

Jump Start to Jira for Administrators



COURSE OVERVIEW



What you will learn

- Use lean and agile principles
- Differentiate Kanban and scrum
- Configure Jira to match your team's current process
- Jira User Management
- Workflows
- Custom Fields

To succeed here, you need to have

- No knowledge of agile or Jira Is assumed

Jira Cloud vs Jira Server/Data Center

A screenshot of the Jira Cloud interface showing a Kanban board. The board has three columns: 'To do', 'In progress', and 'Done'. There are several cards in each column, each with a title like 'add more A', 'add more B', and 'add more C'. The sidebar on the left shows project navigation and settings.

Cloud

A screenshot of the Jira Server/Data Center interface showing a Kanban board. The board has three columns: 'To do', 'In progress', and 'Done'. The cards are identical to the Cloud version. The sidebar on the left includes additional options like 'Backlog' and 'Report board'.

Server /
Data Center

Jira Cloud- classic vs. next-gen projects

The diagram illustrates the comparison between Jira Cloud classic and next-gen projects.

Classic project: A card-style interface with a blue header bar containing a "Create project" button. It features two sections: "Classic project" (with a folder icon) and "Try a next-gen project" (with a star icon). The "Classic project" section contains the text: "All the power and functionality you expect. Created and managed by your Jira admin." The "Try a next-gen project" section contains the text: "Easy setup and reimagined features. Created and managed by project team members."

Next-gen project interface: Two side-by-side screenshots of the Jira Kanban board. The left screenshot shows the "Kanban board" for a "PROJ board" with four columns: "IN PROGRESS 3", "READY FOR DEVELOPMENT 3", "DEVELOPED 0", and "DONE 0". The right screenshot shows the "Kanban board" for a "PROJ board" with four columns: "IN PROGRESS 3", "READY FOR DEVELOPMENT 3", "DEVELOPED 0", and "DONE 0". Both boards contain items labeled "add item 1", "add item 2", and "add item 3". The left board also includes a "Create issue" button. The top navigation bar for both boards includes "Project", "Issues", "Reports", "Components", "Add item", and "Project settings".

Lab 1- Course Overview

- Decide If you want to do the Cloud or Server version of the Lobs
- Login to Jira

Jira Service Management



What will you learn?

- Describe agile
- Describe Jira
- Identify how Jira relates to an agile mindset
- Create Jira project
- Create a Jira Issue
- Use a project board
- Identify/Jira user types

Agile and Jira Overview



Welcome to the Course!

- Just focus on the basics. No need to be an expert yet.
- Learn the terms first. Important terms will be highlighted like this .
- Your screens may look different. That's okay. The concepts and terms are still the same.
- the course.

BLUF (Bottom Line Up Front)

**Jira Service Management is a tool
to help you track your tickets,
communicate with customers, and
resolve requests.**

Power your Service Desk



IT Support

We can help with any questions regarding your computer.



Facilities

Is something broken? We can help. Raise a request here.



Legal

Legal advice, contract reviews and NDAs



Infrastructure support

Need a cloud service? Raise a request here.



Accounting and finance

Contact us for financial approvals and general queries.



Give kudos

Say thanks to your colleagues, send them a kudos here.



Human resources

We can help with new employee onboarding and general queries.



Marketing

Contact us for marketing campaigns and content.



Data engineering

Ask us for anything data or analytics related.



Research & Insights

We manage all outgoing customer research and VOC.



Global Tax

Home to all your Tax needs.



Security

Contact us for any security reviews or breaches.

Why use a Service Desk Tool?

- **It's fast:** 40% faster task resolution than other service desk providers.
- **It's efficient:** Easily organize, prioritize, and assign tickets.
- **It's easy to use:** Clean, intuitive interface that's simple for both customers and agents.
- **It's affordable:** 80% lower cost than other service desk tools, including a free plan.

Avatars



AGENT
(Probably you)



CUSTOMER



**PROJECT
ADMINISTRATOR**



COLLABORATOR

Issues

An **issue** is an individual unit of work.

The screenshot shows the Jira Service Management interface for the 'IT Support' project. The left sidebar displays navigation links like 'All tickets', 'Starred', and 'Team Priority'. The main area is titled 'All open tickets' and lists several customer requests. One ticket, 'Get IT help' (ITS-36), is highlighted with a yellow border. The table columns include Customer Request, Key, Summary, Reporter, Assignee, Status, Created, and Time to re... . The status for ITS-36 is 'IN PROGRESS'.

| Customer Request | Key | Summary | Reporter | Assignee | Status | Created | Time to re... |
|------------------------|--------|------------------------------------------------------|-------------------|----------------|---------------------|-----------|---------------|
| Get IT help | ITS-36 | My VPN is not connecting to the network | Melissa Brimer | Melissa Brimer | IN PROGRESS | 02/Dec/21 | 7h 53m |
| Fix an account problem | ITS-37 | My password expired and I am locked out of my email. | Charlie Teamerson | Unassigned | WAITING FOR SUPPORT | 02/Dec/21 | 7h 54m |
| Get IT help | ITS-38 | Computer won't connect to wifi | Ryan Lee | Unassigned | WAITING FOR SUPPORT | 02/Dec/21 | 7h 55m |
| Request new software | ITS-39 | Need software license for Adobe Creative Cloud | Alana Grant | Andrew DeBell | WAITING FOR SUPPORT | 02/Dec/21 | 7h 55m |
| Fix an account problem | ITS-40 | Password expired and need help resetting it | Charlie Teamerson | Andrew DeBell | IN PROGRESS | 02/Dec/21 | 7h 56m |
| Get IT help | ITS-41 | My VPN is not connecting to the network | Ryan Lee | Andrew DeBell | IN PROGRESS | 02/Dec/21 | 7h 56m |
| Get IT help | ITS-42 | Laptop keeps restarting unexpectedly | Alana Grant | Melissa Brimer | IN PROGRESS | 02/Dec/21 | 7h 57m |

Interacting with Issues

Requests

Agents see this
(called an issue in their view)

The screenshot shows a service management application with a sidebar on the left containing navigation links like 'Dashboard', 'Issues', 'Projects', 'Trans.', 'Entitlements', 'People', 'Helpdesk', 'Audit', and 'Devices'. The main area displays an issue card for 'Laptop keeps restarting unexpectedly'. The card includes sections for 'Details' (with a 'Show details' button), 'Actions' (with a 'Create ticket' button), 'Comments' (with a 'Add comment' button), and 'Activity' (with a 'View activity' button). The right side of the screen shows a detailed view of the issue, including tabs for 'Details', 'Actions', 'Comments', and 'Activity'. The 'Details' tab contains fields for 'Subject' (Laptop keeps restarting unexpectedly), 'Owner' (John Doe), 'Last updated' (1 hour ago), 'Priority' (Medium), 'Status' (Open), 'Last update' (1 hour ago), 'Last modified by' (John Doe), 'Last modified at' (1 hour ago), 'Last updated by' (John Doe), 'Last updated at' (1 hour ago), 'Last modified by' (John Doe), 'Last modified at' (1 hour ago), and 'Last updated by' (John Doe). The 'Actions' tab lists 'Assign ticket' (John Doe), 'Escalate ticket' (John Doe), 'Close ticket' (John Doe), and 'Comment' (John Doe). The 'Comments' tab shows a single comment from John Doe: 'Laptop keeps restarting unexpectedly' posted 1 hour ago. The 'Activity' tab shows a list of recent activities: 'John Doe created ticket' (1 hour ago), 'John Doe assigned ticket to John Doe' (1 hour ago), and 'John Doe updated ticket' (1 hour ago).

Customers see this
(called a request in their view)

The screenshot shows a customer support application with a sidebar on the left containing navigation links like 'Dashboard', 'Tickets', 'Entitlements', 'People', 'Helpdesk', 'Audit', and 'Devices'. The main area displays a ticket card for 'Laptop keeps restarting unexpectedly'. The card includes sections for 'Details' (with a 'Show details' button), 'Actions' (with a 'Add comment' button), and 'Comments' (with a 'View comments' button). The right side of the screen shows a detailed view of the ticket, including tabs for 'Details', 'Actions', 'Comments', and 'Comments'. The 'Details' tab contains fields for 'Subject' (Laptop keeps restarting unexpectedly), 'Owner' (John Doe), 'Last updated' (1 hour ago), 'Priority' (Medium), 'Status' (Open), 'Last update' (1 hour ago), 'Last modified by' (John Doe), 'Last modified at' (1 hour ago), 'Last updated by' (John Doe), 'Last updated at' (1 hour ago), and 'Last modified by' (John Doe). The 'Actions' tab lists 'Assign ticket' (John Doe), 'Escalate ticket' (John Doe), 'Close ticket' (John Doe), and 'Comment' (John Doe). The 'Comments' tab shows a single comment from John Doe: 'Laptop keeps restarting unexpectedly' posted 1 hour ago.

Issues and Requests

An **IT Support team** might have an issue like this:

My laptop keeps restarting unexpectedly



GD-1



A **Human Resources team** might have an issue like this:

Set up onboarding kit for new manager



GD-24



A **Legal team** might have an issue like this:

Review MSA contract and provide feedback



GD-2



W

Service Project and Queue

The screenshot shows the Jira Service Management interface for the 'IT Support' service project. The left sidebar is a navigation menu with several items highlighted by yellow boxes and arrows:

- IT Support** (Service project)
- Queues**
- Service requests**
- Incidents**
- Problems**
- Changes**

The main area displays a table titled 'Open service requests'. The columns are: Key, Summary, Reporter, Assignee, Status, Created, and Time to resol... (with a dropdown arrow). One row is selected, highlighting the entry for 'ITS-42' with the summary 'Laptop keeps restarting unexpectedly'. The status for this request is 'IN PROGRESS', assigned to 'Melissa Brimer', and it was created on '02/Dec/21'.

| Key | Summary | Reporter | Assignee | Status | Created | Time to resol... |
|--------|------------------------------------------------------|------------------|----------------|-------------|-----------|------------------|
| ITS-42 | Laptop keeps restarting unexpectedly | Alana Grant | Melissa Brimer | IN PROGRESS | 02/Dec/21 | 7h 57m II |
| ITS-43 | My password expired and I am locked out of my email. | Charlie Townsend | Melissa Brimer | In Progress | 02/Dec/21 | 7h 58m 30s |
| ITS-44 | Laptop keeps restarting unexpectedly | Alana Grant | Melissa Brimer | In Progress | 02/Dec/21 | 7h 58m 30s |
| ITS-45 | My WiFi is not connecting to the network | Ryan Lee | Andrew Dallas | In Progress | 02/Dec/21 | 7h 58m 30s |
| ITS-46 | Processor overheat and need help removing it | Charlie Townsend | Andrew Dallas | In Progress | 02/Dec/21 | 7h 58m 30s |
| ITS-47 | Need software license for Adobe Creative Cloud | Alana Grant | Andrew Dallas | In Progress | 02/Dec/21 | 7h 58m 30s |
| ITS-48 | Computer won't connect to WiFi | Ryan Lee | Unassigned | In Progress | 02/Dec/21 | 7h 58m 30s |
| ITS-49 | My password expired and I am locked out of my email. | Charlie Townsend | Unassigned | In Progress | 02/Dec/21 | 7h 58m 30s |
| ITS-50 | My WiFi is not connecting to the network | Melissa Brimer | Melissa Brimer | In Progress | 02/Dec/21 | 7h 58m 30s |

Make sure you are on the Right Project

Jira Service Management Your work Projects Filters Dashboards People Insight Apps Create Search

IT Support Service project

Queues

- Service requests
- Incidents
- Problems
- Changes

OPERATIONS

- Change calendar
- Services
- Alerts
- On-call

KNOWLEDGE

- Knowledge base
- Reports

CHANNELS & PEOPLE

- Channels
- Customers

All open tickets

| Customer Requ... | Key | Summary | Reporter | Assignee | Status | Created ↑ | Time to resol... |
|------------------------|--------|------------------------------------------------------|-------------------|----------------|---------------------|-----------|------------------|
| Fix an account problem | ITS-47 | My password expired and I am locked out of my email. | Charlie Teamerson | Unassigned | WAITING FOR SUPPORT | 03/Dec/21 | 8h II |
| Get IT help | ITS-48 | Computer won't connect to wifi | Ryan Lee | Unassigned | WAITING FOR SUPPORT | 03/Dec/21 | 8h II |
| Request new software | ITS-49 | Need software license for Adobe Creative Cloud | Alana Grant | Andrew De... | WAITING FOR SUPPORT | 03/Dec/21 | 8h II |
| Fix an account problem | ITS-50 | Password expired and need help resetting it | Charlie Teamerson | Andrew De... | IN PROGRESS | 03/Dec/21 | 8h II |
| Get IT help | ITS-51 | My VPN is not connecting to the network | Ryan Lee | Melissa Bri... | IN PROGRESS | 03/Dec/21 | 8h II |
| Get IT help | ITS-52 | Laptop keeps restarting unexpectedly | Alana Grant | Melissa Bri... | IN PROGRESS | 03/Dec/21 | 8h II |
| Fix an account problem | ITS-53 | My password expired and I am locked out of my email | Charlie Teamerson | Sumedh S | IN PROGRESS | 03/Dec/21 | 8h II |

Customers POV

Queues

Filtering Issues by Types

The screenshot shows the Jira Service Management interface. On the left, there's a sidebar with a search bar and four filter options: Service requests (selected), Incidents, Problems, and Changes. A yellow box highlights the 'Service requests' option. An arrow points from the text 'Filter issues by request type' to this highlighted box. The main area displays a table of 'Open service requests' with columns: Key, Summary, Reporter, Assignee, Status, Created, and Time to Res. There are seven rows of data, each representing a different service request.

| Key | Summary | Reporter | Assignee | Status | Created | Time to Res. |
|----------|-----------------------------------------------------|-------------------|--------------|-----------------------|------------|--------------|
| ITSM-101 | My password expired and I am locked out of my email | Charlie Isensemen | Unassigned | In Progress | 2024-02-27 | 0h 30m |
| ITSM-102 | My phone is not connecting to the network | Ryan Lee | Unassigned | In Progress | 2024-02-27 | 0h 30m |
| ITSM-103 | My phone is not connecting to the network | Ryan Lee | Unassigned | In Progress | 2024-02-27 | 0h 30m |
| ITSM-104 | Password expired and need help resetting it | Charlie Isensemen | Andrew Delfi | In Progress | 2024-02-27 | 0h 30m |
| ITSM-105 | Need software license for Adobe Creative Cloud | Alice Green | Andrew Delfi | Not Started (Summary) | 2024-02-27 | 0h 30m |
| ITSM-106 | Computer won't connect to wifi | Ryan Lee | Unassigned | Not Started (Summary) | 2024-02-27 | 0h 30m |
| ITSM-107 | My password expired and I am locked out of my email | Charlie Isensemen | Unassigned | Not Started (Summary) | 2024-02-27 | 0h 30m |

Customer Requests

The Queue is Populated

Who gets the ticket?

