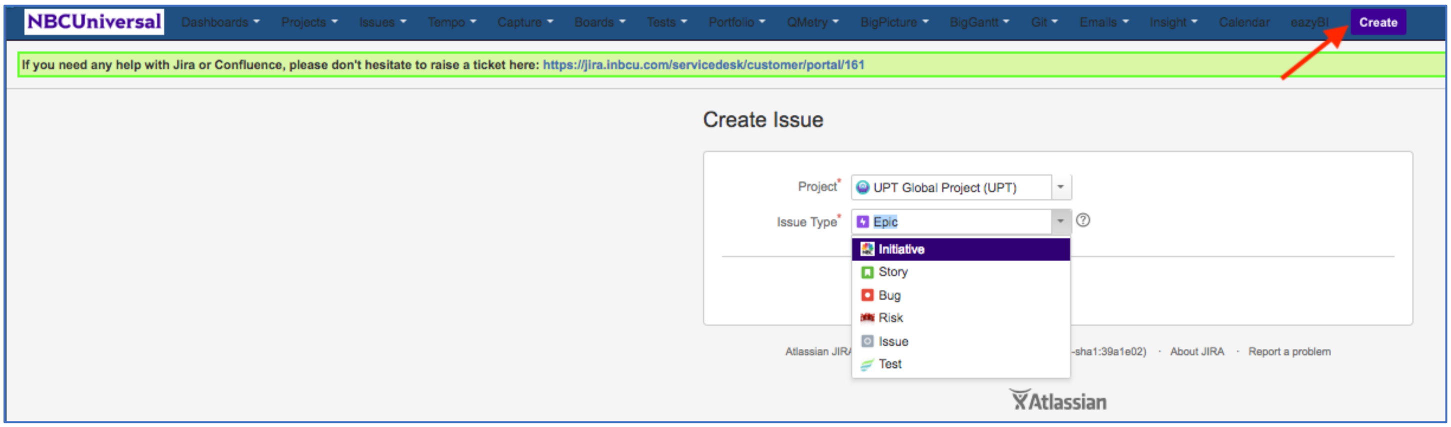
Jira Name	Rally Equivalent	Definition/Purpose
Mitigation Plan		Field is applicable to only Risks and Issues. Corresponds to the “Resolution or Mitigation” portion of the Risk/Issue template in Rally. Represents the actions that can be taken to mitigate the Risk or Issue.
Risk/Issue Resolution	Response	Field only applicable to Risks and Issues. Represents how the Risk or Issue was addressed: • Accepted (i.e. we didn’t change anything and accepted whatever may happen) • Mitigated (i.e. we made changes and have reduced the probability and/or impact of the risk/issue such that we can live with whatever happens) • Resolved (i.e. we didn’t change anything and the risk/issue resolved itself)
Test Type	Type	Field only applicable to Test Cases. Represents the type of test case: Smoke, Regression, Functional.
Test Method	Method	Field only applicable to Test Cases. Represents the method of testing: Automated, Manual.
Preconditions	Pre-conditions	Field only applicable to Test Cases. Free text field representing the preconditions needed before executing the Test Case.

4 Create Issues

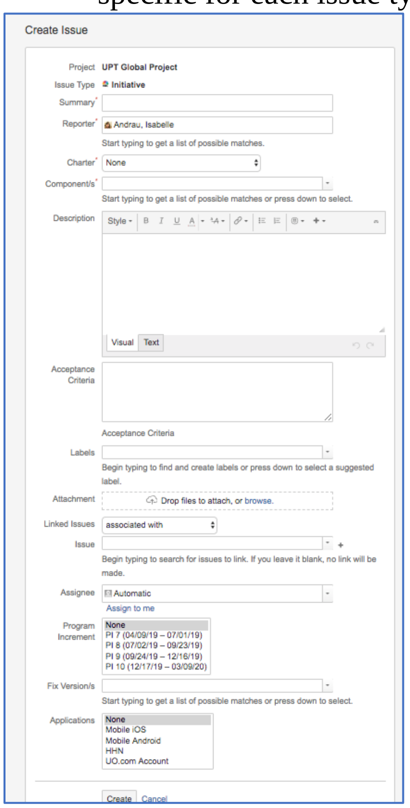
4.1 Initiative

An **Initiative** is a large body of work at the top of the hierarchy. An initiative can last multiple program increments and will include one or more **Epics**.

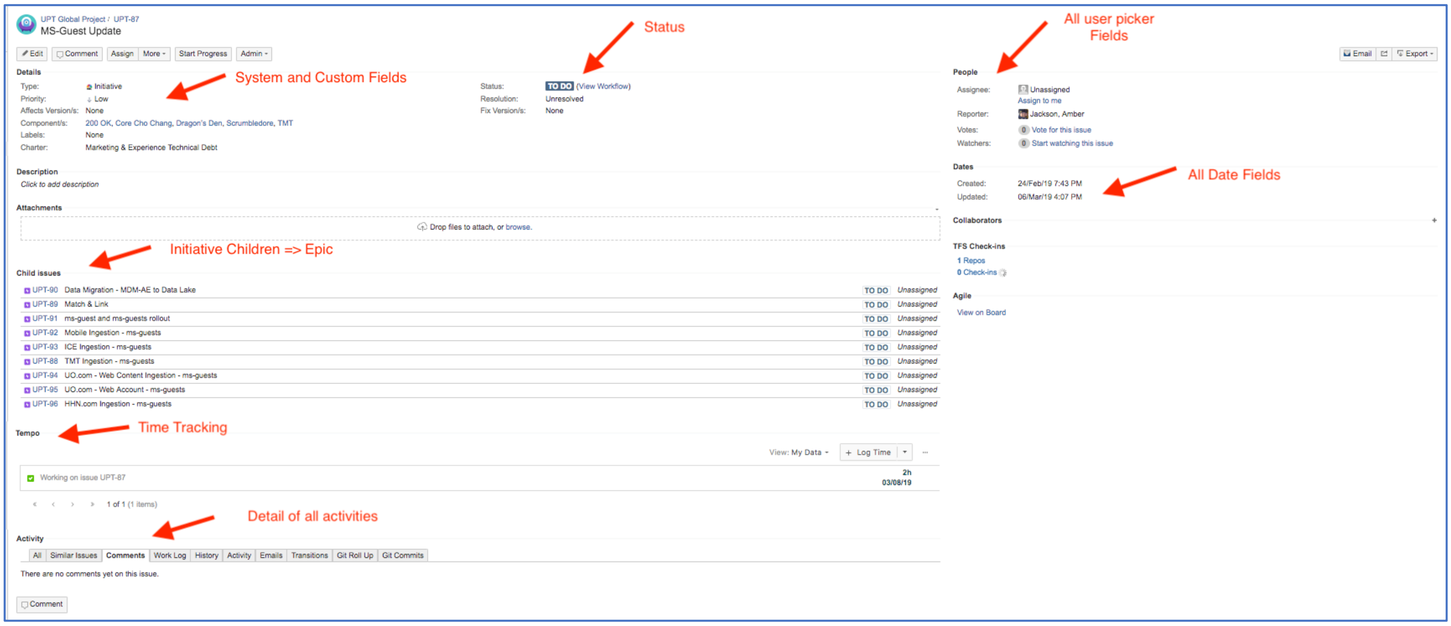
1. Click on the **Create** button in the top menu and a new screen will appear
2. Choose UPT Global Project (UPT) as project
3. Select **Initiative** as Issue Type



4. Click on **Next**
5. Fill out the remaining fields (fields with red star (*) are mandatory). Please note: The create form is specific for each issue type.



6. When the issue is created, click on the Issue key (UPT-xxx) to display issue details.

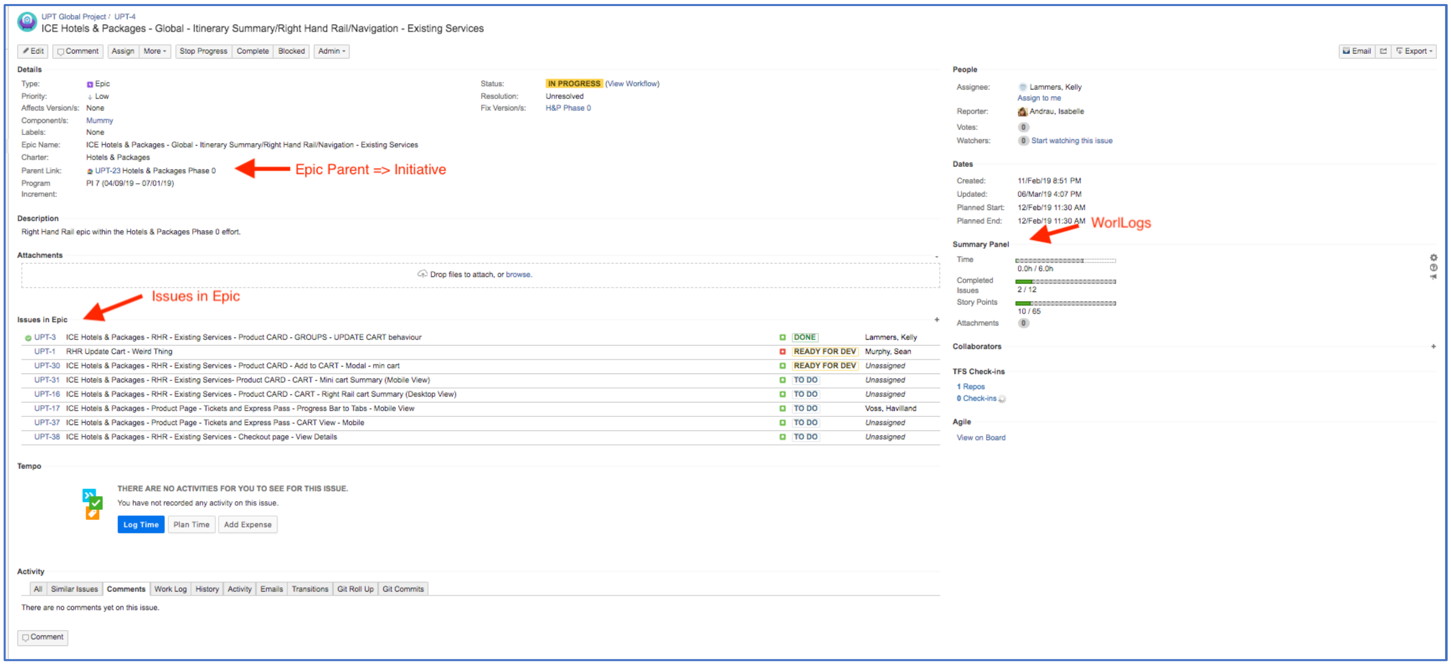


4.2 Epic

An **Epic** captures a large body of work that needs to be broken down into a number of smaller stories. It may take several sprints to complete an epic. The hierarchy for units of work in Jira Software is as follows: *Initiative > Epics > Stories > Subtasks*

1. Click on the **Create** button in the top menu and a new screen will appear
2. Choose UPT Global Project (UPT) as project
3. Select **Epic** as Issue Type
4. Click on **Next**
5. Fill out the remaining fields (fields with red star (*) are mandatory). Please note: The create form is specific for each issue type.

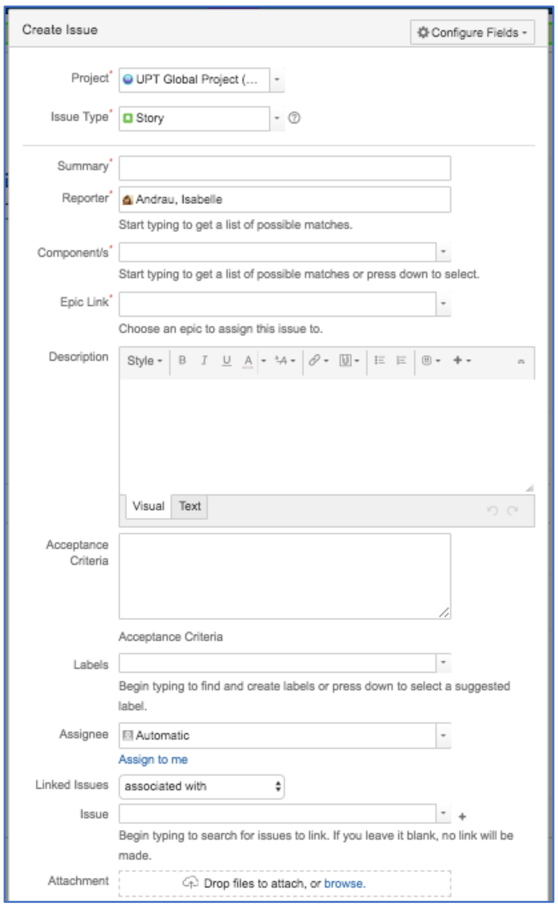
The below is a detail view for Epic. Children for an Epic can be Story, Bug, Risk, Issue, or Test Case. An Epic is the child of an Initiative.

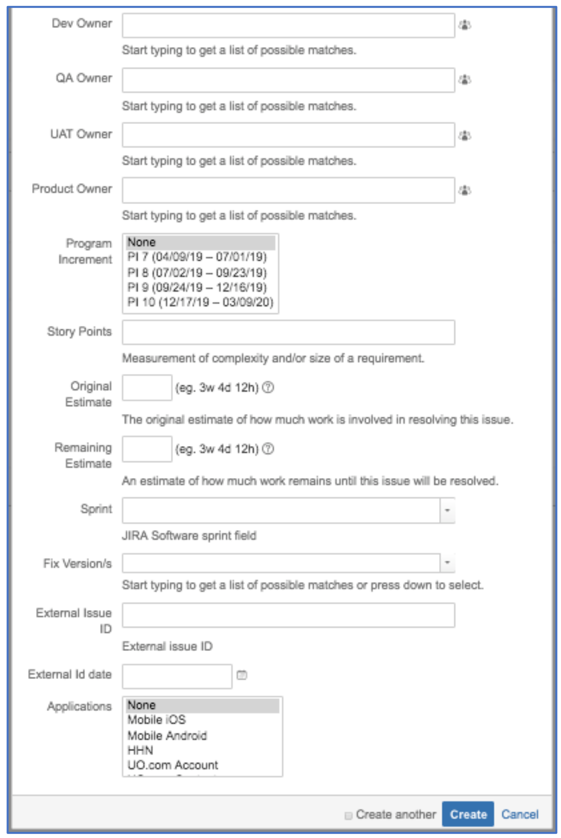


4.3 Story

A **Story** is a listing of acceptance criteria needed to deliver a new epic or piece of work. Generally written from the perspective of a user of the system. A commonly used format is: *As a X, I want to Y, so that Z.*

1. Click on the **Create** button in the top menu and a new screen will appear
2. Choose UPT Global Project (UPT) as project
3. Select **Story** as Issue Type
4. Click on **Next**
5. Fill out the remaining fields (fields with red star (*) are mandatory). Please note: The create form is specific for each issue type.