Make sure the status is updated!

Because this is what the customer sees

The screenshot shows a Jira Service Management ticket interface. At the top, a purple banner says "We've collapsed your details view to help you focus on the work that matters most." Below this, the ticket details are shown in a collapsed state.

Ticket Details:

- Help Center / IT Support / ITS-45
- Title:** My VPN is not connecting to the network
- Creator:** Andrew DeBell (raised on 10/Dec/21 7:24 PM)
- Status:** IN PROGRESS
- Activity:**
 - Melissa Brimer Today 9:15 PM: We're starting work on this request now!
 - Automatic response Today 9:15 PM: Your request status has changed to In Progress.
- Request type:** Get IT help
- Shared with:** Andrew DeBell (Creator)
- Actions:** Notifications on, Resolve this issue, Cancel request

Bottom of the screen:

- Add a comment input field
- Powered by Jira Service Management

Topics

- Agile Overview
- Jira Overview
- Projects, issues and boards



What is Agile?

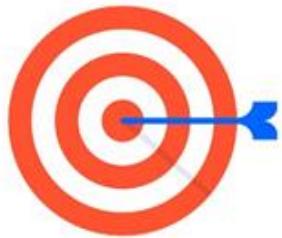
A way of getting things done.

An empirical approach to project management

Continuously develop the plan, process and product

A mindset

Why Agile?



Effectiveness

Perform better than traditional projects



Empower the Team

Leverage team knowledge and increase job satisfaction



Manage Complexity

Simple project management approach to increasing complexity

What is an agile mindset?

- A growth/continuous improvement way of working
- Allowing the data to change your approach
- Uses agile techniques to accomplish work



What is an agile mindset?

- For an agile team to perform its best, all team members must have an agile mindset



The agile coach



BROWSE TOPICS

- [Agile manifesto](#)
- [Scrum](#)
- [Kanban](#)
- [Agile project management](#)
- [Product Management](#)
- [Agile at scale](#)
- [Software development](#)

What is Agile?

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

Topics

- Agile Overview
- Jira Overview
- Projects, issues and boards



What is Jira

- A tool used to help teams perform, visualize and manage work
- Models the team's current processes/workflows

Why Jira?

- Leverage project management technology, allowing teams to focus on their work
- Facilitates planning, prioritizing, organizing and completing work
- Visualizes work using project boards, reports and dashboards
- Facilitates team communication

How does Jira relate to an agile mindset?

- Jira is a tool that teams can use to model and execute their agile processes

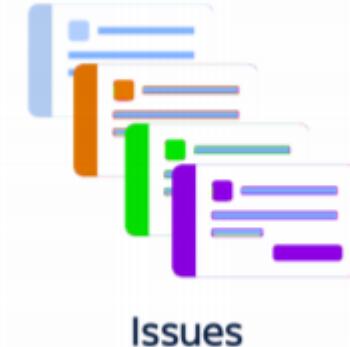
Topics

- Agile Overview
- Jira Overview
- Projects, issues and boards



What is a Jira issue?

- An item of work (work item) identified by the team
- An issue has an associated type (for example, story, task, bug)
- The details of the issues are known as fields



What is a Jira project?

- A collection of related issues
- A team “to do” List
- Can have a fixed end date or be an ongoing project
- A project has an associated type (for example, Kanban, scrum)



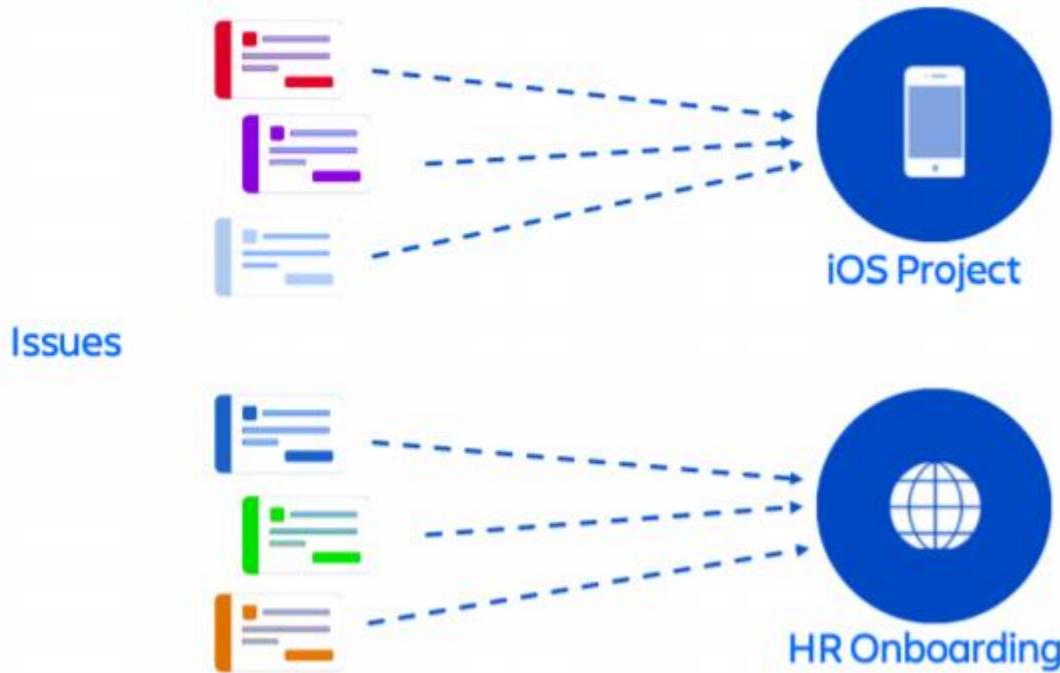
Issue Key



Jira automatically assigns a unique *issue key* to created issues

`<issue_key> = <project_key>-<issue_number>`

Each issue belongs to one project



What is a project board?

- A two-dimensional “to do” List
- A way to visualize issues
- A visualization of the team's process/workflow
- Displays Issues as cards



Jira User Types



Jira Administrator

..... Configures the Jira instance
for all users



Jira Project
Administrator

..... Can configure a Jira project to
match the team's process



Team Member

..... Works on projects

Takeaways

- Agile Is a way of working
- Jira Is a tool teams use to manage and visualize work
- Jira can be configured to match a team's continuously improving processes
- A Jira Issue is on Item of work Identified by the team
- Project boards visualize a team's work
- The main types of Jira users are Jira administrators, Jira project administrators and team members

Lab 2 -Agile and Jira Overview

- Create a project
- Create issues

Visualize work using project boards



What will you learn?

- Describe the Importance of visualizing work
- Describe common workflows
- Differentiate Jira boards and workflows
- Describe the purpose of an issue's status field
- Configure board columns

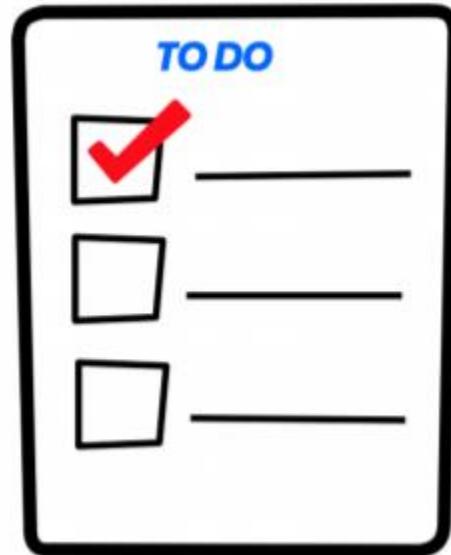
Topics

- Visualizing work
- Workflows
- Jira Boards and workflows
- Configuring board columns



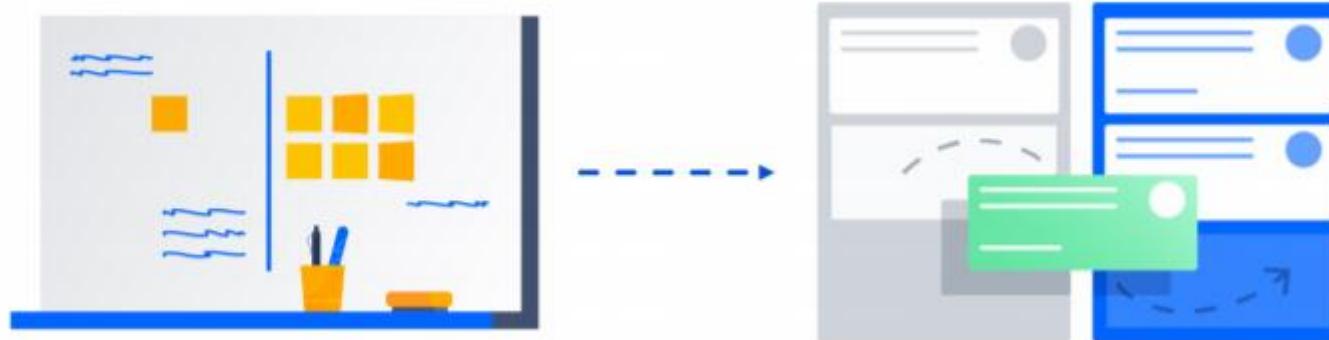
Visualizing work: a “to do” list

- Reminds you
- Focuses you
- Sets priorities
- Tracks progress



Visualizing work: a board

- A principle of agile projects is to "visualize work"
- A board is an agile tool used to visualize and manage work



Visualizing work: reports and dashboards



Reports



Dashboards

Why Visualize Work?

- To easily **see** the work of the project
 - Allows anyone to see the true current state of the project
 - Organizes and focuses the team
- To **manage** things
 - Easy to add and prioritize the work of the project
 - Easy to update work items
- To **improve** the team's way of working
 - Can visually identify problems



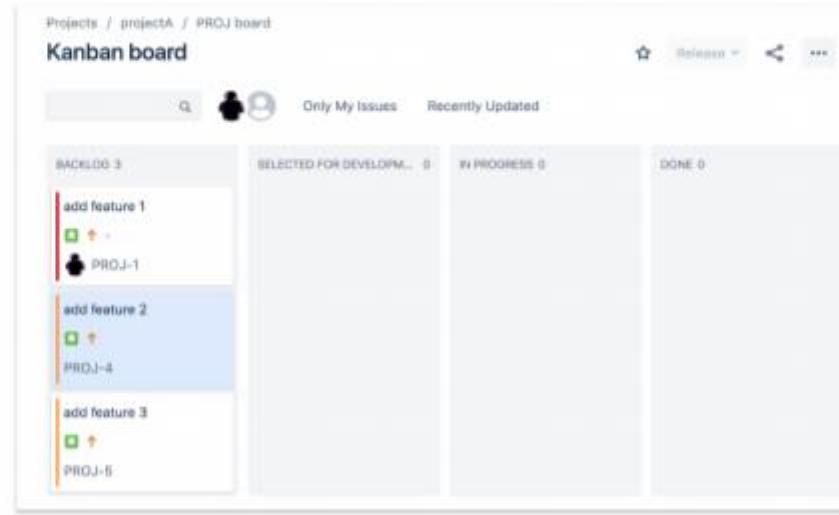
Topics

- Visualizing work
- **Workflows**
- Jira Boards and workflows
- Configuring board columns

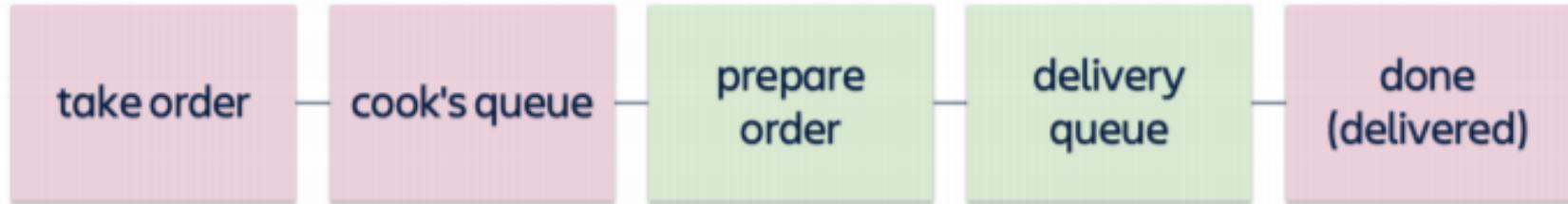


Workflows

- The set of columns of a board represent a **workflow** (or process) for completing the work of an issue
- Workflows are broken down into **statuses** (or steps)

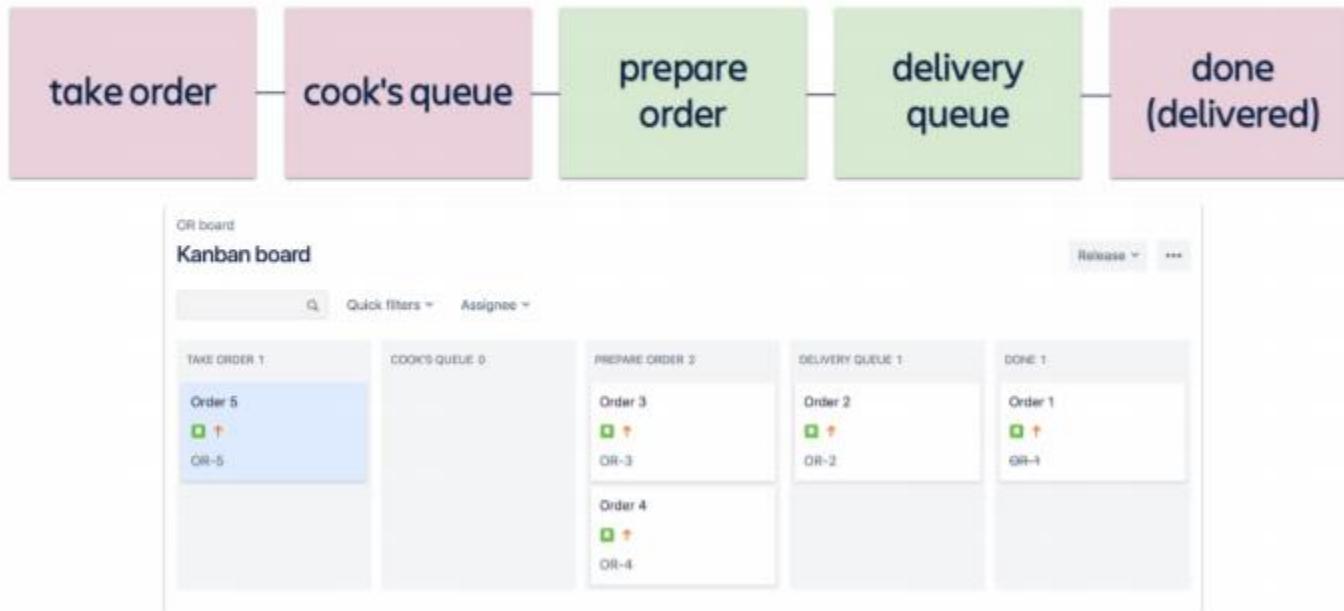


Example workflow: restaurant order and delivery



Boards VS Workflows

- A team works using a board
- The board's structure is defined by an underlying workflow



Topics

- Visualizing work
- Workflows
- **Jira Boards and workflows**
- Configuring board columns



How are boards created?

- Automatically
- Create additional boards at any time

The screenshot shows a Jira Kanban board for projectA / PROJ board. The board has three columns: BACKLOG (3 items), SELECTED FOR DEVELOPME... (0 items), and IN PROGRESS (0 items). A red arrow points from the 'Create board' option in the context menu to the 'Create board' button on the board header.

Projects / projectA / PROJ board

Kanban board

Only My Issues Recently Updated

BACKLOG 3

add item 1

PROJ-1

SELECTED FOR DEVELOPME... 0

IN PROGRESS 0

Release

Board settings

Create board

Hide menus

Show detail view

Open issues in sidebar

Expand all swimlanes

An issue's status field

- Every project automatically has one or more associated workflows
- The status field for each issue must be set to a workflow's status

The screenshot shows a Jira interface for creating a new issue. At the top left, there is a checked checkbox labeled "PROJ-1". On the right side, there are icons for "Give feedback", a blue circular button with the number "1", and a share icon. Below these, the title "add item 1" is displayed. To the right of the title is a red arrow pointing towards a dropdown menu. The dropdown menu is titled "Backlog" and contains three items: "SELECTED FOR DEVELOPMENT" (highlighted in blue), "IN PROGRESS", and "DONE".

PROJ-1

Give feedback 1

add item 1

Backlog

SELECTED FOR DEVELOPMENT

IN PROGRESS

DONE

Boards and status

- Boards are a view of issues arranged by status
- Moving an issue changes the value of its status field

Projects / projectA / PROJ board

Kanban board

Only My Issues Recently Updated

| BACKLOG 3 | SELECTED FOR DEVELOPMENT 0 | IN PROGRESS 0 | DONE 0 |
|----------------------|----------------------------|---------------|-------------------------------------------------------------------------------|
| add item 1 PROJ-1 | | | We're only showing recently modified issues. Q Looking for an older issue? |
| add item 2 PROJ-2 | | | |
| add item 3 PROJ-3 | | | |

Topics

- Visualizing work
- Workflows
- Jira Boards and workflows
- Configuring board columns



Adding a board column

Projects / projectA / PROJ board

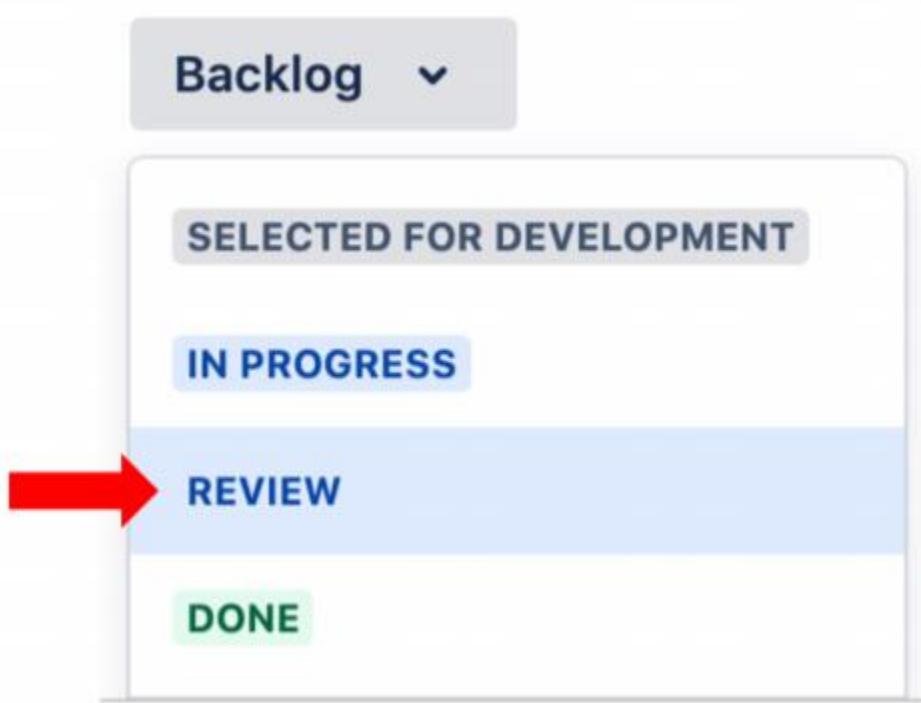
Kanban board

Release   

  Only My Issues Recently Updated

| BACKLOG 3 | SELECTED FOR DEVE... 0 | IN PROGRESS 0 | REVIEW 0 | DONE 0 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|---------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>add item 1</p> <p> </p> <p>PROJ-1</p> | | | | <p>We're only showing recently modified issues.</p> <p> Looking for an older issue?</p> |
| <p>add item 2</p> <p> </p> <p>PROJ-2</p> | | | | |
| <p>add item 3</p> <p> </p> <p>PROJ-3</p> | | | | |

Viewing the new status



Takeaways

- A board is a two-dimensional way to visualize the work of a team
- In Jira, a workflow is often represented using a board
- Board columns usually map to the status field of Issues
- Board columns can be added or removed to match the team's desired process

Lab 3 - Visualize Work Using Boards

- Move Issues through a workflow
- Assign an Issue
- Add a Review column bathe board
- Explore the difference between Jira project administrators and standard users

Enrich Issues



What will you learn?

- Identify ways that Issues can be enriched with Information
- Describe the benefits of using Issue types
- Describe subtasks
- Use Labels to organize Issues
- Introduce Integration with version control and build systems

Topics

- Enrich issues
- Issue types
- Labels
- Developer integration overview



Issues contain work-related information

Issue

Summary: Check network jacks

Description: Each network jack in the new building needs to be checked for signal strength.

Type:  Task

Assignee:



Helena

Priority:  Critical

Status:  IN PROGRESS

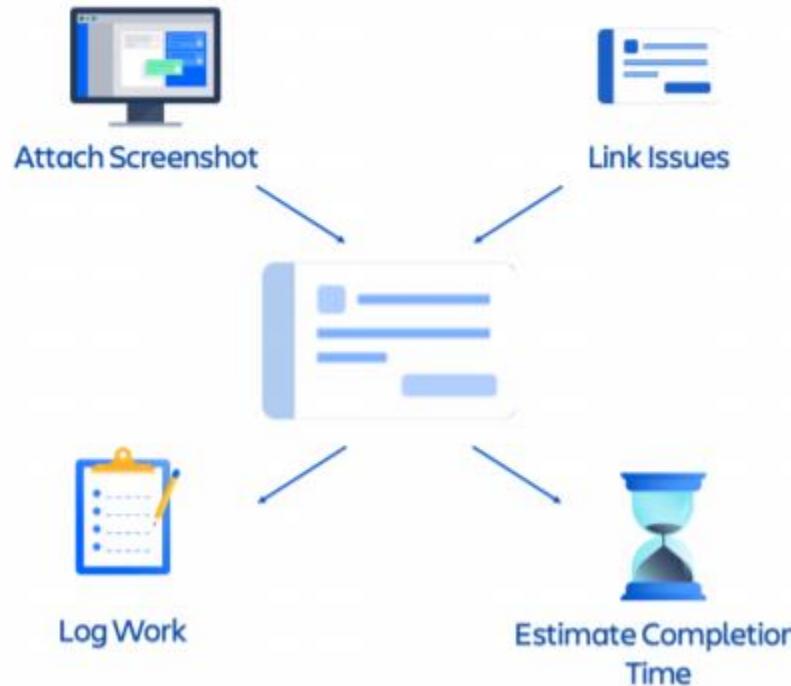
Reporter:



Oliver

Comments: Helena needs the network diagram from IT.

Enriching issues



Mention team members



Topics

- Enrich issues
- Issue types
- Labels
- Developer integration overview



The issue type field

- **Epic** - a big issue that can contain issues
- **Story** - requirement from the user's perspective
- **Task** - team work item
- **Bug** - a flaw that needs to be fixed
- **Subtask** - a child of another issue

Create issue

Project*

projectA (PROA)

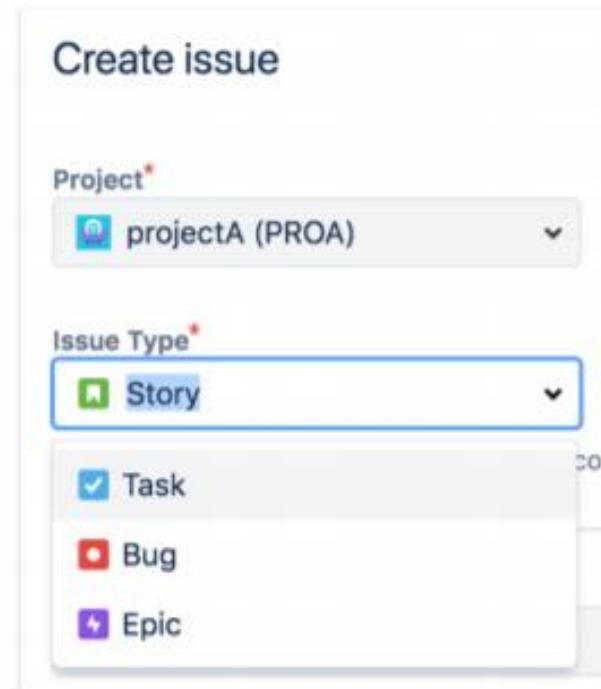
Issue Type*

Story

Task

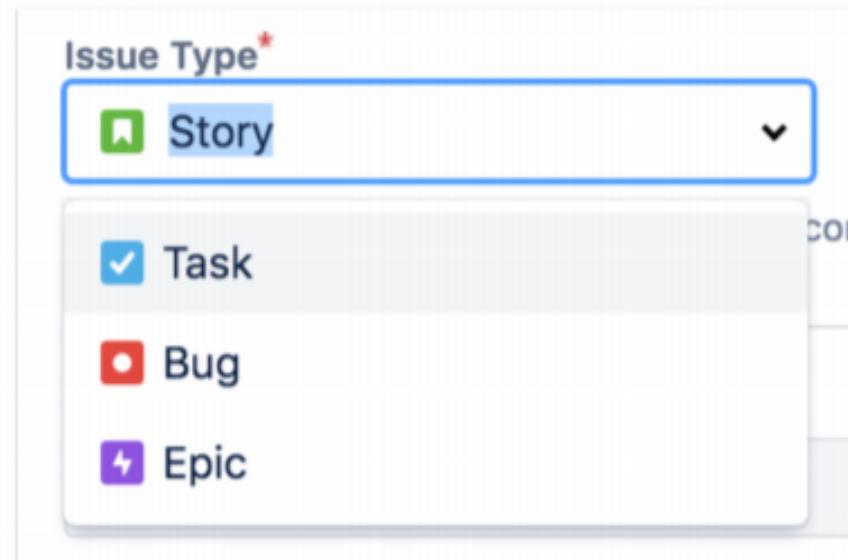
Bug

Epic

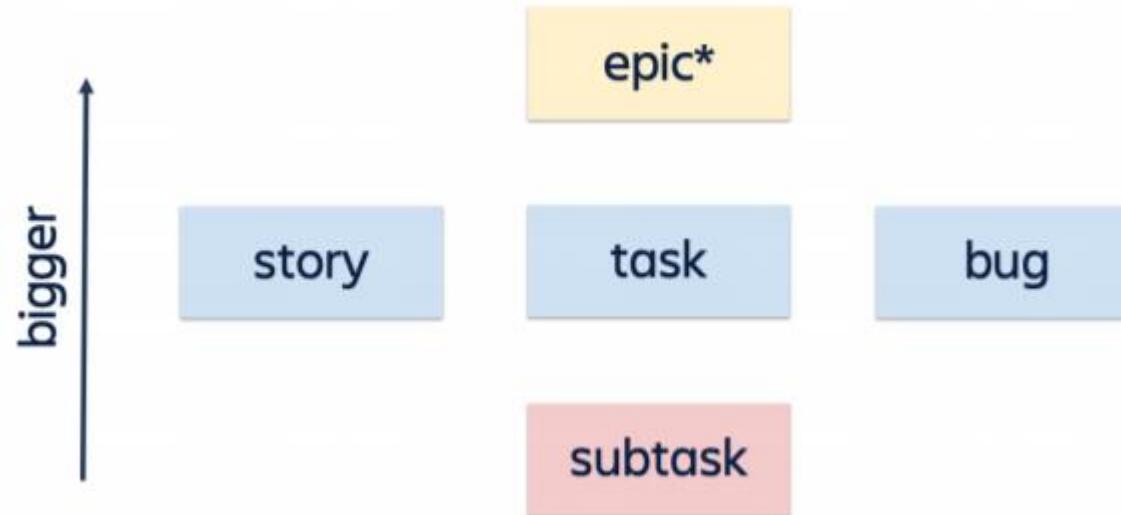


Why issue types

- Supports different types of work
- Each type can have different fields, screens and workflows
- Can report on types separately



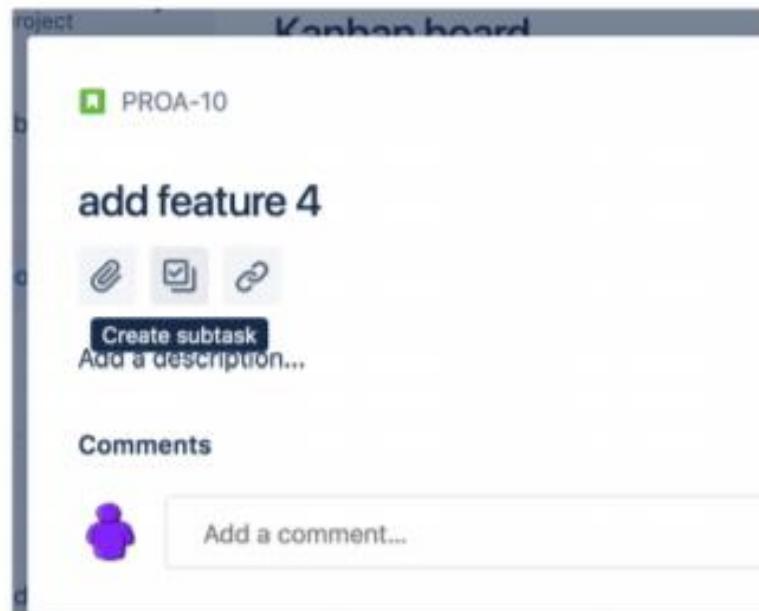
Jira's issue type hierarchy



* Epics are discussed later

Subtasks

- An issue type that must have a parent issue
- Allow an issue to be broken down into individually manageable tasks
- Can be more technical than the parent issue



Subtask characteristics

- Have their own issue key and field values
- Have independent workflow status

The screenshot shows a Jira subtask creation dialog. At the top, it displays the parent issue key 'PROA-10' and the subtask title 'add feature 4'. Below the title are three icons: a person icon, a checklist icon, and a gear icon. A placeholder text 'Add a description...' is present. The 'Subtasks' section lists two subtasks: 'PROA-17 add feature 4a' and 'PROA-18 add feature 4b', both currently in the 'BACKLOG' status. A large text input field at the bottom contains the placeholder 'What needs to be done?'. At the bottom right are 'Create' and 'Cancel' buttons.