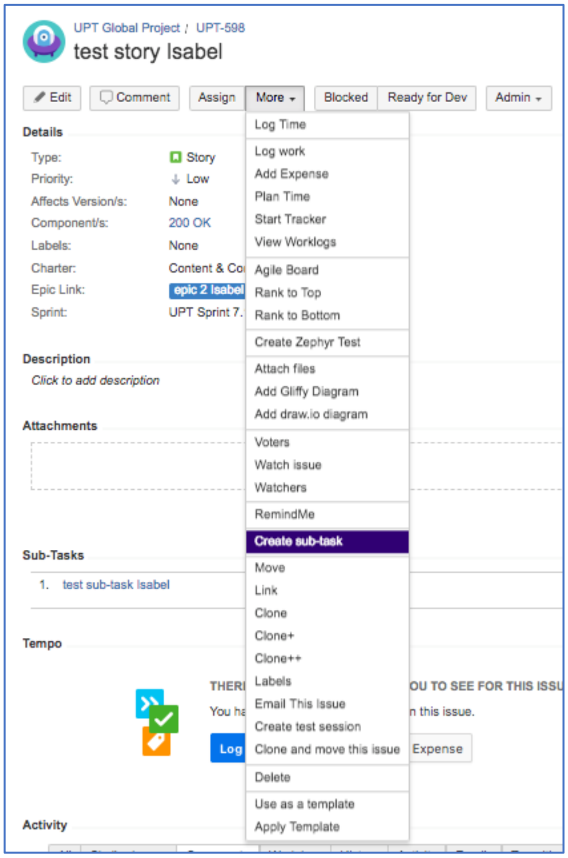


4.4 Sub-task

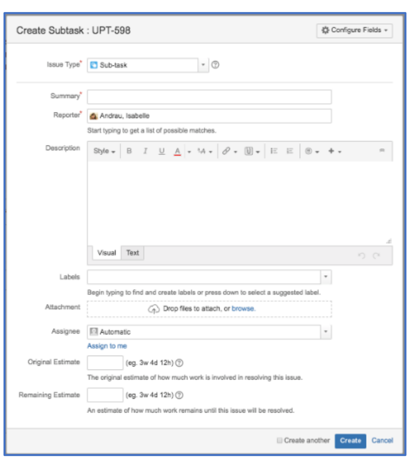
A **Sub-task** can be a "child" of any issue type, depending on the Issue Type Scheme of the project. For UPT, sub-tasks will only be used for stories and bugs.

To create a new sub-task:

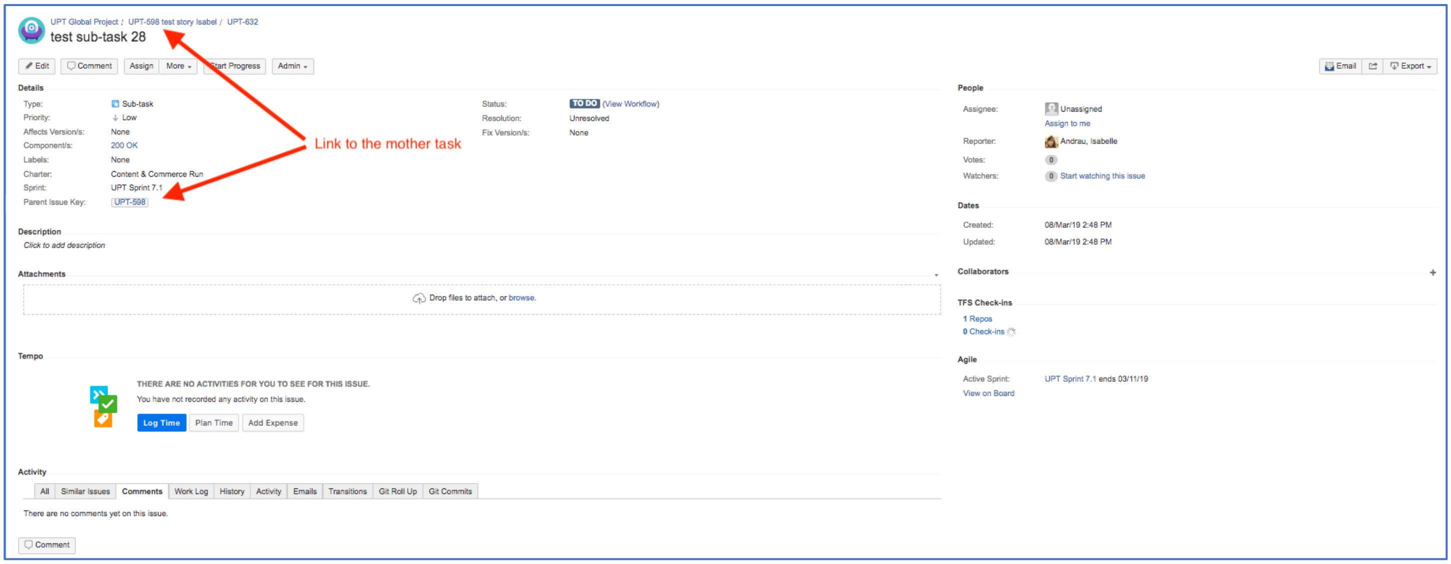
1. From a story or a bug detail, click on the **More** tab
2. Choose Create a sub-task



3. Fill out the remaining fields (fields with red star (*) are mandatory)
4. Click on **Create**



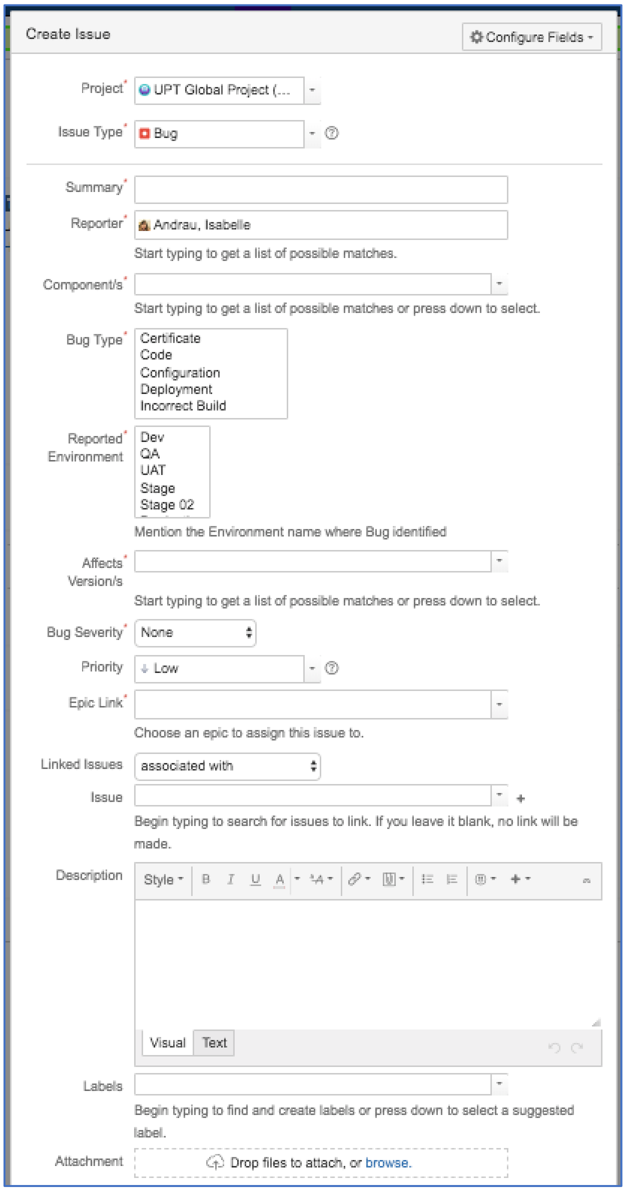
A sub-task looks like any issue. You can easily go back to the parent work item from the summary (or in this project from Parent Issue Key link).

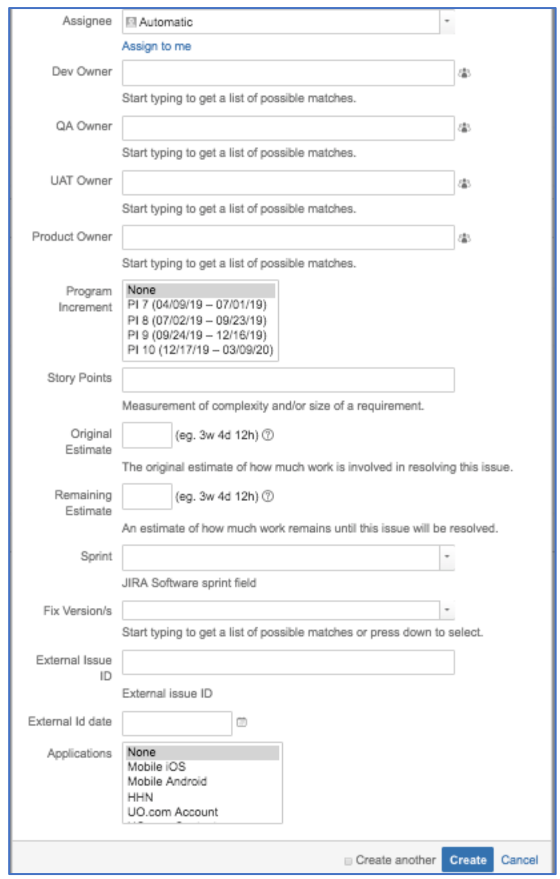


4.5 Bug

A **Bug** is an error, flaw, mistake, failure, or fault in a computer program that prevents it from behaving as intended (such as producing an incorrect result).

1. Click on the **Create** button in the top menu and a new screen will appear
2. Choose UPT Global Project (UPT) as project
3. Select **Bug** as Issue Type



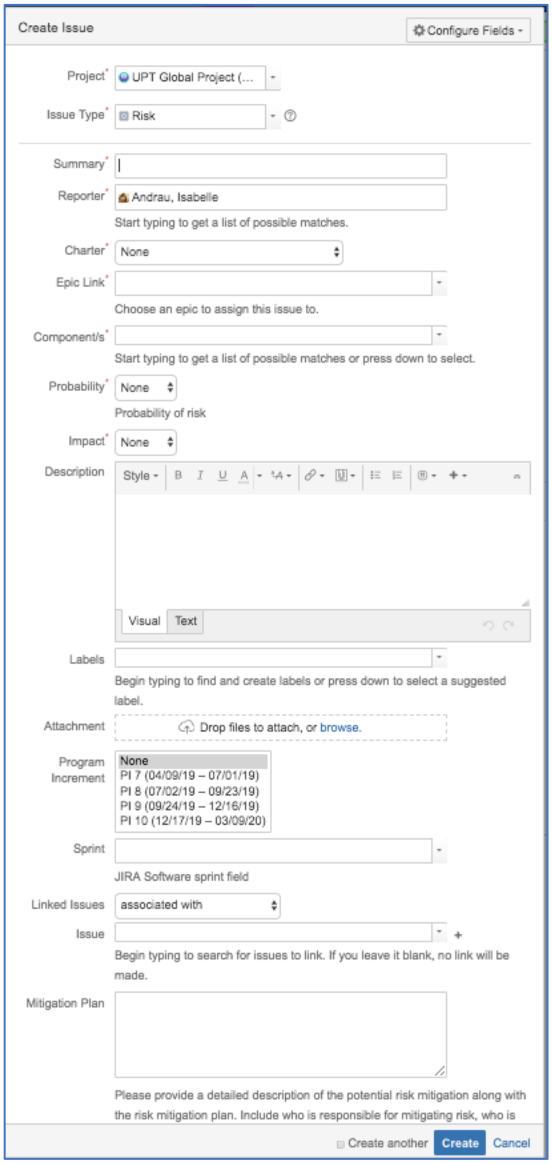


4. Fill out the remaining fields (fields with red star (*) are mandatory)
5. Click on **Create**

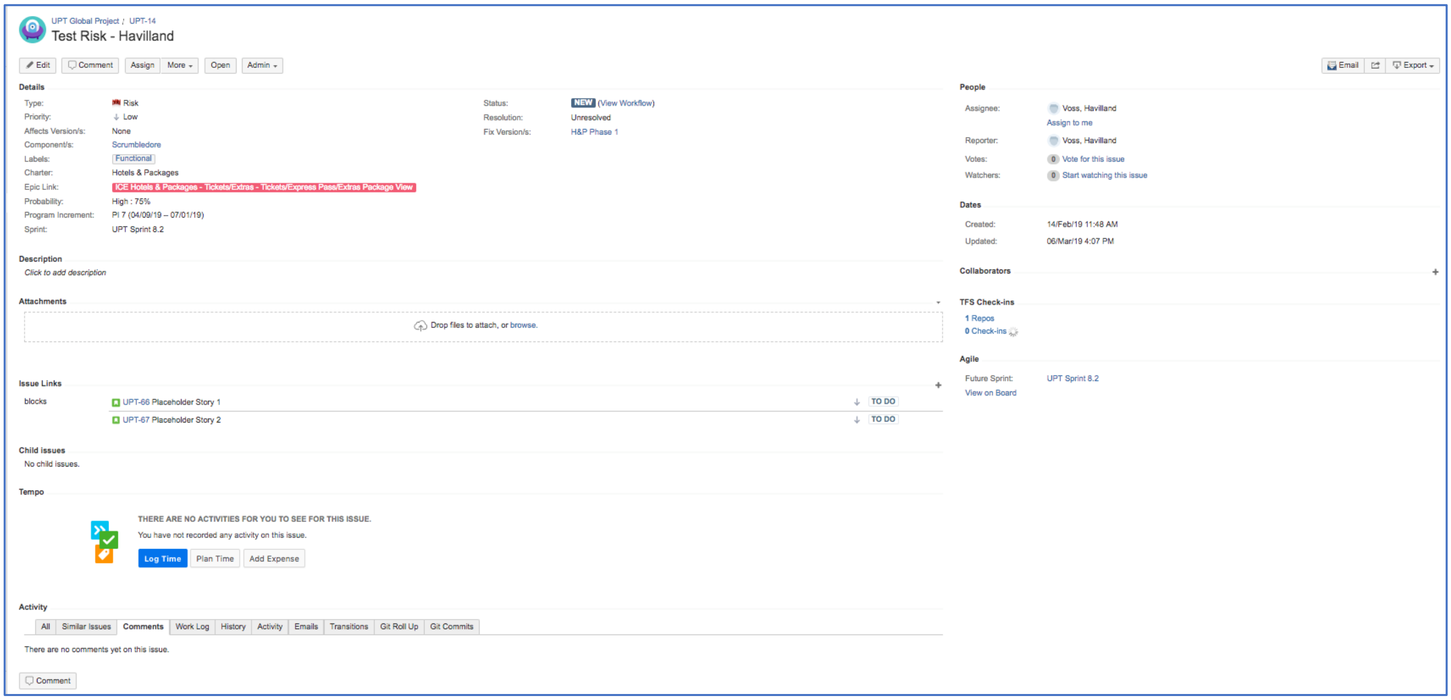
4.6 Risk

A **Risk** is something that could happen that would negatively impact the program or the ability to complete the work on time. Once a risk has been realized, it becomes an **Issue**.