Topics

- Enrich issues
- Issue types
- Labels
- Developer integration overview



Labels

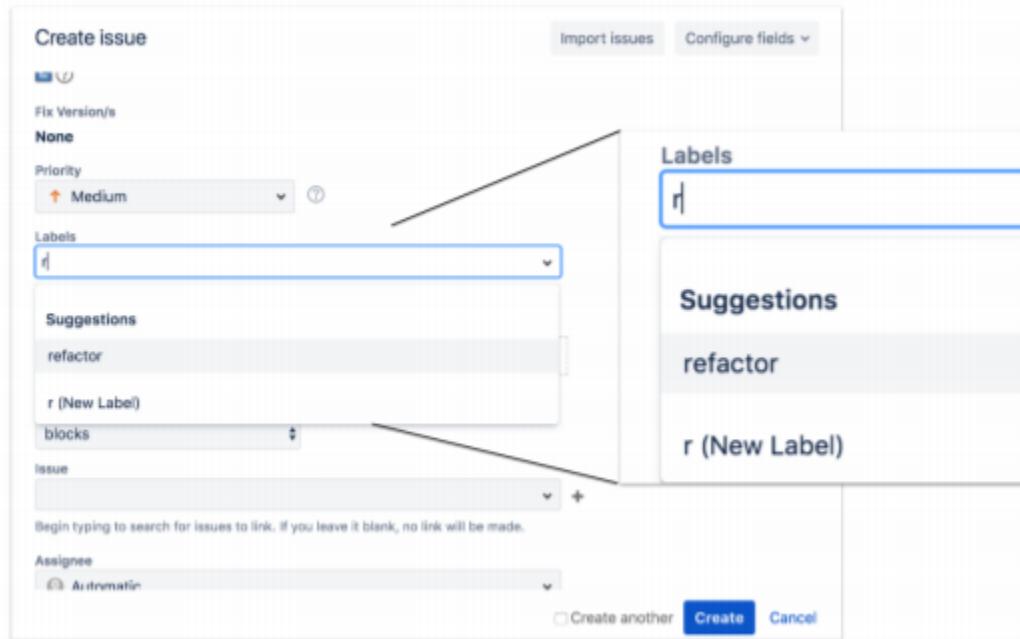
A field used to categorize and search for issues

The screenshot shows a Jira issue card for 'PROJ-6'. The card has the following fields:

- Summary:** complete task 1
- Status:** Selected for Development
- Assignee:** Unassigned
- Labels:** refactor, database
- Priority:** Medium
- Reporter:** Steve Byrnes
- Comments:** Add a comment...
- Subtasks:** PROJ-10: complete task 1a (Backlog)

Two labels, 'refactor' and 'database', are highlighted with blue boxes and arrows pointing to them from the text 'Labels' on the right.

Adding or creating a labels



Searching for a Labeled issues

Click on a label to search for all issues with this label

The image shows a Jira interface. On the left, a new issue is being created for 'PROJ-6' with the title 'complete task 1'. The 'Labels' field contains 'database refactor'. On the right, a 'Search' results page displays the query 'labels = "refactor"' with two results: 'PROJ-10 PROJ-6 / complete task 1a' and 'PROJ-6 complete task 1'. A red arrow points from the 'refactor' label in the issue creation screen to the 'refactor' label in the search results.

PROJ-6

complete task 1

Add a description...

Subtasks:

Comments

Status: Selected for Development

Assignee: Unassigned

Labels: database refactor

Priority: Medium

Reporter: Steve Byrnes

Show more

Created 6 days ago Updated 8 minutes ago

Labels

database refactor

Search Save as

labels = "refactor"

1-2 of 2

T Key Summary

PROJ-10 PROJ-6 / complete task 1a

PROJ-6 complete task 1

1-2 of 2

Topics

- Enrich issues
- Issue types
- Labels
- **Developer integration overview**



The issue detail development panel

Issue detail view
for selected issue



Development

4 commits

Latest 2 days ago

1 pull request MERGED

Updated 2 days ago

1 build ✓

Latest 2 days ago

Deployed to QA

Create a branch or view existing
branches related to this issue



[Create branch](#)

Integration works through the issue key

Using a Commit Message

Include an issue key in a commit message
“Initial commit – TIS-498”

Using a Branch Name

Include an issue key in a branch name
“Feature branch TIS-498”

For Pull Requests

Include an issue key in a pull request title, or Jira can use the issue key from an associated commit or branch

For Builds and Deployments

Jira uses the issue key associated with a commit in the build

Takeaways

- An Issue contains a diverse set of fields that are used to add Information to the Issue
- Issues can facilitate team communication with comments and @mentions
- Issue types can have unique fields, screens and workflows
- Subtasks are children of another issue type
- Subtasks have their own issue key and field values
- Labels can be used to categorize and search for Issues
- Jira can be integrated with version control and/or build systems to improve developer-related communication

Lab 4- Enrich Issues

- Add Information to an Issue
- Use team-related Issue features
- Create Issue of different types
- Create subtasks
- Add Labels to issues

Kanban Meth



What will you learn?

- Describe the Kanban method
- Describe the importance of flow
- Identify the purpose of work-in-progress limits
- Differentiate pull vs. push processes
- Identify reasons to separate the backlog from the board

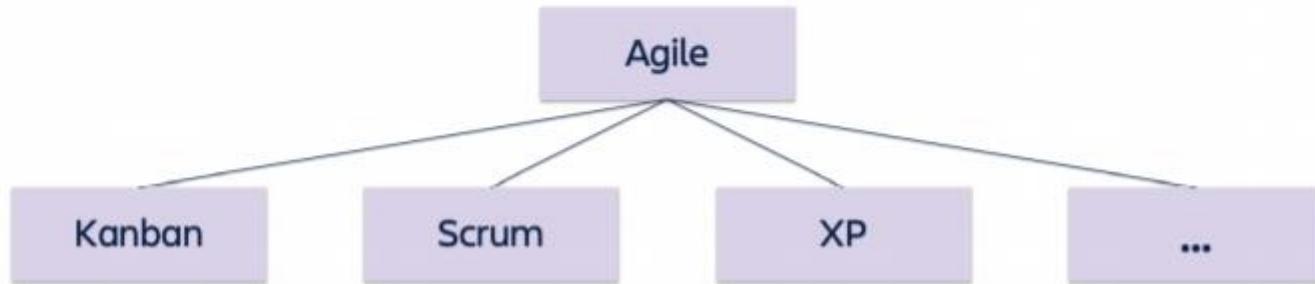
Topics

- Kanban method overview
- Improving flow
- Pull VS Push
- Separate Backlog



Agile methods

- Agile is a way of thinking (mindset) and working
- An agile method (or framework) is an approach to implementing agile
- Common agile methods include kanban and scrum
 - Each embody core principles of agile
 - These are often combined



What is Kanban method?

- An agile method used to manage a continuous queue of work items
- Commonly used ideas:
 - Visualize work
 - Remove process bottlenecks to Improve "flow" of value
 - Limit work in progress/ small batch size
 - Pull work rather than push (where it makes sense)
 - Continuously prioritize work items

Why choose the Kanban method

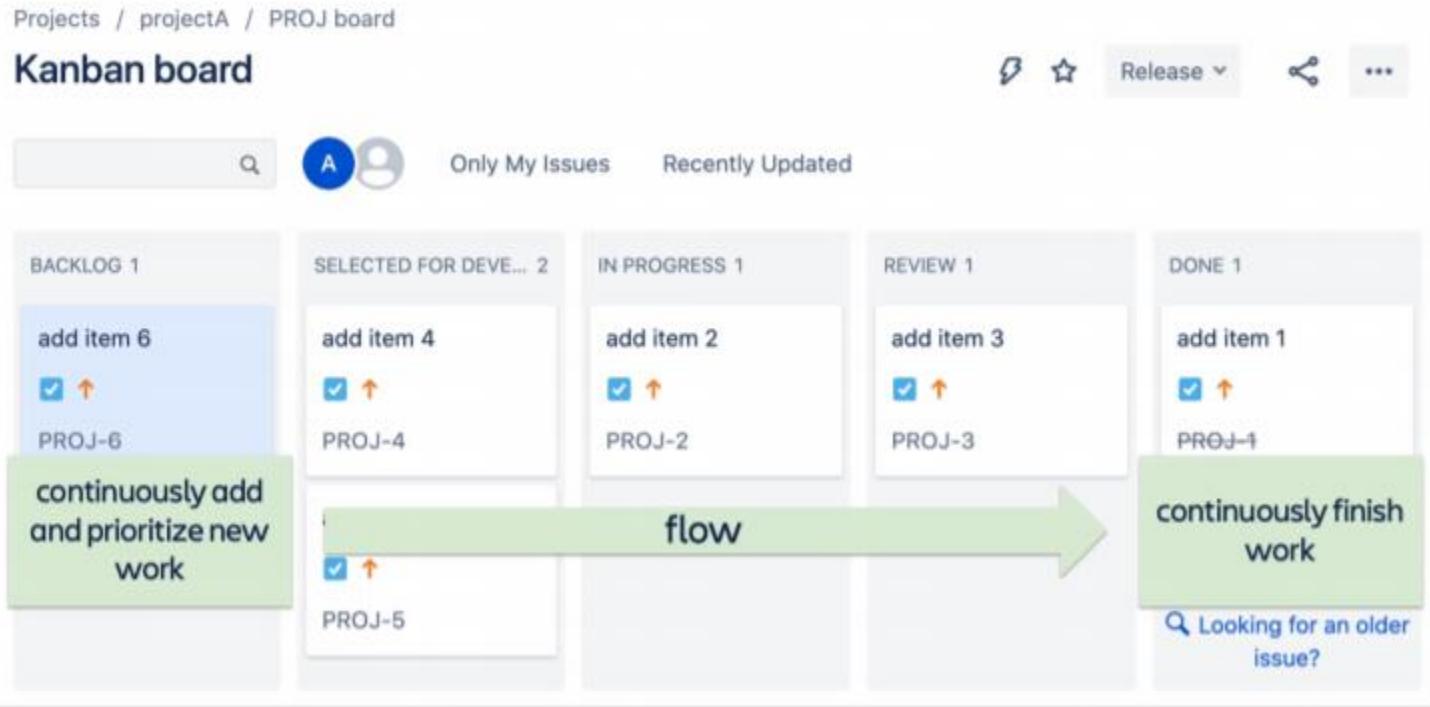
- Very Lightweight and efficient
- Evolutionary approach of transforming to agile
- Works well If the workflow is service-oriented
- operations
- support
- maintenance development
- new hire funnel
- Supports multi-team and multi-project workflows

Topics

- Kanban method overview
- Improving flow
- Pull VS Push
- Separate Backlog



Continuous flow of work items



Improving flow- limit work in progress (WIP)

- How?
 - Specify the minimum and/or maximum number of uses allowed in certain project board columns
- Why?
 - Better flow
 - Limits waste
 - Promotes teamwork

Column under minimum limit

Projects / projectA / PROJ board

Kanban board

Only My Issues Recently Updated

| BACKLOG 2 | SELECTED FOR DEVE... 1 | IN PROGRESS 1 | REVIEW 1 | DONE 1 |
|----------------------|------------------------|----------------------|----------------------|----------------------|
| add item 5 PROJ-5 | add item 4 PROJ-4 | add item 2 PROJ-2 | add item 3 PROJ-3 | add item 1 PROJ-1 |

Min 2

IN PROGRESS 1

REVIEW 1

DONE 1

We're only showing recently modified issues.

Looking for an older issue?

Column over maximum limit

Projects / projectA / PROJ board

Kanban board

The screenshot shows a Jira Kanban board for projectA. The board has five columns: BACKLOG, SELECTED FOR DEVEL..., IN PROGRESS, REVIEW, and DONE. A red arrow points down to the REVIEW column, which is currently at its maximum capacity of 2 items. The DONE column is empty.

| BACKLOG | SELECTED FOR DEVEL... | IN PROGRESS | REVIEW | DONE |
|----------------------|-----------------------|-------------|----------------------|------|
| add item 6 PROJ-6 | add item 4 PROJ-4 | | add item 1 PROJ-1 | |
| | add item 5 PROJ-5 | | add item 2 PROJ-2 | |
| | | | add item 3 PROJ-3 | |

Only My Issues Recently Updated

Release ...