1. Click on the **Create** button in the top menu and a new screen will appear
2. Choose UPT Global Project (UPT) as project
3. Select **Risk** as Issue Type



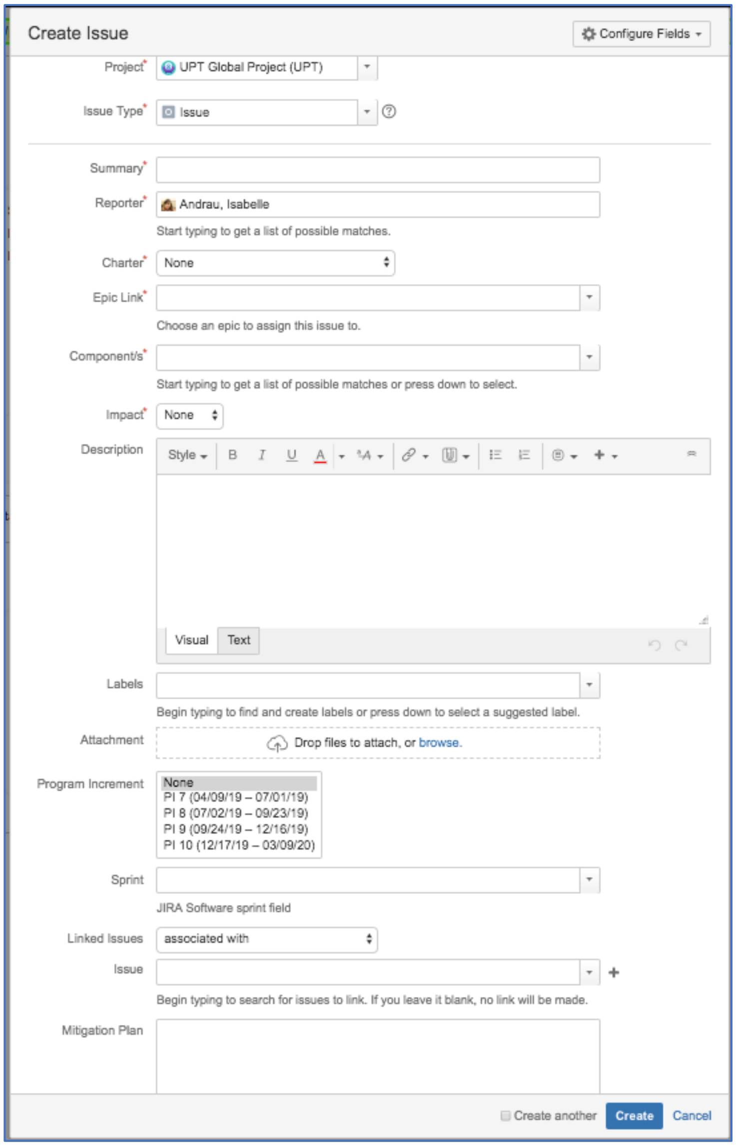
4. Fill out the remaining fields (fields with red star (*) are mandatory)
5. Click on **Create**


UPT Global Project / UPT-14
Test Risk - Havilland
Edit
Comment
Assign
More
Open
Admin
Email
Export
Details
Type: Risk
Priority: Low
Affects Version/s: None
Component/s: Scrumbledore
Labels: Functional
Charter: Hotels & Packages
Epic Link: ICE Hotels & Packages - Tickets/Extras - Tickets/Express Pass/Extras Package View
Probability: High : 75%
Program Increment: PI 7 (04/09/19 – 07/01/19)
Sprint: UPT Sprint 8.2
Status: NEW (View Workflow)
Resolution: Unresolved
Fix Version/s: H&P Phase 1
Description
Click to add description
Attachments
Drop files to attach, or browse.
Issue Links
blocks
UPT-66 Placeholder Story 1
TO DO
UPT-67 Placeholder Story 2
TO DO
Child issues
No child issues.
Tempo
THERE ARE NO ACTIVITIES FOR YOU TO SEE FOR THIS ISSUE.
You have not recorded any activity on this issue.
Log Time
Plan Time
Add Expense
Activity
All
Similar Issues
Comments
Work Log
History
Activity
Emails
Transitions
Git Roll Up
Git Commits
There are no comments yet on this issue.
Comment
People
Assignee: Voss, Havilland
Assign to me
Reporter: Voss, Havilland
Votes: 0 Vote for this issue
Watchers: 0 Start watching this issue
Dates
Created: 14/Feb/19 11:48 AM
Updated: 06/Mar/19 4:07 PM
Collaborators
TFS Check-ins
1 Repos
0 Check-ins
Agile
Future Sprint: UPT Sprint 8.2
View on Board

4.7 Issue

An **Issue** is a realized risk.

1. Click on the **Create** button in the top menu and a new screen will appear
2. Choose UPT Global Project (UPT) as project
3. Select **Issue** as Issue Type



4. Fill out the remaining fields (fields with red star (*) are mandatory)
5. Click on **Create**

5 UPT QA

5.1 Creating a Test Run

A test run should be set up on the first day of your sprint. The Test Run will eventually hold your Sprint's Test Scenarios.

1. Select the **Create** button in the top navigation bar
2. Select **Test Run** as Issue Type
3. Fill out the remaining fields (fields with red star (*) are mandatory)
4. Click on **Create**

Regression Naming Conventions:

- **Test Run** Name: Use the Release name that you are testing against.
  - Mobile Examples: <App : Platform : Release Number>
    - UO iOS 1.19.0 Regression; USH Android 1.19.0 Regression

5.2 Creating a Test Case

A Test Case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing **test cases** can also help find problems in the requirements or design of an application.

1. Open the QMetry plug-in in Jira
2. Select **Manage Test Cases**
3. Expand the **All Test Cases** folder and **Select** your team folder
4. Navigate to the **Master Library** folder and find the functional folder related to **your story**
5. Select the **+New Test Case** button located in the top right corner and a new form will appear
6. Choose **UPT Global Project (UPT)** as project
7. Select **Test Case** as Issue Type
8. Fill out the remaining fields (fields with red star (*) are mandatory)
9. Click on **Create**
10. Once the test cases are complete, select the check boxes on the newly created stories and select the **Bulk Operations** button and **Add to Story**
11. Select the **submit** button to add the test cases to the related story

Test Case Naming Conventions:

Mobile:
[Location (UO USH)] [Platform iOS/Android] [Functionality/Feature] [Test Case Name (what is being verified)]

Mobile Example: UO iOS - Virtual Line - Edit Return Time - Registered Guest

Web:
[Functionality/Feature] [Test Case Name (what is being verified)]

Web Example: ICE - Purchase 2-Park 1-Day Ticket - Adult - Unregistered Guest

5.3 Creating a Test Scenario

QMetry **Test Scenarios** only have to be created one time. You will reuse the **Test Scenarios** (and the associated **Test Cases**) each regression by adding them to your new **Test Run**.

1. Click on the **Create** button in the top menu and a new screen will appear
2. Choose UPT Global Project (UPT) as project
3. Select **Test Scenario** as Issue Type
4. Fill out the remaining fields (fields with red star (*) are mandatory)
5. Click on **Create**

Test Scenario Naming Conventions:

- **Test Scenario** Name: This is where you would specify your scenarios across different browsers, devices and/or platforms.
  - Mobile Examples: <App : Platform : Release Number : Type Of Test : Functionality (if applicable) : Device : OS Version>
    - UO iOS 1.19.0 – Full Regression - iPhone X (12.1.4)
    - UO Android 1.19.1 - Smoke - MS-Guest - Samsung Galaxy S8 (8.0)
  - Web Example: <Release : Type of Test : Functionality (if applicable) : Browser>
    - Web ICE Fast Follow – Full Regression – ICE Flow – Chrome
    - Web ICE Fast Follow – Full Regression - Account Profile – Firefox
    - Web ICE Fast Follow – Smoke – IE

5.4 Adding Test Cases to Test Scenario

The purpose for adding In-Sprint **Test Cases** into **Test Scenarios**, as opposed to all **Test Cases** right into the Sprint **Test Run**, is so that if a **Story** is not completed, that the **Test Cases** don't have to be manually picked out of the **Test Run**. You can reassign the **Test Scenario** to the next Sprint **Test Run**. Also, if you have common functionalities that continually get edited/changed, you can copy and reuse (even change the name) the **Test Scenario**, as opposed to gathering all of the **Test Cases** again.

1. Open a test scenario and select **Reuse Test Case**
2. Search for the test cases that were previously written
3. Select the check boxes for all of the **Story Test Cases** and select the 'Submit' button

5.5 Adding Test Scenarios to a Test Run

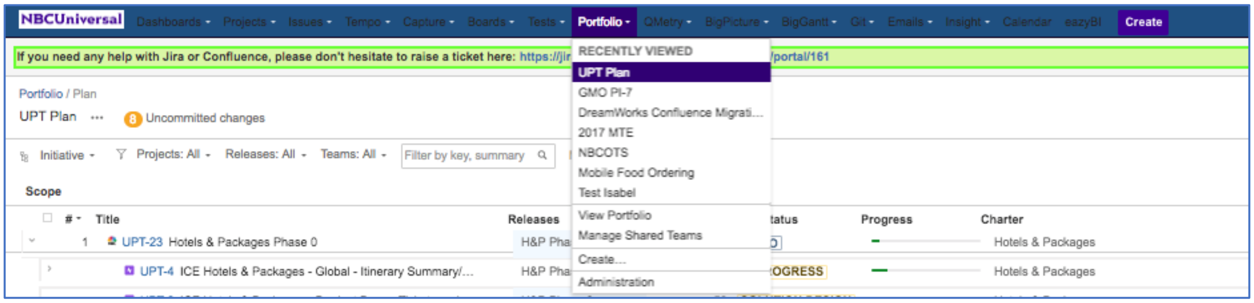
1. Open your sprint **Test Run**
2. In the "Issues in Test Run" section, select the **+Add** button
3. Search for the test scenarios that were previously written
4. Select the check boxes for all of the **Test Scenarios** and select the 'Submit' button

5.6 Executing a Test Run

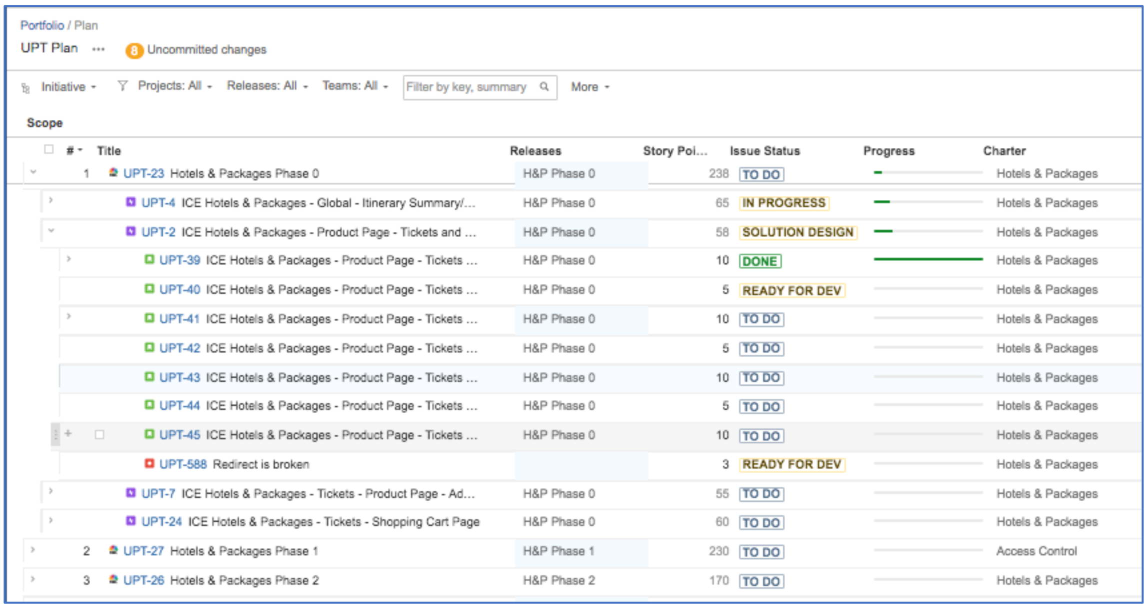
1. Open your sprint **Test Run**
2. In the "Issues in Test Run" section, select the **Execute** button
3. On the test execution page, add your test results

6 Portfolio View

1. Click on the **Portfolio** button in the top menu or under the **More** tab
2. Select **UPT Plan** (or **View Portfolio** and search UPT Plan in the list)



In this view, you can easily see the hierarchy for all your issues.



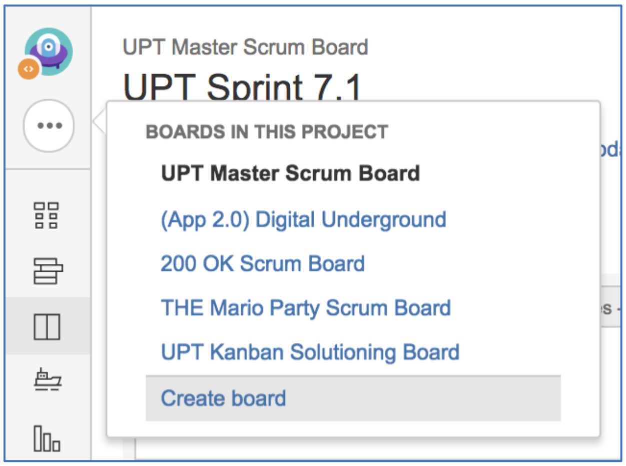
All changes made in this view have to be committed in order to be visible in your agile view.

7 Agile Boards

An **Agile board** displays issues from one or more projects, or from part of a project, giving you a flexible way of viewing, managing, and reporting on work in progress. The UPT Global Project will be set up with Team Boards at the start.

7.1 Create a new board

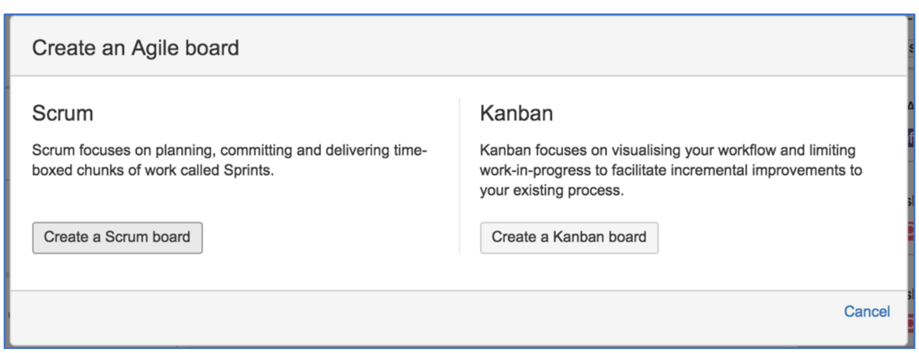
1. Click **"*Create Board*"** in the top right corner under project name



There are 2 board types:

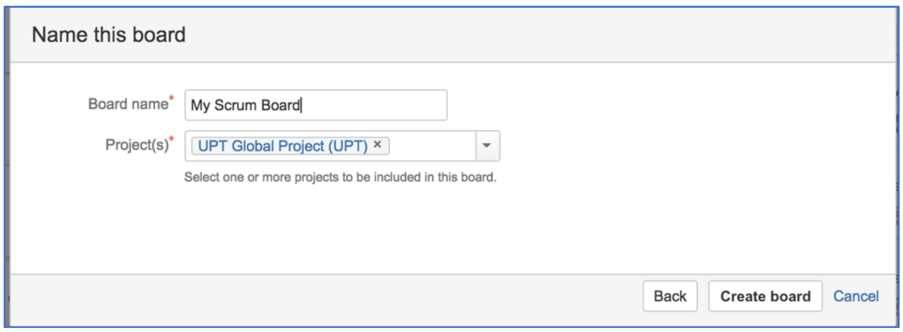
Scrum boards	1. Manage stories, tasks, and, and bugs in **sprints** 2. Suits teams that deliver work on a regular schedule
Kanban boards	• Manage stories, tasks, and bugs in a **continuous flow** • Suits teams who control work volume from a backlog

2. Select "***Scrum***" or "***Kanban***" based on the above definitions



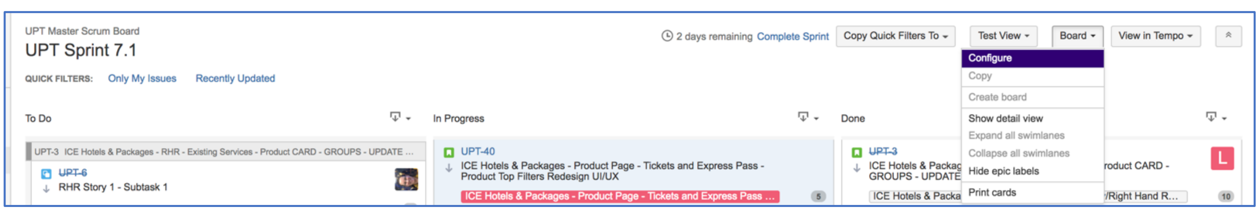
3. Select "board from an existing project" and click Next

4. Input your project information and select **Create Board**



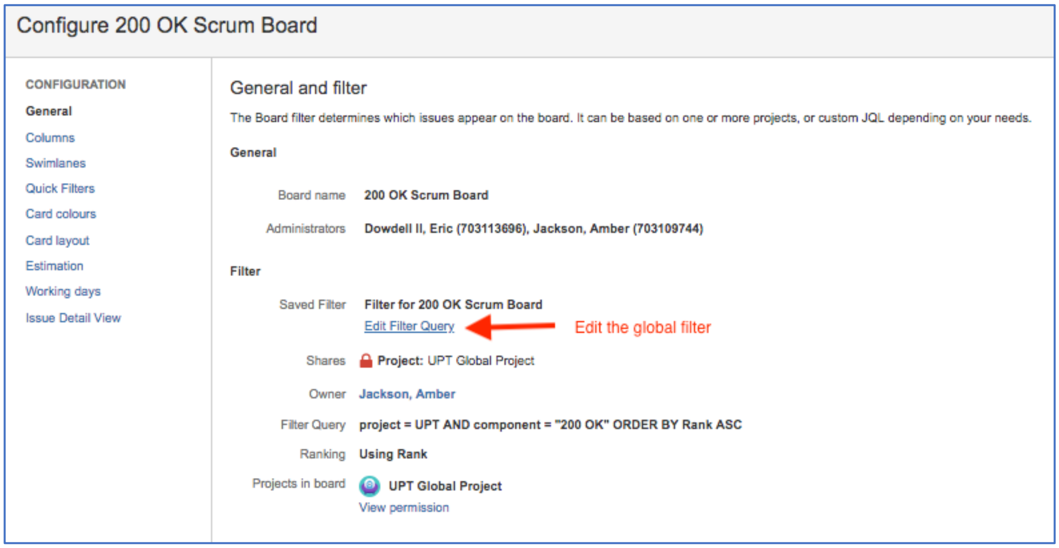
7.2 Configure Scrum/Kanban boards

Click on **Board** and then select **Configure**

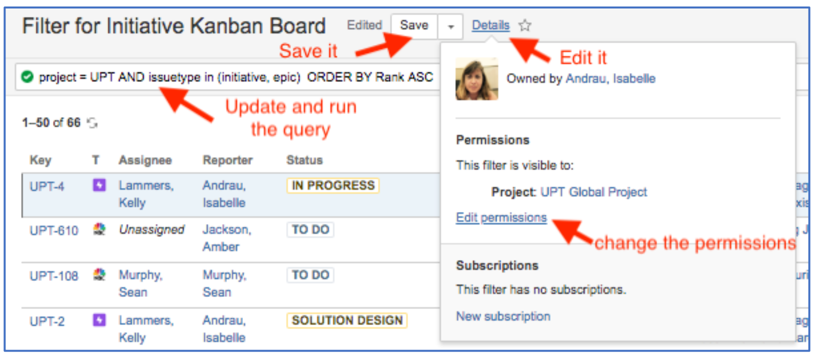


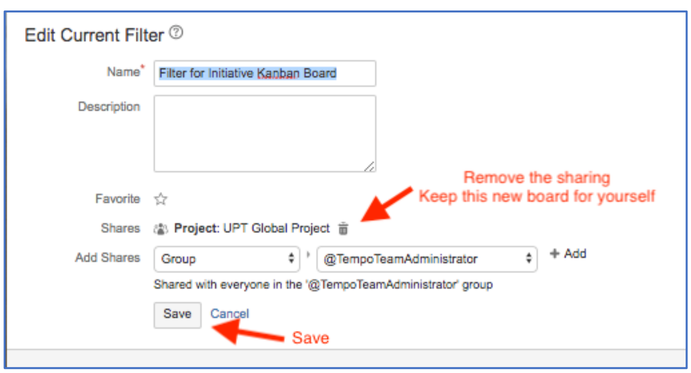
7.2.1 Global Filter and sharing

You can change the filter that is applied on this board. For example, you can decide to see only issues in a specific component.



1. Click on the filter and change it
2. Change the search parameters and save the filter
3. Edit your filter and change the permissions

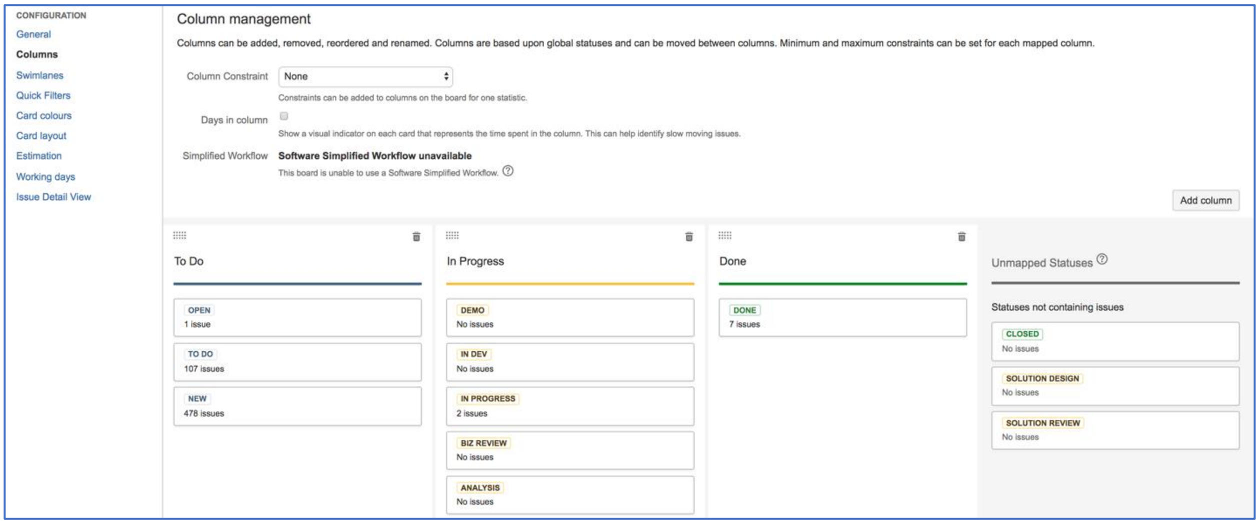




Do not share this board if it is for your own usage. New boards could confuse other Team Members.

7.2.2 Board Columns

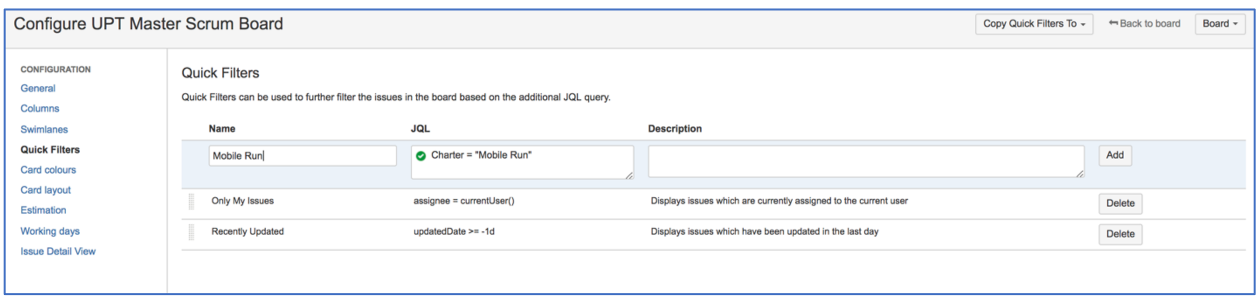
Columns can be customized based on your preferences. Drag and Drop each status field into the corresponding column to update. Generally, columns represent the workflow of the work items.



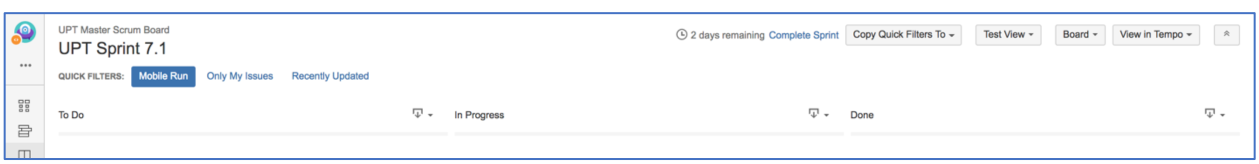
7.2.3 Quick Filters

You can create quick filters to get a restricted view of the work items that appear on your board.

1. **Name** your quick filter
2. Create your JQL query
3. Select **Add**



From the board, click on the quick filter to have it applied on your board. Click again (unclick) to get back the global view.