Max 2

We're only showing recently modified issues.

Looking for an older issue?

What should WIP limits be set to?

- Could start with no WIP limits
- Add WIP limits as the process shows problems
- Could set WIP limits to discourage multitasking
- Could set WIP limits on steps that the team neglects

Projects / projectA / PROJ board

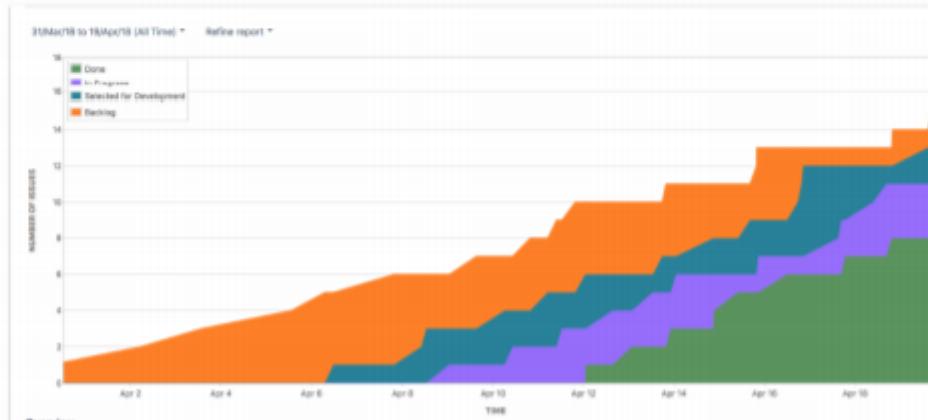
Kanban board

Only My Issues Recently Updated

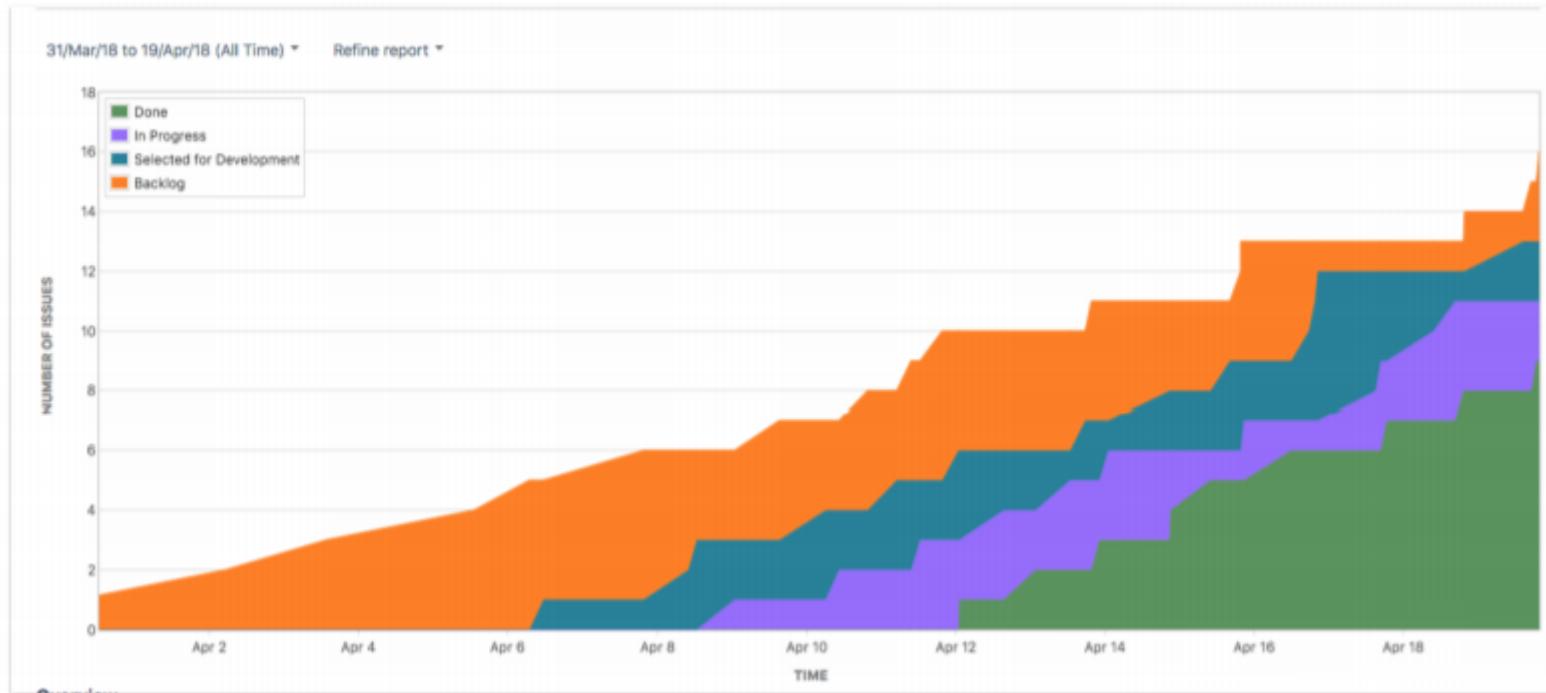
| BACKLOG 1 | SELECTED FOR DEVELOP... 2 Min 2 | IN PROGRESS 1 | REVIEW 2 Max 2 | DONE 6 |
|----------------------|---------------------------------|----------------------|----------------------|-----------------------------------------------------------------------------|
| add item 6 PROJ-6 | add item 4 PROJ-4 | add item 3 PROJ-3 | add item 1 PROJ-1 | We're only showing recently modified issues. Looking for an older issue? |
| | add item 5 PROJ-5 | | add item 2 PROJ-2 | |

What agile reports

- Visualize the work
- Promote transparency
- Aid troubleshooting and continuous improvement
- Aid planning and estimating



Cumulative flow



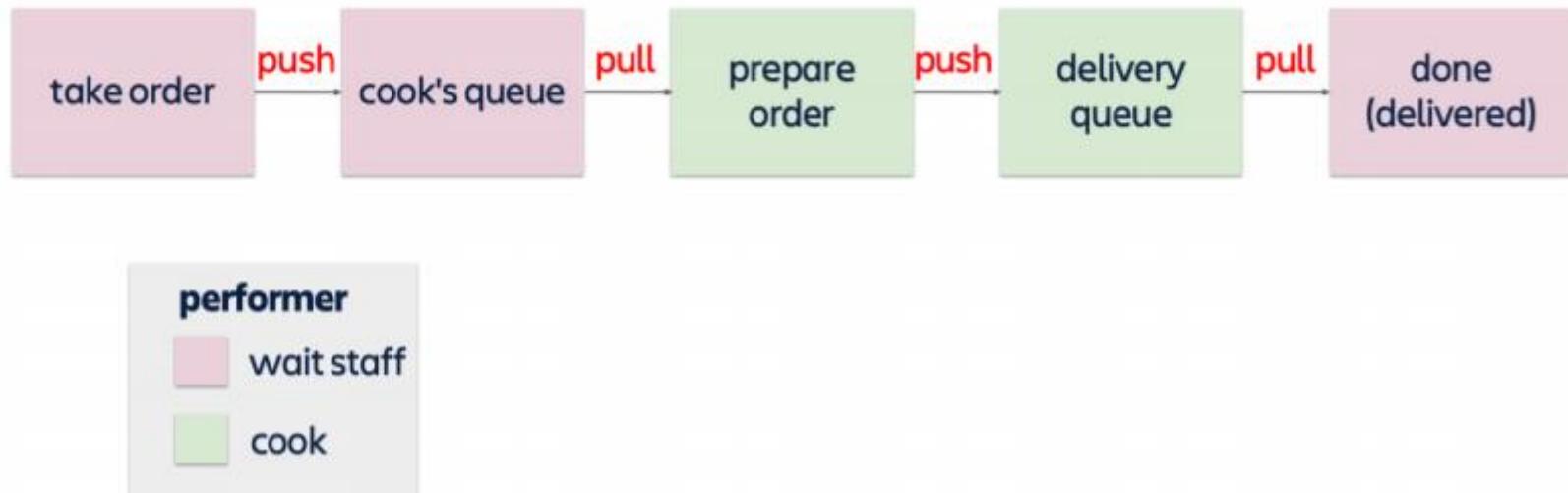
Topics

- Kanban method overview
- Improving flow
- Pull VS Push
- Separate Backlog



Pull vs Push in process steps

Performers either push work to the next step or pull from the previous step



Pull vs Push

Projects / projectA / PROJ board

Kanban board

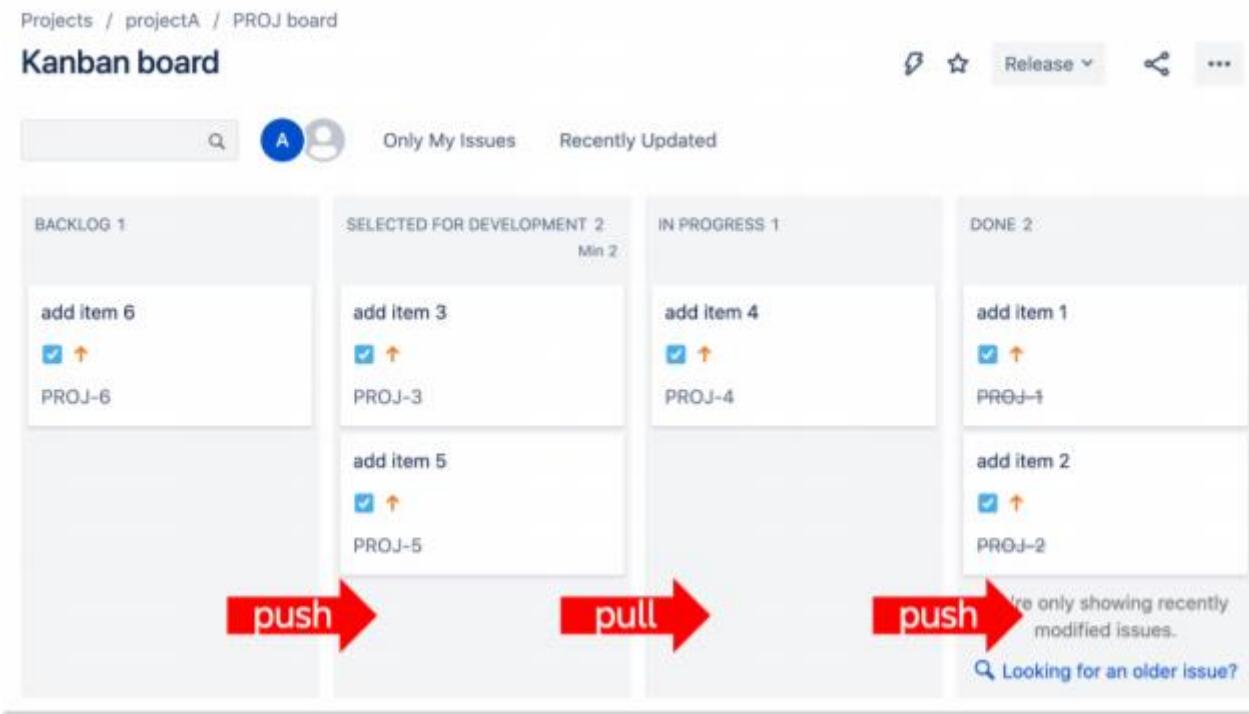
Release

Only My Issues Recently Updated

| BACKLOG 1 | SELECTED FOR DEVELOPMENT 2 | IN PROGRESS 1 | DONE 2 |
|--------------------------|----------------------------|--------------------------|--------------------------|
| add item 6 PROJ-6 | add item 3 PROJ-3 | add item 4 PROJ-4 | add item 1 PROJ-1 |
| | add item 5 PROJ-5 | | add item 2 PROJ-2 |

push **pull** **push**

We're only showing recently modified issues.
Looking for an older issue?



Adding queues to enabled pull (1 of 2)

Projects / projectA / PROJ board

Kanban board

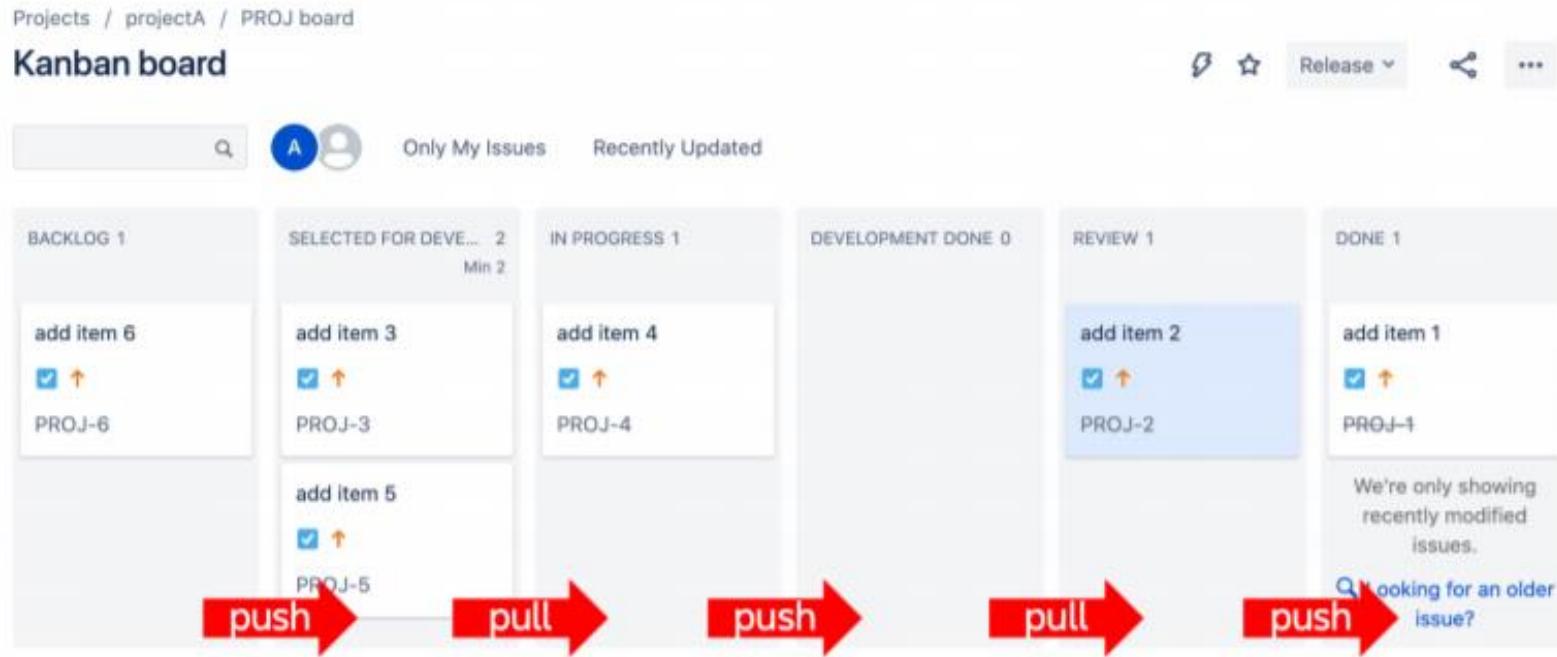
Release    

Only My Issues Recently Updated

| BACKLOG 1 | SELECTED FOR DEVEL... 2 | IN PROGRESS 1 | REVIEW 1 | DONE 1 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| add item 6   PROJ-6 | add item 3   PROJ-3 | add item 4   PROJ-4 | add item 2   PROJ-2 | add item 1   PROJ-1 |
| add item 5  PROJ-5 | | | | We're only showing recently modified issues.  Looking for an older issue? |

Adding queues to enabled pull (2 of 2)



Why Pull ?