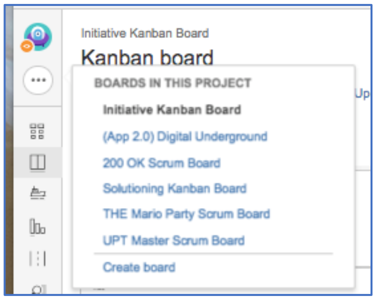
You can add multiple quick filters together.
For example: Charter=Mobile Run AND Assignee= CurrentUser() => Click on both filter: *Mobile Run* and *Only My Issues*

8 UPT Agile Boards

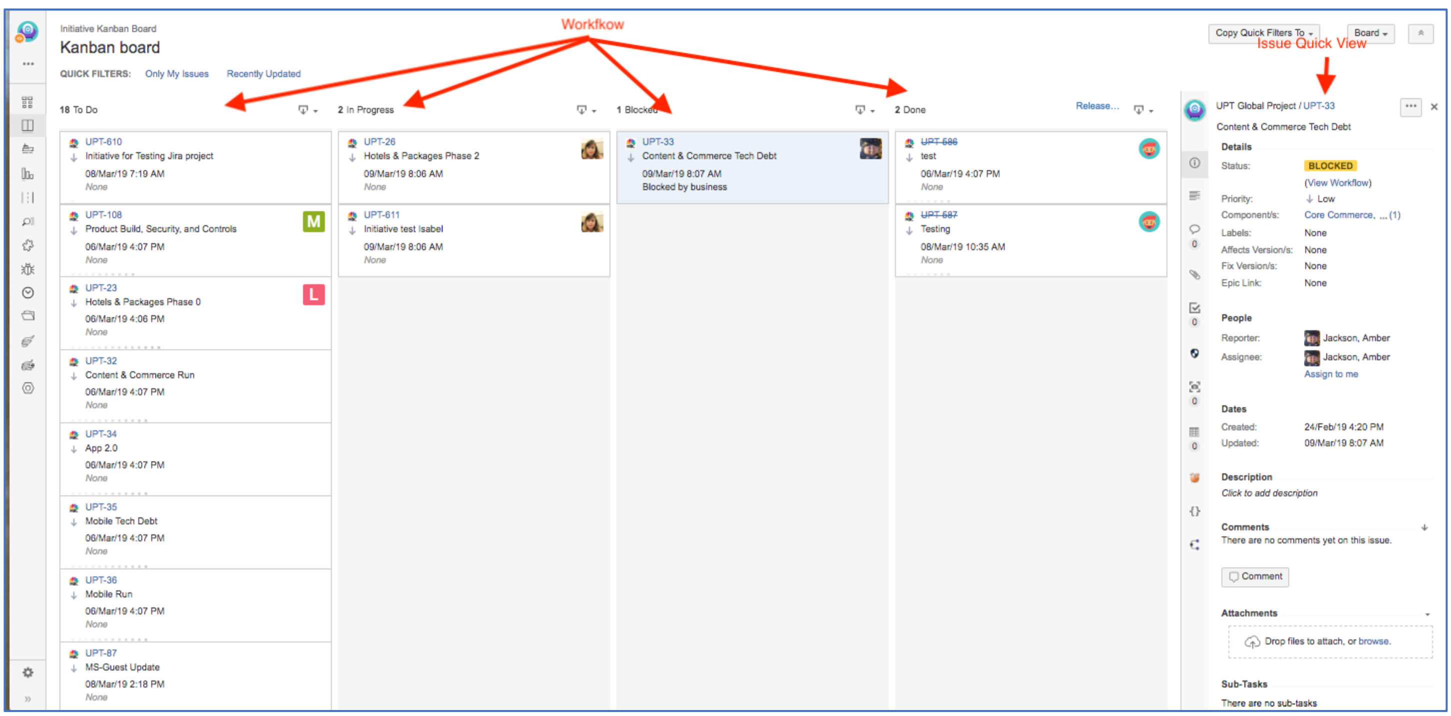
Several boards have already been created for your team. Boards provide a shared view of all work that hasn't started, work that is in progress, and work that is completed.

8.1 UPT Initiative Board

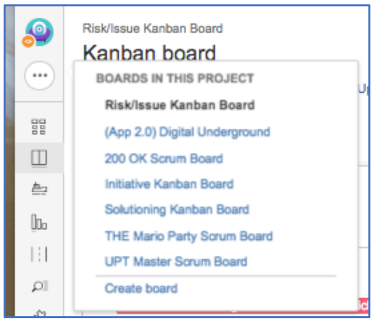
The initiative board provides a global view of our work at the top of our work item hierarchy.



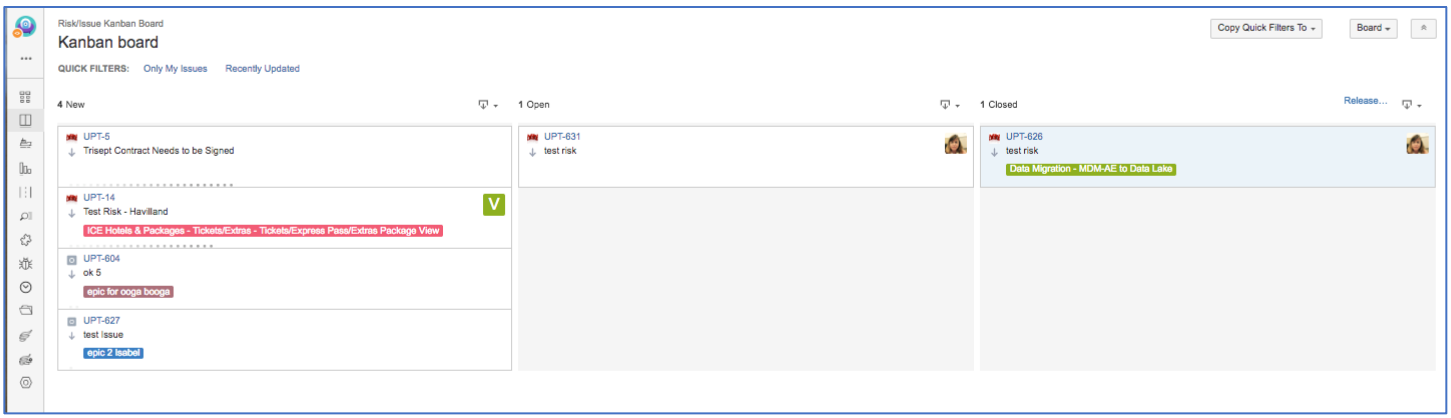
You can get a global view on all initiatives and the status for each of them.
When you click on any issue key, you get a quick view on the right panel where you can update the issue.



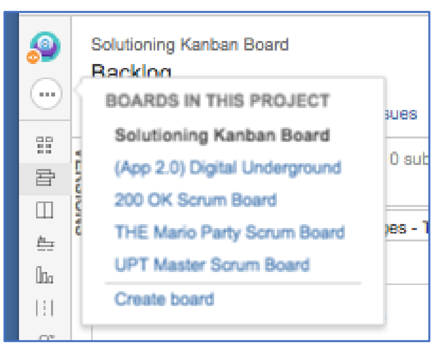
8.2 UPT Risk/Issue Board



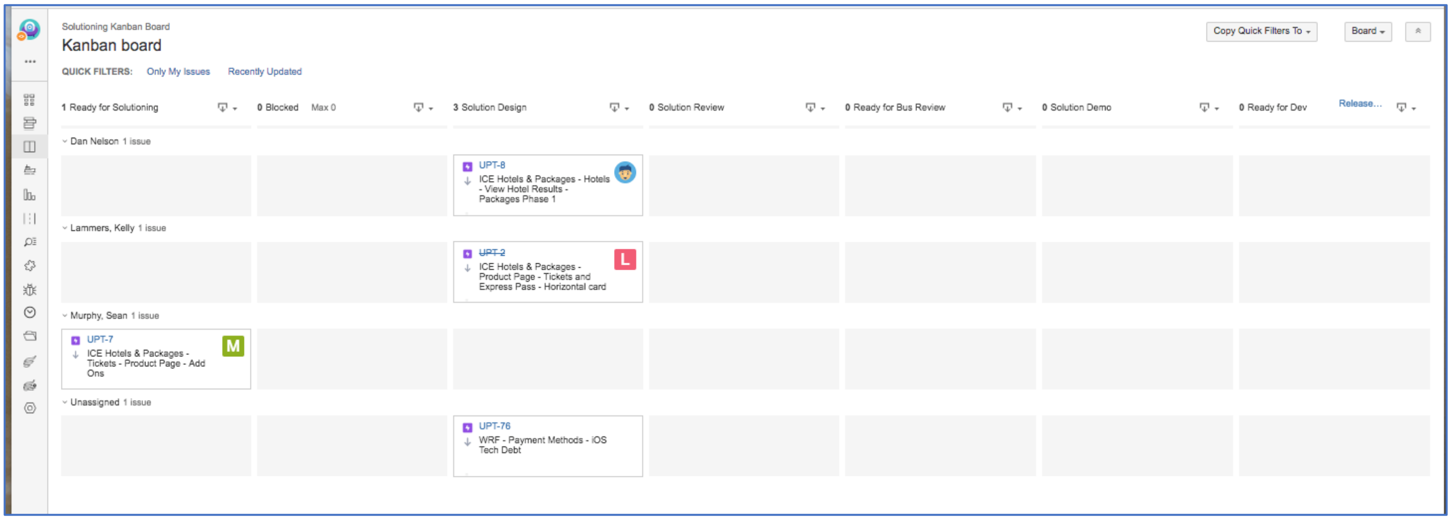
You can get a global view on all risk/issues and the status for each of them.
When you click on any issue key, you get a quick view on the right panel where you can update the issue.



8.3 Solutioning Kanban Board

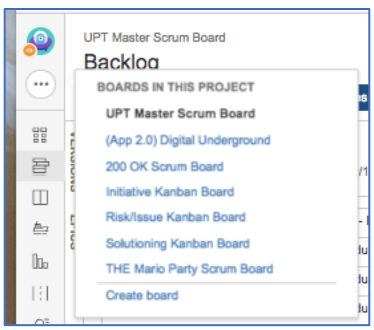


This board is specific to managing first steps for Epics before going in development, representing Solutioning Process workflow.



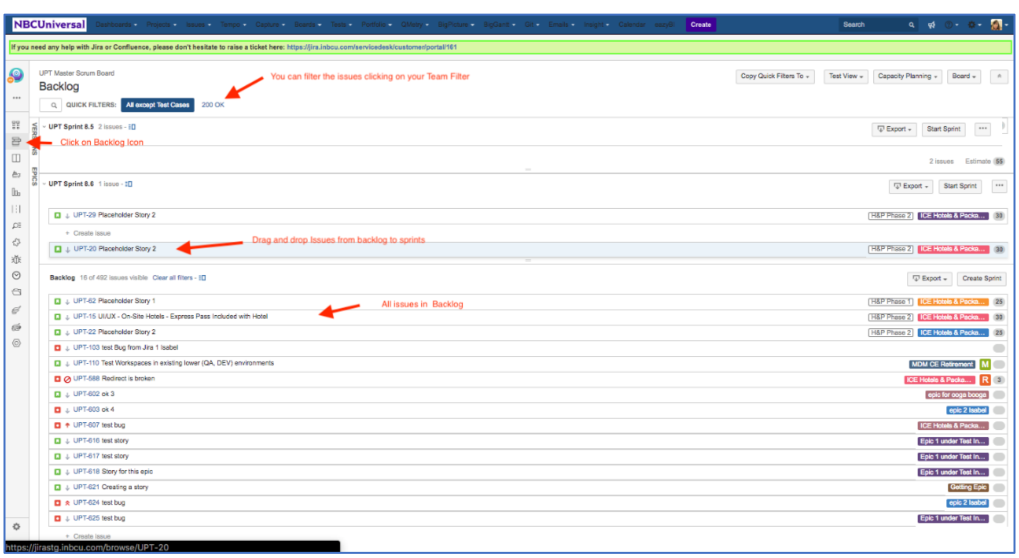
8.4 UPT Master Scrum Board

This master scrum board has been created to manage sprints across all UPT scrum teams.



This will help you and your teams manage all issues and organize them from backlog to sprint, and all UPT scrum teams will follow the same process.

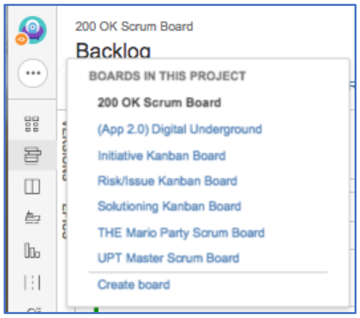
Sprints are created, started, and closed by project administrators only.



8.5 "Team" Scrum Board

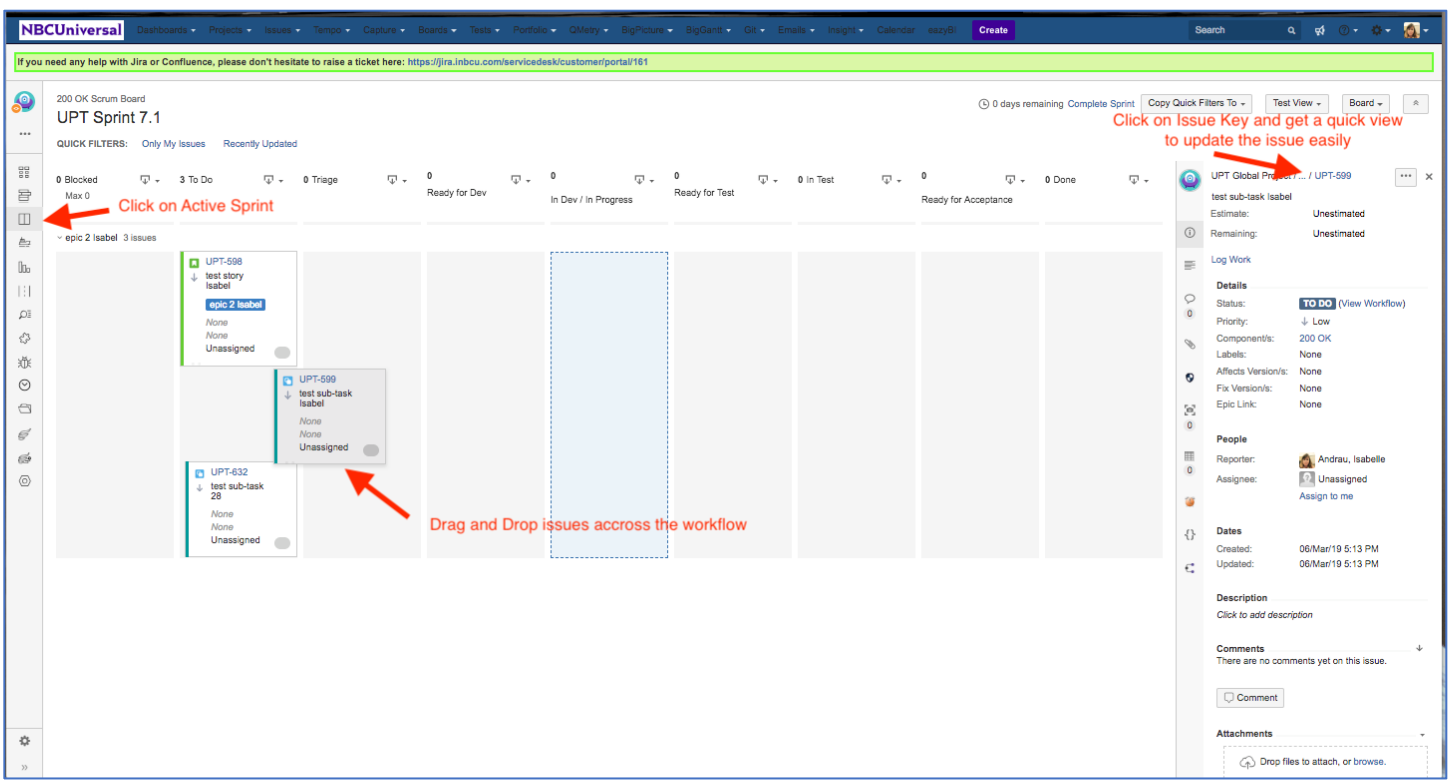
Team collaboration and incremental delivery starts on the **Jira Scrum Board**. The **Scrum** framework enables software **teams** to manage complex projects by creating a culture of collaboration. The **Jira Scrum Board** is the tool that unites **teams** around a single goal and promotes iterative, incremental delivery.

Each team will have its own scrum board. Only issues assigned to this team will be present in the board.



8.5.1 Active Sprints

The Active sprints of a Scrum board displays the issues that your team is currently working on. You can create and update issues, and drag and drop issues to transition them through a workflow.The Active Sprint view will be the most important view. It gives you the situation for all issues open for this sprint. You can easily update the issues from this view.



What can I do in the Active sprints of a Scrum board?

Task	Instructions
View an issue's details	Click the desired issue on the board. The issue's details will display in a panel on the right of the board. You can also select multiple issues, right-click, then select **View in Issue Navigator** for a different view of the issues.
Transition an issue	Drag and drop an issue between columns.
Create Sub-tasks	Select the desired issue, press the '.' key, and type 'create sub-task' to open the 'Create Sub-task' dialog. Create the sub-task as desired. Sub-tasks are useful for breaking a story (issue) down into implementable chunks.
Filter issues	Click the desired Quick Filter (below the board name), e.g. **Only My Issues**.
Remove issues from a sprint	Right-click (or select multiple issues and right-click) to open a menu and select **Remove from Sprint**.

8.5.2 Using your backlog

The **backlog** of a Scrum board shows the issues for your project grouped into a backlog and current and future sprints. In the Scrum backlog, you can create and update issues, drag and drop issues to rank them, or assign them to sprints, and more.

What can I do in the Backlog?

Task	Instructions
Add issues to the backlog	Click **Create** in the header to open the 'Create Issue' dialog and create your issue. The issue will be added to the backlog under the currently selected issue, or at the top of the backlog if no issue is selected. Tips: • Tick the **Create another** checkbox in the 'Create Issue' dialog to keep it open, if you are creating multiple issues. • You can quickly create issues using the inline issue create feature. Just click **+ Create issue**.

Prioritize the backlog	Drag and drop an issue to rank it. You can also right-click the issue to open a menu that allows you to send it to the top or the bottom of the backlog.
View and edit an issue's details	Click the desired issue on the board. The issue's details will display in the issue detail view, where you can also edit some issue details, depending on the issue detail view configuration. If you want to select an issue, rather than open the issue's details, use ctrl-left click or command-left click.
Estimate stories	Use the 'J' and 'K' keys to move through issues in the backlog and show the details on the right-hand side of the screen. Use the 'E' key to edit an issue, then update estimates or story points as you go. The Story Points field should be used for stories and bugs.
Identify the workload for specialists	The avatars for users (specialists) who have work assigned to them in a sprint are shown at the top of a sprint. Click ... (next to avatars) to view the sprint workload for assignees. If your JIRA uses time tracking, you will also see the time required to complete this workload, shown in the Remaining Time Estimate column.
Create sub-tasks	Select the desired issue, press the '.' key, and type 'create sub-task' to open the 'Create Sub-task' dialog. Create the sub-task as desired. Sub-tasks are useful for breaking a story (issue) down into implementable chunks.
Move an issue into a sprint	Drag and drop an issue from the **Backlog** section to the relevant **sprint** section, to move the issue from the backlog to the relevant sprint.
Delete an issue	Right-click an issue and select **Delete**.
Find issues	Create your own Quick Filters to view only the issues you want. Search for an issue by name, code, or assignee.
Plan sprints	Project Admins will create sprints for the PI. If you do not see the sprint in your backlog view, you will need to add the sprint by editing the story/bug. Once one item is in the sprint, it will appear in the backlog view. You can also drag and drop the horizontal divider to add or

	remove multiple issues. The sprint footer will display the number of issues and total estimated work. While a sprint is active in the Active sprints, you can still plan subsequent sprints in the Backlog, but you won't be able to start them until the active sprint is completed. (You can, however, drag and drop an issue in the Backlog onto the active sprint.)
Add goals to sprints	Click > **Add sprint goal**, then enter a goal for this sprint to keep your team focused. The sprint goal will be displayed below the sprint name in the backlog, scrum board, and sprint report.

9 Time Tracking

Work Logs can be recorded in any issue in Jira.

9.1 Type of Work to be Logged in Jira

9.1.1 Logging Time for Program Administration and Time Off

For work that was previously logged to "Labor – Program" in Rally, we will have an initiative created for each item to cover 2019, with an epic created for each program increment, and a story created for each sprint. Each user logs time to the corresponding story as needed. An example:

In Rally, time was charged to:

- Labor – Program
    - DEP Program New
        - PTO / Time Off / Sick

In Jira, there will be:

- An Initiative for "PTO / Time Off / Sick - 2019"
    - An Epic for "PTO / Time Off / Sick – PI7"
        - A story for "PTO / Time Off / Sick – S7.1"
        - A story for "PTO / Time Off / Sick – S7.2"
        - Etc.

These items will have an Initiative created with the same name, which can be found when viewing the UPT Initiative Board (see above section 8.1 UPT Initiative Board). Once viewing the Initiative that corresponds to your work, you can drill down to the appropriate Epic and then to the appropriate Story.

Alternatively, you can use the "Search for Issues" capability to find the appropriate Story (see below section 11.1 Search Issues).

9.1.2 Logging Time for Feature Work

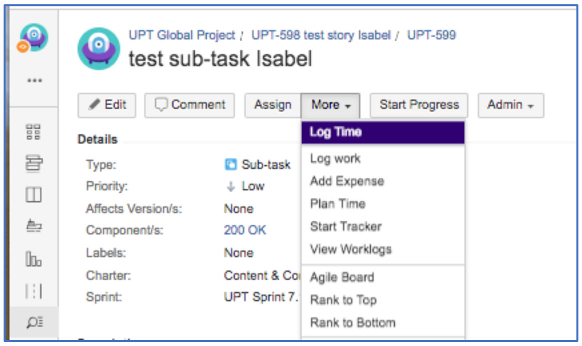
For time spent on development and testing work, log time directly to the Story, Bug, or Sub-task that covers that work.

Time spent in scrum ceremonies should be tracked on the development Initiative/Epic/Story that the team/individual is working on at the time.

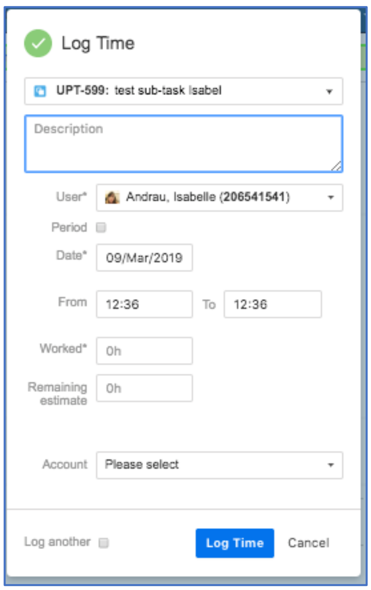
- Example: during a backlog refinement session, everyone participating would log time on the Epic being reviewed/groomed.
- Example: during daily scrum stand up, each person should log that time against which ever story they're in the middle of working.

9.2 Create a Work Log

1. Open the detailed view of the issue you want to update
2. Click on **More** and select **Log Time**

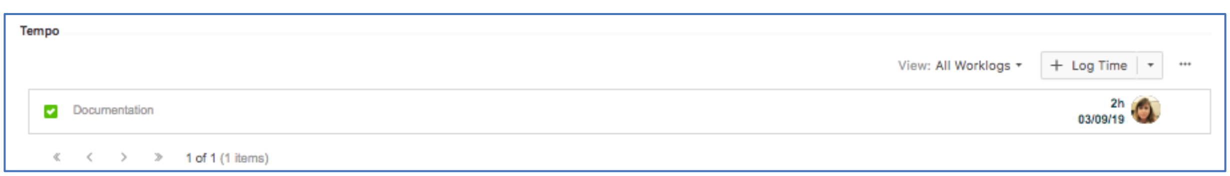


3. Add your worklog and validate



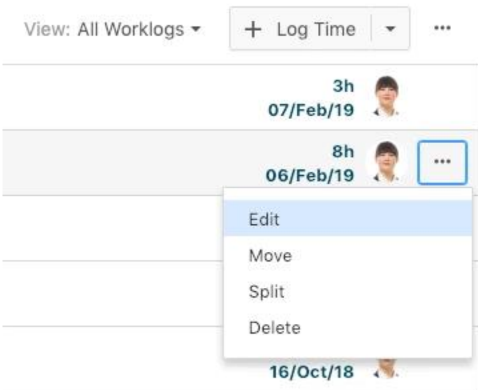
You can easily manage all the previous records in the ***tempo part*** of issue detail

4. Click on the ellipsis at the end of the line. Choose to edit/move/split or delete the worklog



9.3 Edit a Work Log

1. In the Jira issue view, click the ellipses (**...**) to the right of the relevant worklog, plan or expense in the **Tempo** panel
2. Select **Edit**



3. Make the necessary changes in the dialog
4. Click **Update**

9.4 Delete a Work Log

1. In the Jira issue view, click the ellipses (**...**) to the right of the relevant worklog in the **Tempo** panel
2. Select **Delete**
3. In the **Delete Worklog** dialog box, click **Delete** to confirm your action

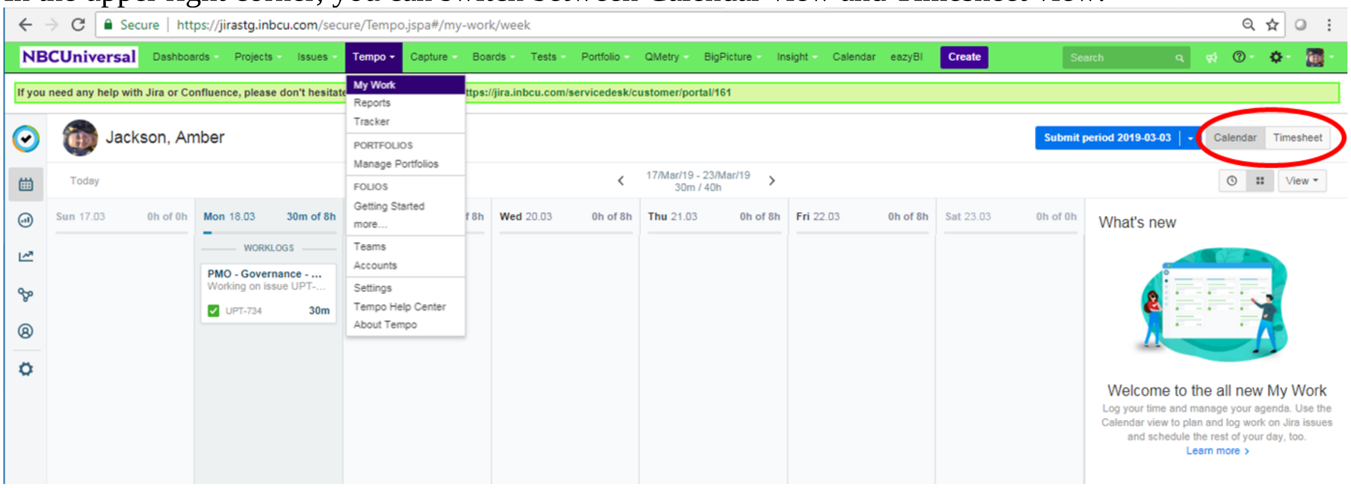
9.5 Move a Work Log between Issues

1. Select My work in the sidebar
2. Select Timesheet at the top-right
3. Click the worklog entry you want to move
4. Click the ellipses (...) at right, then select Move
5. Select the issue the worklog should be moved to
6. Optional-Update the remaining estimate
7. Click Move

9.6 View Your Time Sheet

To view your time sheet for the week:

2. Select Tempo in the top banner
3. Select My Work
4. You will see your time sheet for the week (Sunday-Saturday).
5. In the upper right corner, you can switch between Calendar view and Timesheet view.



9.7 Logging other type of time (overhead)

There will be times you need to log your time a little more granularly. To help with this, we have created additional epics to use when warranted.