- Empowers the team - team members select work, they are not assigned work
- Maintains a sustainable pace

Projects / projectA / PROJ board

Kanban board

Only My issues Recently Updated

| BACKLOG 1 | SELECTED FOR DEVELOPMENT 2 | IN PROGRESS 1 | DONE 3 |
|----------------------|----------------------------|----------------------|----------------------|
| add item 6 PROJ-6 | add item 3 PROJ-3 | add item 4 PROJ-4 | add item 1 PROJ-1 |
| | add item 5 PROJ-5 | | add item 2 PROJ-2 |

pull

We're only showing recently modified issues.

Looking for an older issue?

Topics

- Kanban method overview
- Improving flow
- Pull VS Push
- Separate Backlog



Separate Backflow

Projects / projectA / PROJ board

Kanban board

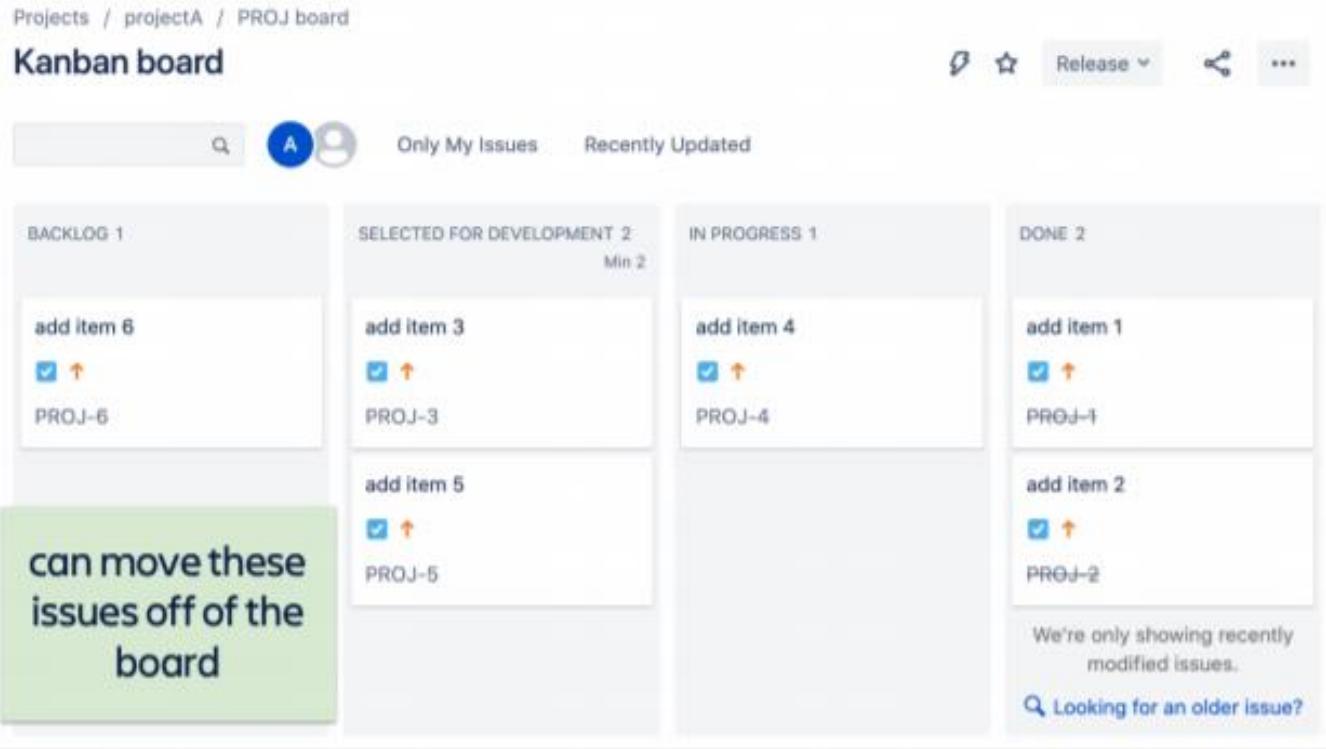
Only My Issues Recently Updated

| BACKLOG 1 | SELECTED FOR DEVELOPMENT 2 | IN PROGRESS 1 | DONE 2 |
|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| add item 6 <input checked="" type="checkbox"/> <input type="button" value="↑"/> PROJ-6 | add item 3 <input checked="" type="checkbox"/> <input type="button" value="↑"/> PROJ-3 | add item 4 <input checked="" type="checkbox"/> <input type="button" value="↑"/> PROJ-4 | add item 1 <input checked="" type="checkbox"/> <input type="button" value="↑"/> PROJ-1 |
| | add item 5 <input checked="" type="checkbox"/> <input type="button" value="↑"/> PROJ-5 | | add item 2 <input checked="" type="checkbox"/> <input type="button" value="↑"/> PROJ-2 |

can move these issues off of the board

We're only showing recently modified issues.

Looking for an older issue?



Why a Separate Backflow?

- Simplicity - separates the planning of issues from the project board
- The team can focus on work items that are ready to be worked on

Projects / projectA / PROJ board

Kanban board

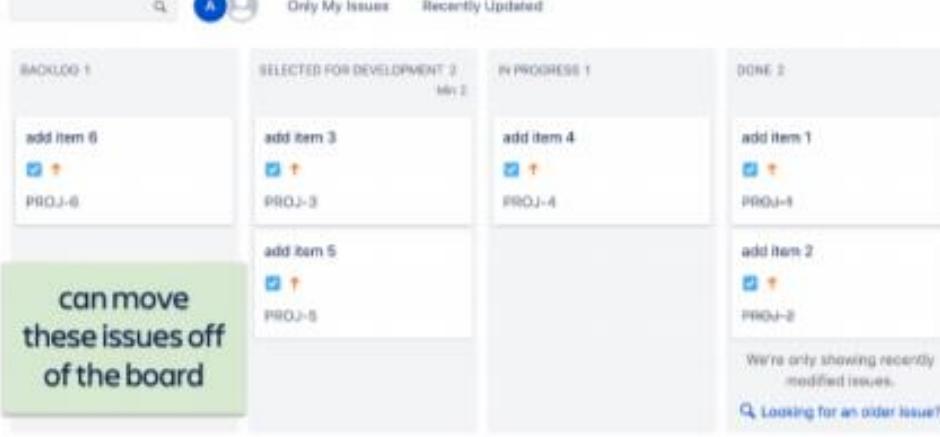
Only My Issues Recently Updated

| BACKLOG 1 | SELECTED FOR DEVELOPMENT 2 | IN PROGRESS 1 | DONE 2 |
|----------------------|----------------------------|----------------------|----------------------|
| add item 6 PROJ-6 | add item 3 PROJ-3 | add item 4 PROJ-4 | add item 1 PROJ-1 |
| | add item 5 PROJ-5 | | add item 2 PROJ-2 |

can move these issues off the board

We're only showing recently modified issues.

Looking for an older issue?



Managing a Separate Backflow

The screenshot shows a Jira project interface for 'projectA' under 'PROJ board'. A red arrow points to the 'Backlog' option in the left sidebar, which is currently selected. The main view displays two sections of the backlog:

- Selected for Development**: Contains 2 issues:
 - add item 3 (PROJ-3)
 - add item 5 (PROJ-5)
- Backlog**: Contains 1 issue:
 - add item 6 (PROJ-6)

At the bottom of the backlog section, there is a link to '+ Create issue'.

Takeaways

- Kanban Is a lightweight agile method
- A board should have a continuous flow of Issues moving from backlog to done columns
- Work in progress Limits can improve the flow of value by focusing team
- In Jira, the backlog can be separated from the board, simplifying the board and allowing separate backlog work

Lab 5- Kanban Method

- Configure WIP Limits
- View a cumulative flow diagram
- Configure a separate backlog

Scrum Overview Artifacts



What will you learn?

- Define scrum
- Describe an increment
- Identify scrum artifacts
- Define velocity

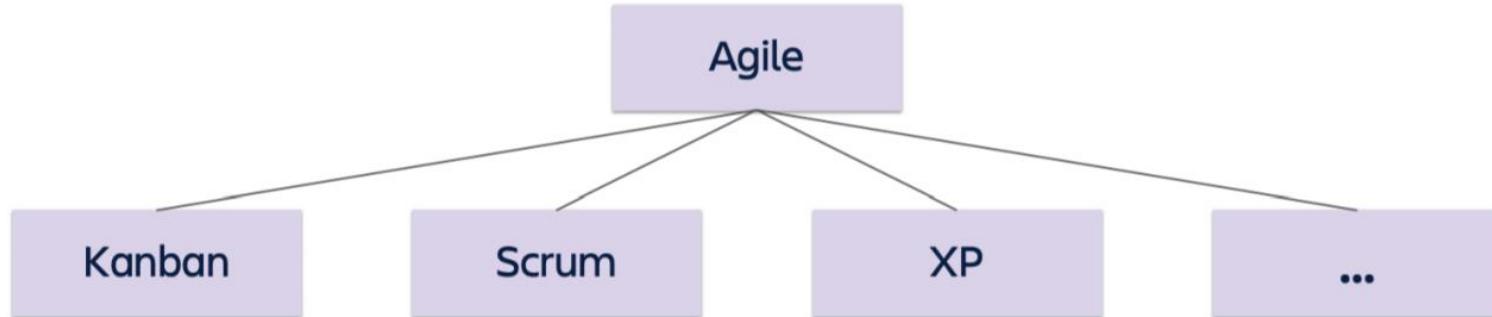
Topics

- What is scrum?
- Scrum Artifact

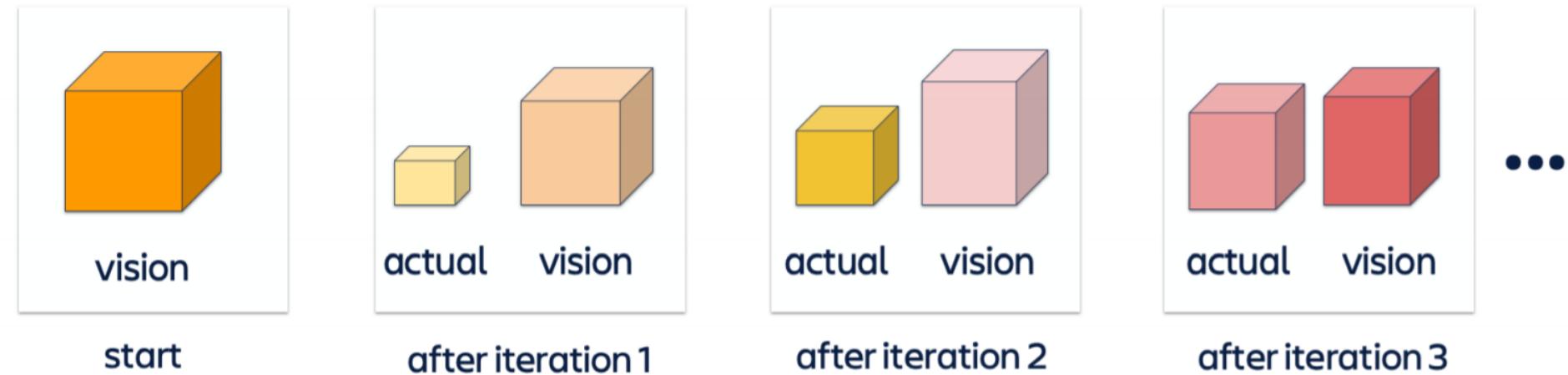


What is scrum?

- "Scrum is a framework for developing, delivering, and sustaining complex products."
Scrum Guide 2017
 - <https://www.scrum.org/resources/scrum-guide>
- A way of achieving agility



Continuous learning



Increment

- A usable product that may be given to the customer
- Meets the organization's "definition of done"
- Contains the work of the current iteration, as well as all prior iterations



Sprint

A time-boxed period used to work on an increment of the product

- Usually 1-4 weeks (typically 2 weeks)



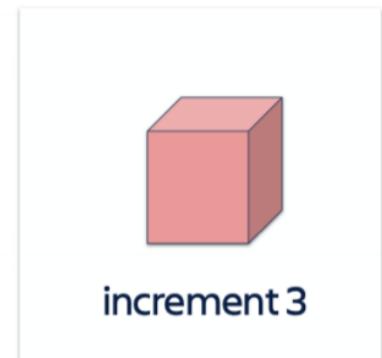
increment 1

after sprint 1



increment 2

after sprint 2



increment 3

after sprint 3

• • •

Parts of the scrum framework

- **Artifacts**- product backlog, sprint backlog, sprint goal, sprint board, reports
- **Roles** - product owner, scrum master, development team members, stakeholders
- **Events/Meetings/Ceremonies** - sprint, sprint planning meeting, daily standups, sprint review, sprint retrospective

Topics

- What is scrum?
- Scrum Artifacts



Scrum Artifacts

- Artifacts:
 - Product backlog
 - Sprint backlog
 - Sprint goal
 - Sprint board
 - Reports
- Provide project transparency
- Enable shared understanding
- Enable inspection and adaptation

Projects / projectB / PRJ board

Backlog

The screenshot shows a digital backlog interface. At the top, there is a search bar and a user icon labeled 'A'. Below the header, the word 'Backlog' is displayed next to '3 issues'. On the left, there are navigation tabs for 'VERSIONS' and 'EPICS'. The main area lists three backlog items, each with a green plus icon and the text 'add item 1', 'add item 2', and 'add item 3'. The third item, 'add item 3', is highlighted with a blue background. At the bottom right, there is a button labeled '+ Create issue'.

| Backlog | | 3 issues |
|----------|--|------------|
| VERSIONS | | add item 1 |
| EPICS | | add item 2 |
| | | add item 3 |

+ Create issue

Product Backlog

- An ordered, ever-changing to do list for the project
- Can include features, improvements, bug fixes, etc.
- Issues near the top should include more detail
- Modifying the product backlog is called product backlog refinement

Projects / projectB / PRJ board

Backlog

Share ...

Only My Issues Recently Updated

SEARCH A 8

Backlog 3 issues Create sprint ...

| VERSIONS | EPICS | Issues | Sprints |
|----------|-------|------------|-----------|
| | | add item 1 | PRJ-1 ↑ - |
| | | add item 2 | PRJ-2 ↑ - |
| | | add item 3 | PRJ-3 ↑ - |

+ Create issue

Creating a sprint with Jira

Projects / projectB / PRJ board

Backlog

Share ...

Only My Issues Recently Updated

A

PRJ Sprint 1 0 issues

Start sprint Linked pages 0 ...

VERSIONS

EPICS

Plan your sprint
As a team, agree on what work needs to be completed, and drag these issues to the sprint.

+ Create issue

0 issues Estimate 0

Backlog 3 issues

Create sprint ...

PRJ-1 ↑ -

PRJ-2 ↑ -

PRJ-3 ↑ -

add item 1

add item 2

add item 3

+ Create issue

Sprint Backlog

- A subset of the product backlog
- The list of issues to be completed in the sprint
- Includes the plan on how to accomplish the work of the issues
- Can fully emerge during the sprint

Projects / projectB / PRJ board

Backlog

Share ...

Only My Issues Recently Updated

PRJ Sprint 1 2 issues

Start sprint Linked pages 0 ...

VERSIONS EPICS

sprint backlog

| Issue | Epics |
|------------|-----------|
| add item 1 | PRJ-1 ↑ 1 |
| add item 2 | PRJ-2 ↑ 2 |

+ Create issue

2 issues Estimate 3

PRJ Sprint 2 1 issue

Create sprint ...

| Issue | Epics |
|------------|-----------|
| add item 3 | PRJ-3 ↑ 3 |

+ Create issue

Estimation-story point

- Story points are a relative measure of the amount of work (effort) required to complete the story
- Used to help decide how many stories can be completed in the sprint

Projects / projectB / PRJ board

Backlog

Share ...

A Only My Issues Recently Updated

Versions

PRJ Sprint 1 2 issues

Start sprint Linked pages 0 ...

EPICS

add item 1 PRJ-1 ↑ 1

add item 2 PRJ-2 ↑ 2

+ Create issue

2 issues Estimate 3

Backlog 1 issue

Create sprint ...

+ Create issue

PRJ-3 ↑ 3

add item 3

Description Add a description...

Story Points

1

The screenshot shows a Jira backlog board for a project named 'projectB'. At the top, there are sections for 'Start sprint' and 'Linked pages'. Below this, the backlog is divided into two main sections: 'PRJ Sprint 1' and 'Backlog'. 'PRJ Sprint 1' contains two items: 'add item 1' (Story Point 1) and 'add item 2' (Story Point 2). The 'Backlog' section contains one item: 'add item 3' (Story Point 3). On the right side of the board, there is a sidebar titled 'Story Points' which displays the value '1'.

Sprint Details

Start sprint

2 issues will be included in this sprint.

Sprint name: *
PRJ Sprint 1

Duration: *
2 weeks

Start date: *
01/Jul/20 4:38 PM 

End date: *
15/Jul/20 04:38 PM 

Sprint goal:

Sprint Goal

- Represents the objective of the sprint's increment
- Is reached by completing the sprint backlog
- Does not change during the sprint
- The sprint is a success if the sprint goal is reached

Start sprint

2 issues will be included in this sprint.

Sprint name: * PRJ Sprint 1

Duration: * 2 weeks

Start date: * 01/Jul/20 4:38 PM 

End date: * 15/Jul/20 04:38 PM 

Sprint goal:

Create the first product increment.

Why have a sprint goal

1. Provides coherence to the product increment
2. Enables flexibility with the sprint backlog

Start sprint

2 issues will be included in this sprint.

Sprint name: *
PRJ Sprint 1

Duration: *
2 weeks

Start date: *
01/Jul/20 4:38 PM 

End date: *
15/Jul/20 04:38 PM 

Sprint goal:
 Create the first product increment.

Sprint board

Only contains issues from the sprint backlog

The screenshot shows a Jira Sprint Board interface for a project named "projectB". The board is titled "PRJ Sprint 1" and has a due date of "9 days remaining". The board is divided into three columns: "TO DO", "IN PROGRESS", and "DONE". In the "TO DO" column, there are two items: "add item 1" and "add item 2". Both items have a green "New" icon, an orange "Up" arrow icon, and a circular badge with the number "1" for "add item 1" and "2" for "add item 2". To the right of each badge is the label "PRJ-1" for the first item and "PRJ-2" for the second. The "IN PROGRESS" and "DONE" columns are currently empty.

Projects / projectB / PRJ board

PRJ Sprint 1

Create the first product increment.

Only My Issues Recently Updated

TO DO IN PROGRESS DONE

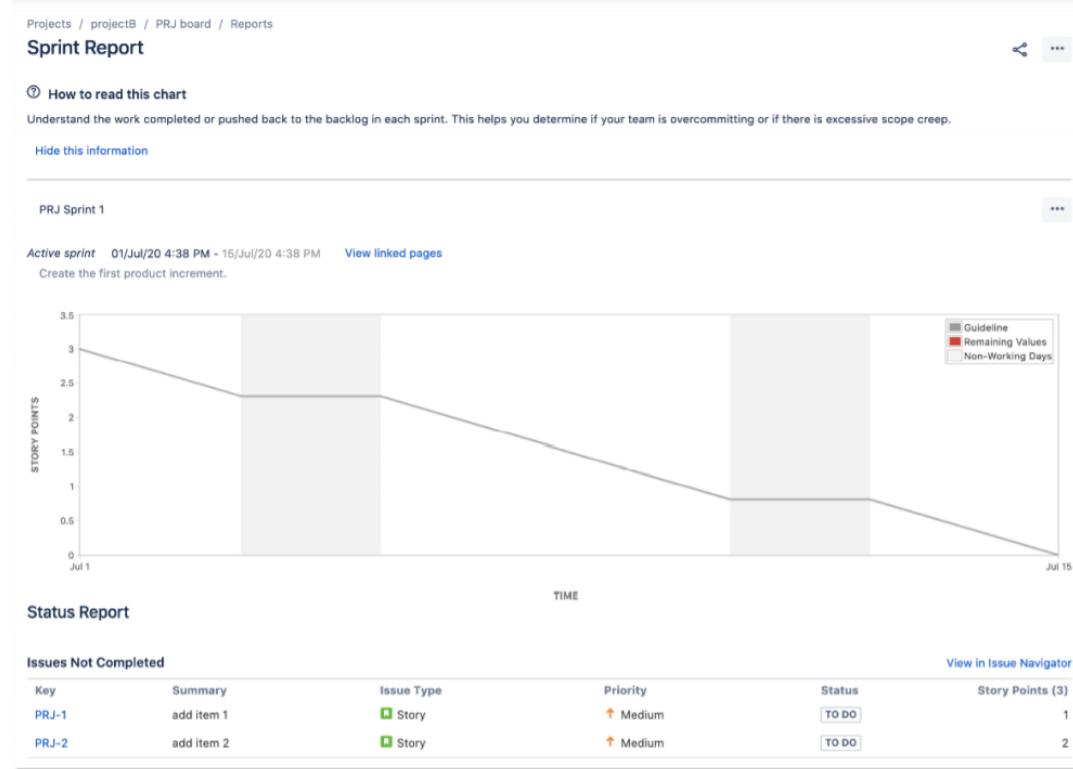
add item 1
PRJ-1

add item 2
PRJ-2

Scrum Reports – Burndown chart



Scrum Reports – Sprint Report



Velocity

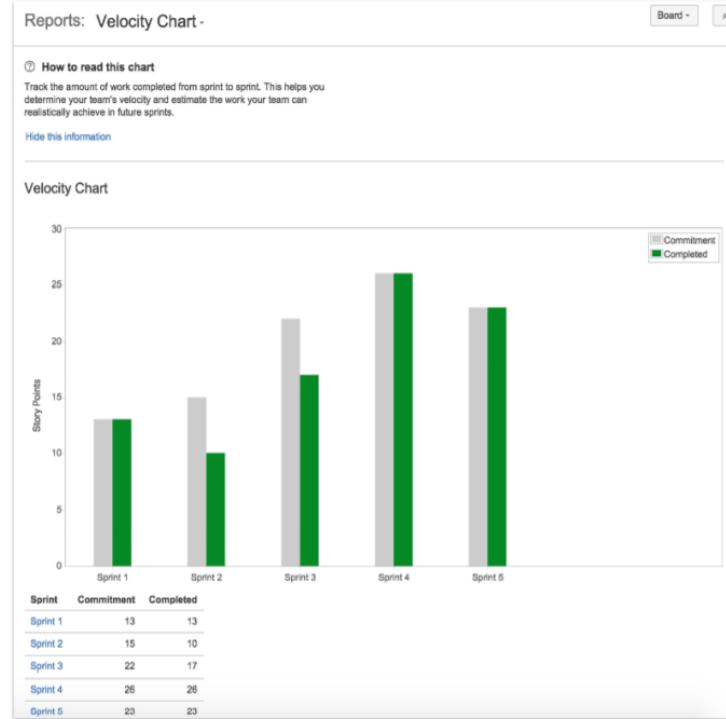
Represents the rate at which the team accomplishes work

- Usually it is the number of story points completed per sprint
- In this example, the velocity is 3 story points per sprint



Scrum Reports – Velocity Charts

Shows the estimated and actual velocity of the team over time



Takeaways

- Scrum is an agile framework
- An increment is a potentially shippable portion of the project that meets the "definition of done"
- A sprint is a time-boxed period in which an increment is created
- Scrum artifacts provide project transparency, enable shared understanding, and enable inspection and adaptation
- Artifacts include the product backlog, the sprint backlog, the sprint goal, sprint boards and reports
- Velocity is the rate at which the team accomplishes work, usually in story points per sprint

Scrum Overview Roles and Ev



What you will learn

- Describe scrum roles
- Differentiate the product owner and scrum master
- Identify common characteristics of scrum events
- Identify the purpose of the sprint planning meeting, daily standup, sprint review and sprint retrospective

Topics

- Scrum roles
- Scrum Events



Scrum Team

- Made up three roles: product owner, scrum master, development team
- Cross-functional
- Flexible/adaptable
- Self-organizing



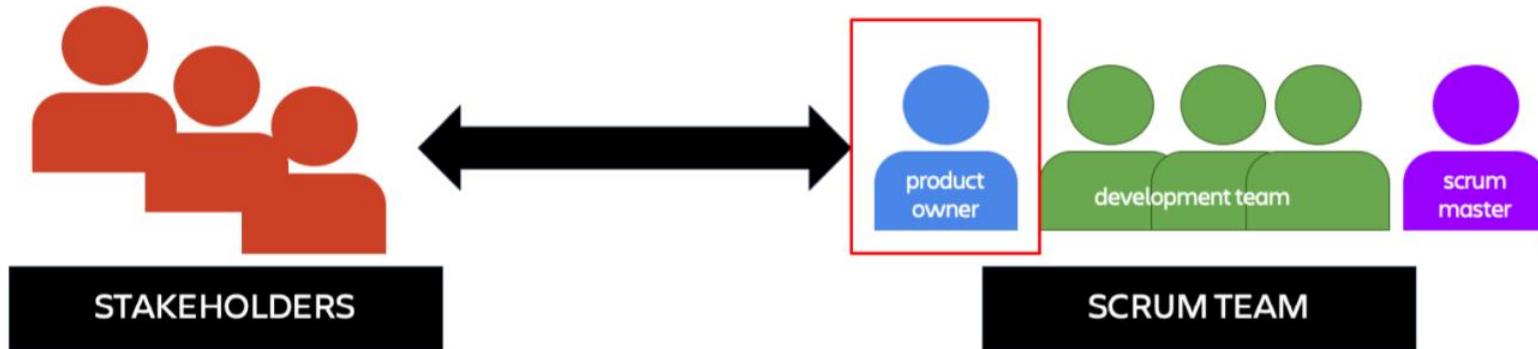
Stakeholders

- Others interested in the success of the project
- Internal- company managers, executives, other scrum teams
- External- customers, partners, investors



Product Owner

- Responsible for:
 - communicating the product vision
 - maximizing the value of each increment
 - the product backlog
- Interacts with, represents and is accountable to stakeholders



Scrum Master

- Responsibilities include:
 - promoting and supporting scrum
 - improving the day-to-day effectiveness of the team
 - protecting the focus of the team
 - increasing the transparency of the project
- Typical tasks:
 - coaching the scrum team and stakeholders on scrum
 - removing blocking issues
 - facilitating scrum events
 - configuring scrum artifacts
 - monitoring sprint progress



Product Owner VS Scrum Master



Why separate roles?