- PTO / Office Holiday (This is for vacation, sick time, personal business, and company holidays)
- Training
- Town Halls
- Non-Project Meetings (1-on-1s with your manager, Xtras, agile ceremonies (e.g. sprint planning, sprint retrospectives, PI planning events), etc.)
- Administration (Manager use only)
- Intake/Charter Creation
- Production Support

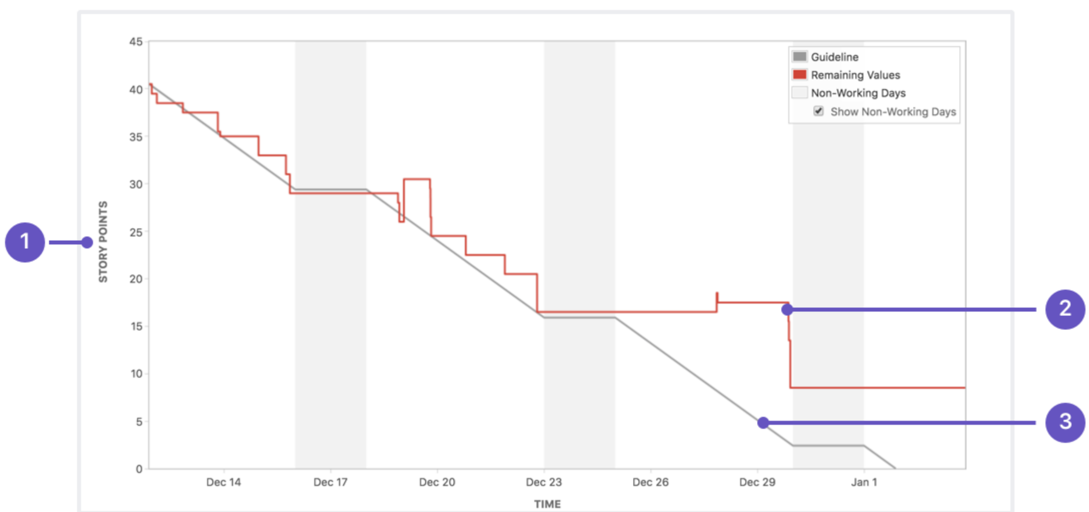
10 Reports

Jira has several agile reports that are created automatically.

1. To view, click on **Reports**
2. Select the report you want to view

Burndown Chart is often the most popular report. **Always estimate your issue before starting a sprint.** If you start your sprint without having estimated your issues, and you add issues during the sprint, your burndown chart won't have the look you expect.

Understanding your burndown chart:



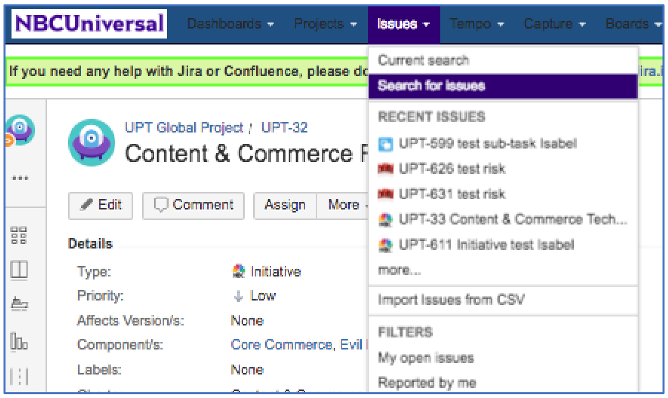
1. **Estimation statistic**: The vertical axis represents the estimation statistic that you've selected.
2. **Remaining values**: The red line represents the total amount of work left in the sprint, according to your team's estimates.
3. **Guideline**: The grey line shows an approximation of where your team should be, assuming linear progress. If the red line is below this line, congratulations - your team's on track to completing all their work by the end of the sprint. This isn't foolproof though; it's just another piece of information to use while monitoring team progress.

11 Issue Search/Filter

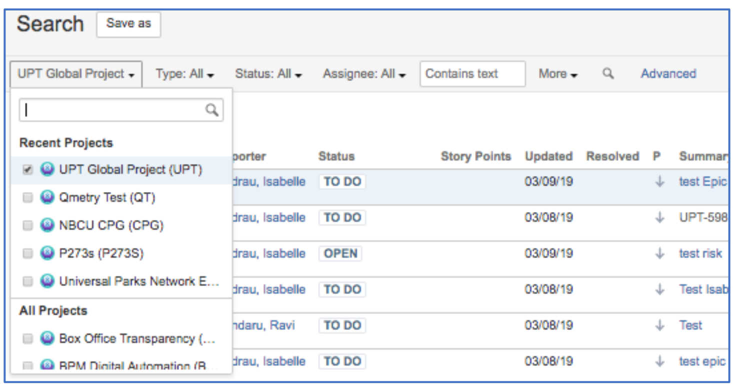
A feature that is really important in Jira is to be able to search issues easily and save your searches.

11.1 Search Issues

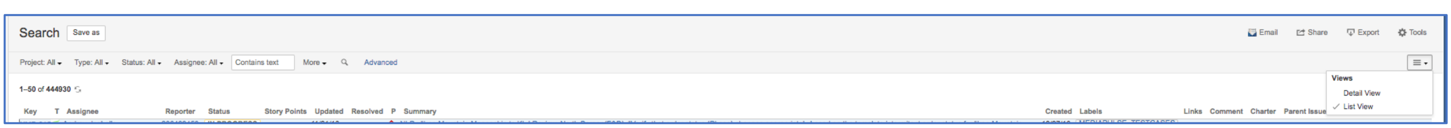
Go to Issues and select Search for Issues



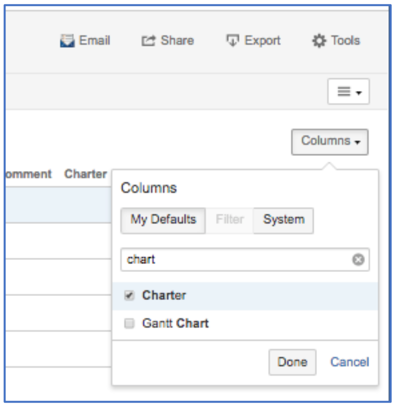
Start with a basic search: select projects, issue types, statuses, etc. from drop down lists and click on the magnifying glass button. To search on additional criteria, click More, and select the variable you want to search.



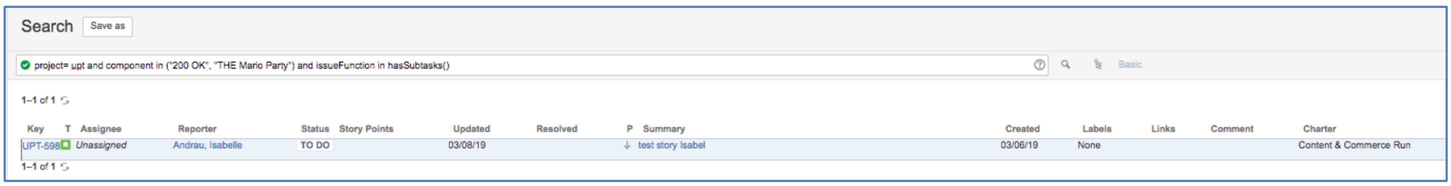
By default, the view for the results is detail view, you can change it to list view. To update, go to the list icon on the right, and change to list view.



You can customize the columns you want to display
Go to columns button and **Select/Deselect** the fields you want



If you need to create a more complex search, you can use JQL language to create a query
Click on the **advanced** link to switch to **advanced mode**

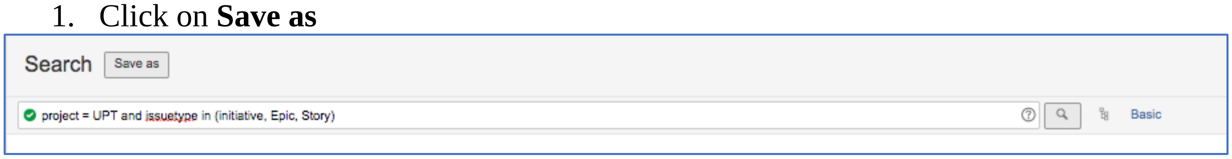


You can export this data to many file types. When selecting the file type, make sure "current fields" is specified.

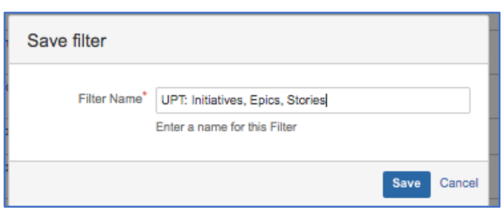
11.2 Create and manage filters

You can save your search

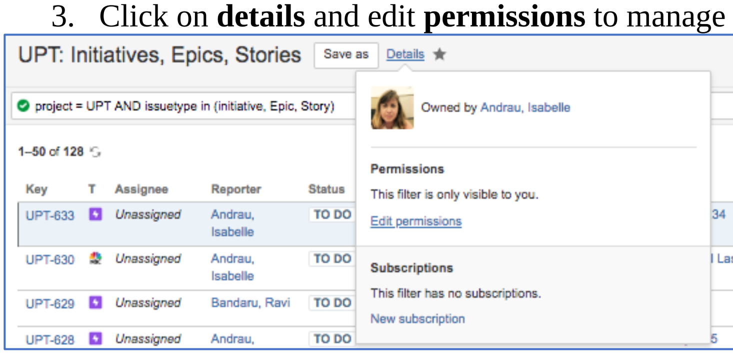
1. Click on **Save as**



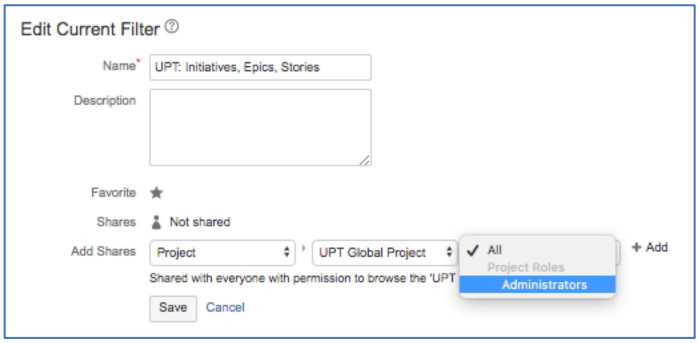
2. Name your filter



3. Click on **details** and edit **permissions** to manage permissions



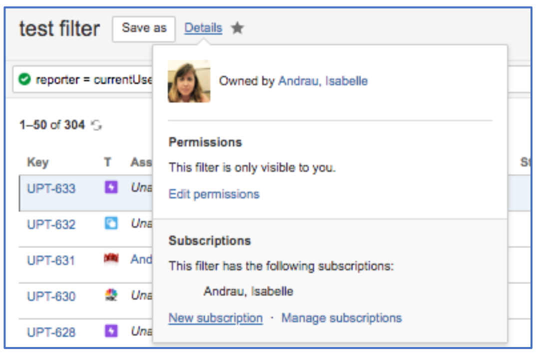
4. Choose the group of people or project users you want to share this filter with



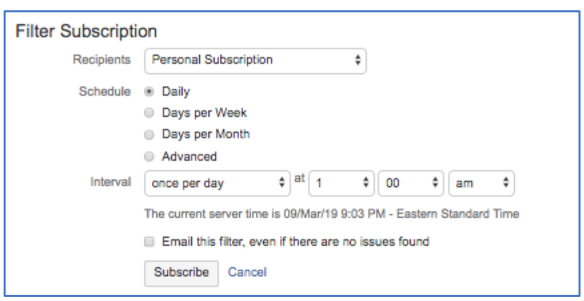
Don't share your filters with people that the information doesn't apply to

You can also subscribe to your filters.

5. Click on detail and Select **New subscription**



6. Fill out the form and hit **Subscribe** to submit

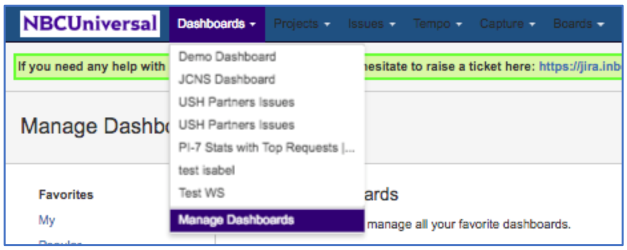


12 Dashboard

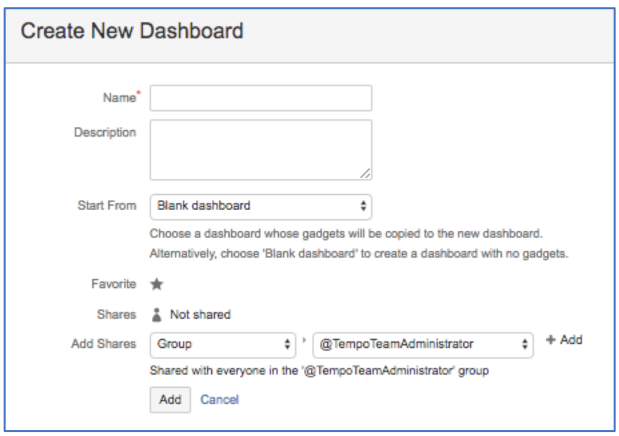
12.1.1 Creating Dashboards

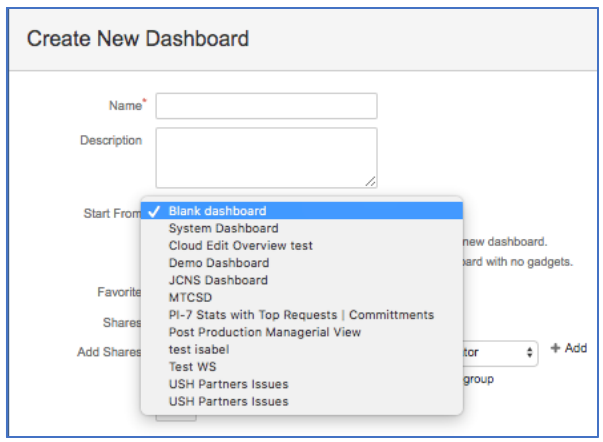
You can create and customize your own dashboard to display the information you need. Only Jira admins can customize the system dashboard.

1. Go to Dashboards and Select Manage Dashboards

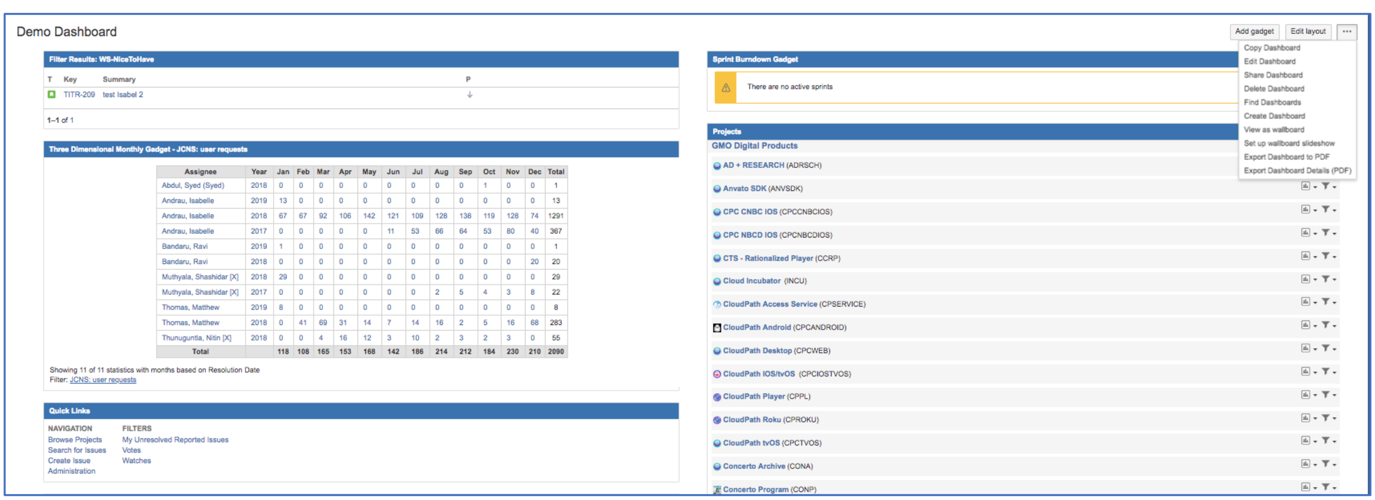


2. Select **Create new Dashboard** (you can start from a blank dashboard or start from another dashboard)





3. Name your dashboard and add a description so your team knows when to use it
4. Fill out the remaining fields and click **Create**
5. Click on the ellipsis and get access to all options and manage your dashboards



6. To change your dashboard's sharing permissions, click …> **Edit dashboard**

12.1.2 Choose a dashboard layout

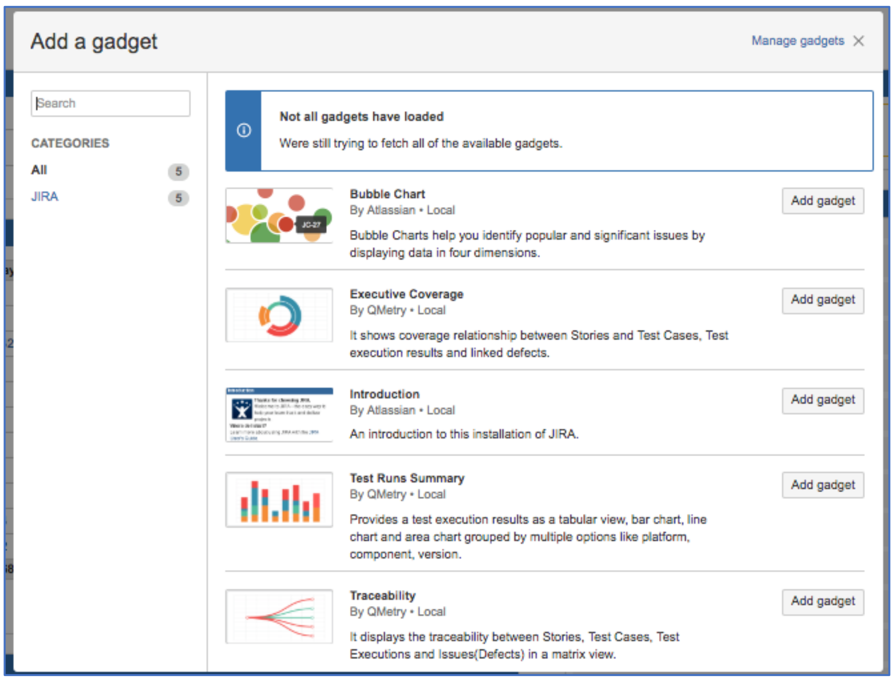
7. To choose a different layout for your dashboard page (three columns instead of two, for example):
8. Click **Edit Layout**
9. Select your preferred layout

12.1.3 Managing gadgets

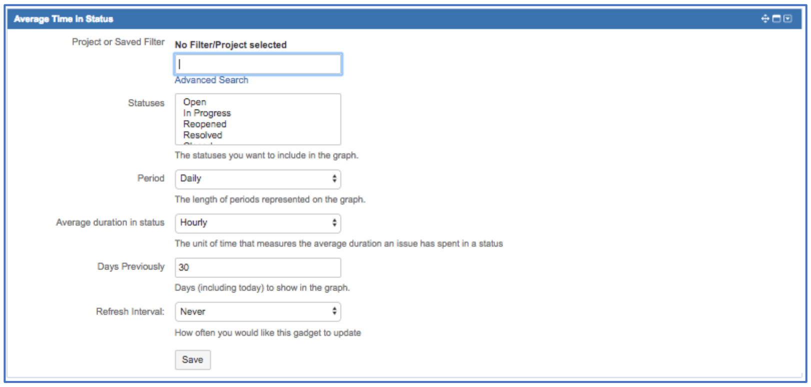
Gadgets let you customize the information that appears on dashboards in JIRA applications.

You can add and remove gadgets from your dashboard.

1. Click add gadget
2. Choose all the gadgets you want. They are added to your board automatically. You'll configure them after.



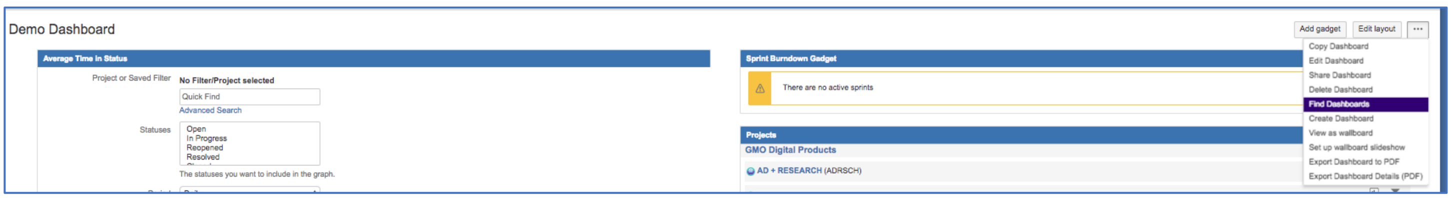
3. Close this window
4. All gadgets you chose are now in your board and now you'll need to configure them
5. Most of the gadgets are based on filters. Create your filters before you create your dashboards and use them in dashboards.



12.1.4 Searching Dashboards

To search for a dashboard

1. Click ••• > **Find dashboard**
2. Enter your search criteria into the search field.



Each shared dashboard indicates its:

1. Current owner. This is the user who created the shared dashboard.
2. List of shares applied to the shared dashboard by its owner.

Jira looks for exact matches to the search terms you enter in the search field. You can use different operators to help you find wildcard or fuzzy matches to your search queries. For example:

- Use the **fuzzy search operator** to help find dashboards that might contain misspellings:

dashboard~

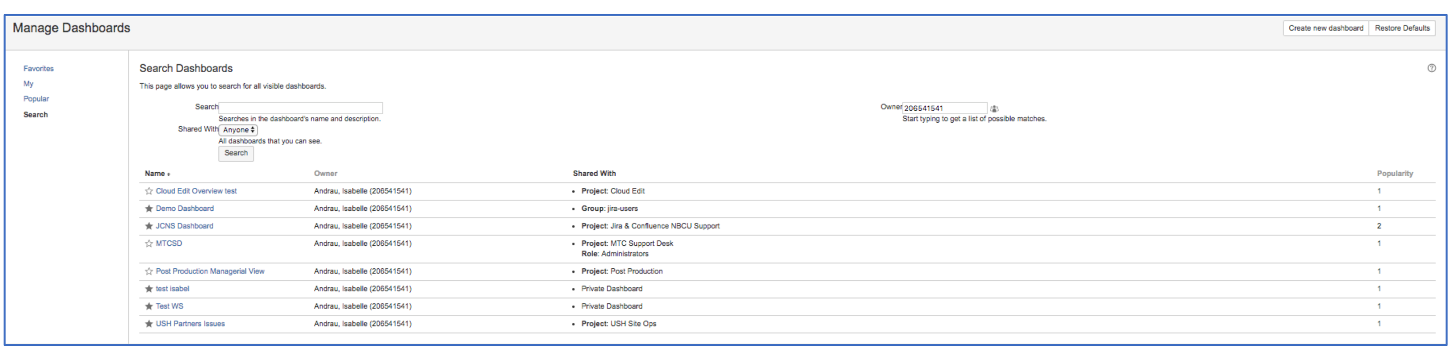
- Use the **wildcard search operator** to help find dashboards that begin with a phrase:

dash*

12.1.5 Favorite Dashboards

If you find a dashboard you use frequently, you can add it to your favorite dashboards so it appears in the sidebar.

1. Find the dashboard and select the star icon next to its name



12.1.6 Recommended Gadgets

The “UPT Master Dashboard” is a good starting point for setting up a dashboard for your team. It contains the following gadgets:

- Sprint Health – This gadget shows the percentage of time elapsed in the sprint alongside high level metrics like amount of work complete and amount of scope change. It is based on a selected scrum board, and so can be configured for a single scrum team.
- Days Remaining in Sprint
- Sprint Burndown – This gadget shows the sprint burndown for the selected filter. For a team scrum board, you should select the same filter that your team’s scrum board is using.
- Average Time in Status – On the UPT Master Dashboard, this gadget is looking at the filter for the Solutioning Kanban Board. It shows the average time that work items are in each state. Configuration for the gadget allows you to select the filter, the amount of time being considered for the average, and the states to include on the chart.
- Time Logging Chart – This shows the time logged by individuals, and is an easy way to check if anyone on your team is behind on logging their time. The gadget can be configured to look at a project (e.g. all of UPT), or at a filter (e.g. a scrum board filter). You can also adjust the time span that is shown (1 week, an entire month, etc).

13 Glossary

Backlog – A backlog is a list of the outstanding user stories, bugs and features for a product or sprint.
Board – The tool that teams use to visualize units of work moving through their specific workflow. It can be adapted for different styles of agile development (e.g. a Scrum board shows work items moving from the product backlog to the sprint backlog whereas a Kanban board typically has a three-step workflow: To Do, In Progress, and Done).
Bug - An error, flaw, mistake, failure, or fault in a computer program that prevents it from behaving as intended (such as producing an incorrect result).
Burndown Chart – A Burndown Chart shows the actual and estimated amount of work to be done in a sprint.
Control Chart – A Control Chart can show the cycle time or lead time for your product, version or sprint.
Cycle Time – Cycle time is the time spent working on an issue – typically, the time taken from when work begins on an issue to when work is completed, but also includes any other time spent working on the issue. For example, if an issue is reopened, worked on, and completed again, then the time for this extra work is added to the cycle time.
Daily stand-up – Also known as a daily scrum, a 15-minute mini-meeting for the software team to sync.
Epic – Captures a large body of work that needs to be broken down into a number of smaller stories. It may take several sprints to complete an epic. The hierarchy for units of work in Jira Software is as follows: *Initiative > Epics > Stories > Subtasks*
Filter – Filters determine what is visible on each of your boards. Using Jira's simple query builder (JQL), you can customize exactly which issues are displayed on your board.
Initiative - a large body of work at the top of the hierarchy. An initiative can last multiple program increments and will include one or more **Epics**.
Issue – An issue is simply a unit of work within Jira that will be traced through a workflow, from creation to completion. It can range from representing a single unit of work, like a simple task or bug, to a larger parent work item to be tracked, like a story or an epic.
Kanban – Kanban is a system for visualizing the flow of work and limiting work in progress. Kanban is not oriented towards sprints, like Scrum development methodology, as it is more ongoing.
Risk - the level of impact on the successful outcome of a work item.
Scrum – Scrum is an Agile development methodology where the product is built in a series of fixed-length iterations called sprints. It gives teams a framework for shipping software on a regular cadence.
Scrum of Scrums – Scrum of Scrums is a means of scaling Scrum to large, multi-team projects. Scrum of Scrums is the Agile version of what is traditionally known as program management.
Sprint – A sprint — also known as an iteration — is a short (ideally two to four week) period in which the development team implements and delivers a discrete product increment, e.g. a working milestone version.
Sprint planning – A team planning meeting that determines what to complete in the coming sprint.
Sprint retrospective – A review of what did and didn't go well with actions to make the next sprint better.
Story – A listing of acceptance criteria needed to deliver a new epic or piece of work. Generally written from the perspective of a user of the system. A commonly used format is: *As a X, I want to Y, so that Z.*
Story Point – A story point is an estimate of the relative complexity of a story.
Swimlane – A swimlane is a means of categorizing issues so that agile teams can see which issues they should work on next.
Sub-Task – A sub-task can be a "child" of any issue type, depending on the Issue Type Scheme of the project.
Task - A task is another out-of-the-box issue type in Jira.
Velocity – The velocity of a team is a measure of how much work that the team can handle within a specific time period, i.e. how much of the product backlog can be completed by the team in a sprint. Velocity can be calculated on the basis of story points, business value, hours, issue count, or any numeric field of your choice.
Workflow – Workflow is the logic that drives movement of a stories / sub-task along their development journey on a scrum or kanban board.