- Divide and concur
- Checks and balances

Development team

- Cross-functional, adaptive team that does the work of the project
- Responsibilities include:
 - estimating issues
 - deciding how much work can be done in a sprint
 - deciding how to organize to do the work of the sprint
 - creating the increment of each sprint
 - ability to modify the sprint backlog during the sprint
- The Scrum Guide recommends three to nine members



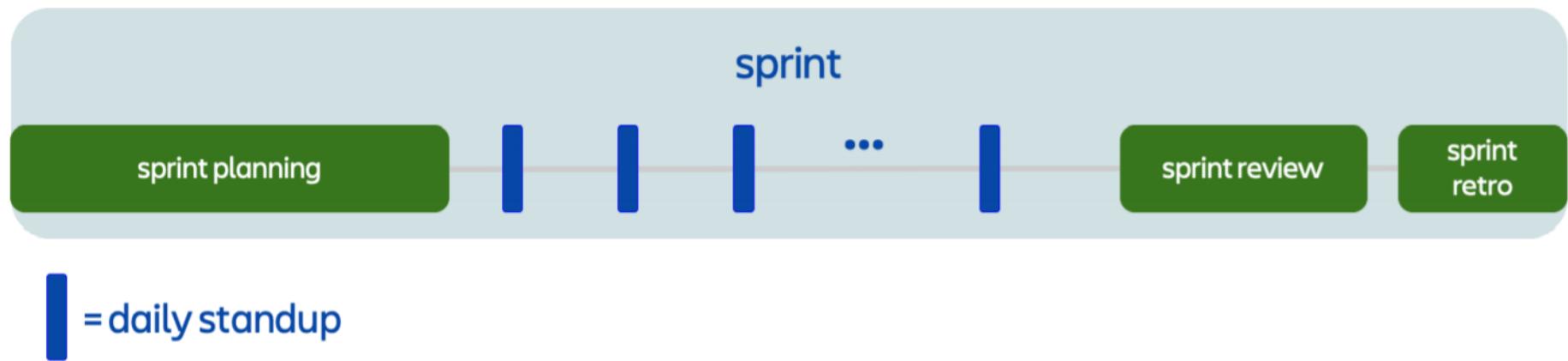
DEVELOPMENT TEAM

Topics

- Scrum roles
- Scrum Events

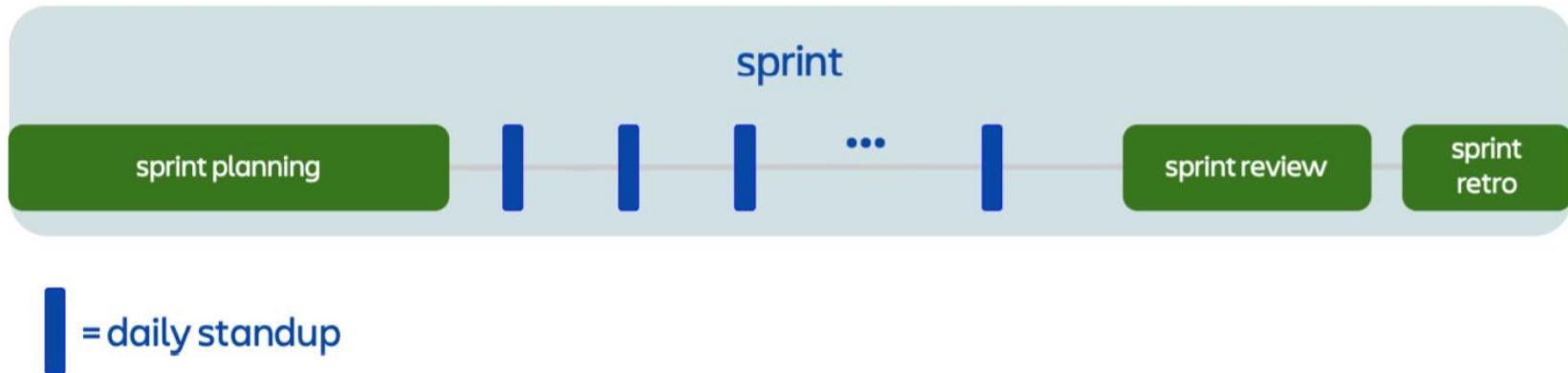


Scrum Events



Characteristics of all scrum meetings

- Fixed maximum time limit, no minimum time limit
- Meetings are primarily to plan, inspect and adapt
- Primarily about collaborating, not updating status
- Primarily spend time on things of value to all participants



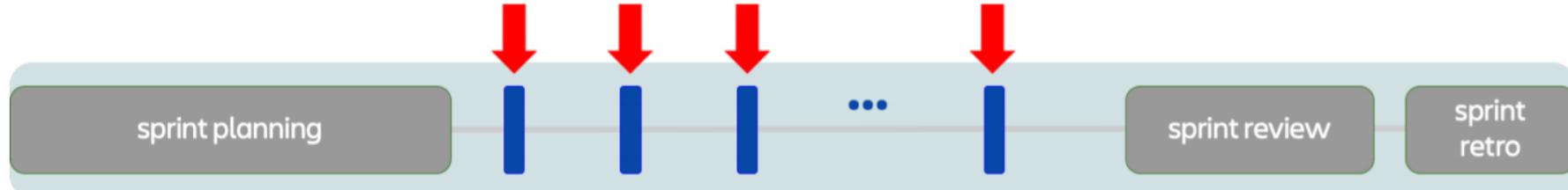
Sprint planning meeting

- **Attendees:** entire scrum team
- **Duration:** typically four hours for a two week sprint
- **Purpose:** plan the work of the sprint
- **Output:** sprint goal, sprint backlog



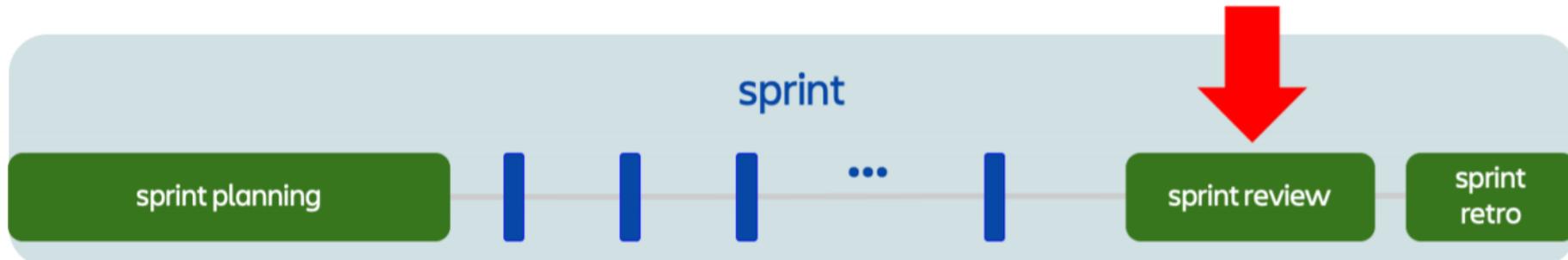
Daily Standup

- **Attendees:** development team and scrum master (primarily)
- **Duration:** 15 minutes
- **Purpose:**
 - Inspect recent progress toward the sprint goal
 - Plan the day's work
 - Identify any impediments, and plans to resolve them
- **Output:** plan for the day



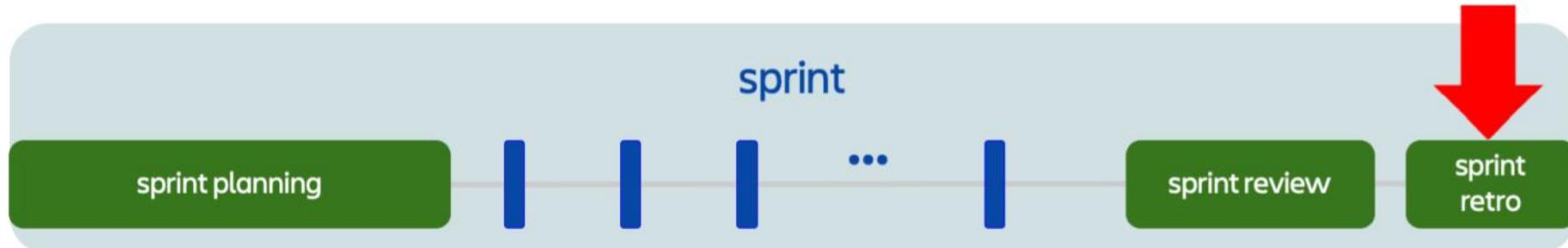
Sprint Review

- **Attendees:** scrum team and stakeholders
- **Duration:** typically 2 hours for a 2 week sprint
- **Purpose:** Inspect the increment and collaboratively update the product backlog
- **Output:** first-pass next sprint backlog



Sprint Retrospective

- **Attendees:** scrum team
- **Duration:** typically 90 minutes for a 2 week sprint
- **Purpose:** the team inspects itself, including its processes, tools and team interaction
- **Output:** Improvement issue(s) added to the next sprint's backlog

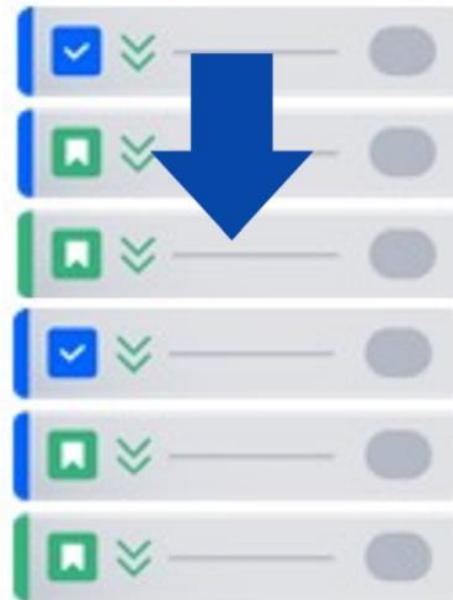


Scrum meetings - summary

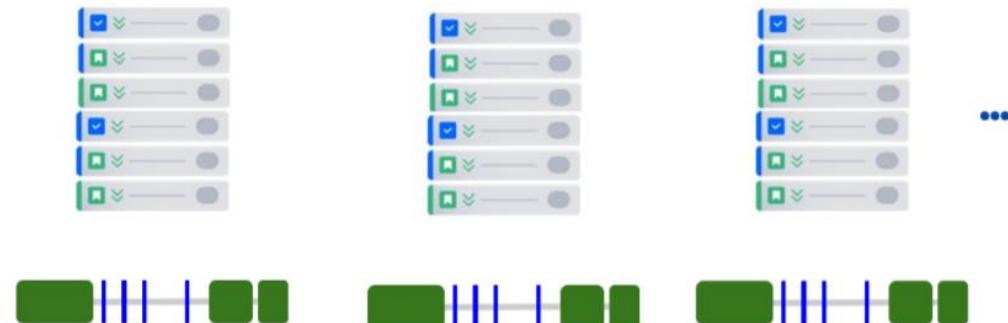
| | Sprint Planning | Daily Standup | Sprint Review | Retrospective |
|------------------|--------------------------------|---------------------------------|-------------------------------------------|-----------------------------|
| Attendees | Scrum team | Development team (primarily) | Scrum team and stakeholders | Scrum team |
| Duration* | 4 hours | 15 minutes | 2 hours | 90 minutes |
| Purpose | Plan the work of the sprint | Inspect recent work, plan today | Inspect increment, brainstorm next sprint | Inspect team |
| Output | Sprint goal, Sprint backlog | Today's plan | Proposed next sprint backlog | Amended next sprint backlog |

* typical max duration assuming a two week sprint

Kanban vs Scrum



Kanban



Scrum

Takeaways

Scrum roles include:

- product owner
- scrum master
- development team members
- stakeholders

Scrum meetings include:

- sprint planning meeting
- daily standups
- sprint review
- sprint retrospective

Lab 8 - Scrum Overview II

- Create a scrum project
- Create issues in the product backlog
- Create and plan a sprint
- Execute a sprint
- Complete a sprint

What is Jira Admin



WHAT IS JIRA ADMIN?

JIRA Admin is JIRA's control panel which allows you access to customize the tool specific to your organization needs. The JIRA Admin has access to and can customize

Workflow

Configurations

Schemes

Users

Screen

Roles & Permissions

Setting



JIRA

Jira Admin Architecture



Which Agile Methodology is Used in Jira?

JIRA Software is an agile project management tool that supports any agile methodology, be it Scrum, Kanban or your own Unique flavor





Cloud

Fast Start-up

Reduced Cost

No Upgrades

Security

No Access to Changes

Database Structure



Server

Install, Host & Run Yourself

Administrator

Manage Org Specific

Access to Changes

Database Structure

Groups, Roles and Users



Users

Invite Users: Useful for letting new users set up their own login credentials. Necessary when you want to speed up the onboarding process but ensure each user sets up their own secure access.

Groups

Understanding Groups:

Groups are used to organize users into different roles or designations, streamlining permission management. This is especially useful in larger projects or organizations where different teams need different access levels.

Anonymity

- User anonymization in Jira allows you to hide or delete any data that can identify a user as a real person.
- This is crucial for protecting user privacy and adhering to organizational policies or legal requirements.

Integrate User Accounts

Many organizations use multiple Atlassian products like Jira Core, Confluence, and Jira Software. Integrating these platforms can streamline user management across different teams and projects.

Advanced User Management

1. External directories allow organizations to manage user accounts outside of Jira. This is useful for organizations already using systems like Microsoft Active Directory, LDAP, or Atlassian Crowd.
2. Integrating an external directory prevents the need to duplicate user account setups within Jira, simplifying user management across different platforms.

Projects



Overview of Project Management

- In Jira, the project management section allows you to manage all your existing projects.
- This includes viewing details of projects, editing them, or even deleting them (though deletion is generally not recommended due to potential data loss).

Project Categories

- Project categories in Jira allow you to classify projects into groups such as development, support, or maintenance.
- This classification helps in better organization and management of projects according to their nature and operational focus.

Schemes and Application Management



Licenses, Versions, Upgrades

- To dive into applications management, access the Jira Server administrator menu by clicking on the cogwheel icon, then navigate to the "Applications" section.
- This area centralizes management of all installed products on your Jira Server instance.

Plan your Upgrade

- Following the versions and licenses segment, the upgrade planning tool allows administrators to check for new updates or versions released by Atlassian.
- This tool is invaluable for keeping your Jira installation current and secure.

Application Access

- Beyond upgrades, another critical area is application access control.
- This feature allows administrators to define and restrict which users or groups within the organization have access to specific applications installed on the server.

Custom Fields



Custom Issue Types

- Custom issue types are essential for projects that require unique categorizations beyond the default settings provided by Jira.
- They allow teams to track and manage tasks, bugs, requests, and more with specific attributes and workflows that align with project demands.

Create New Issue Types

- To begin customization, navigate to the Jira Administrator dashboard. Click on the cogwheel icon to access the administration menu, then select 'Issues' to enter the issue types configuration area.

Associate Issue Types with Projects

- Issue types are grouped into schemes that can be applied to one or more projects.
- This allows administrators to apply consistent issue type settings across multiple projects or customize them for individual projects.



JumpStart to Jira for Administrators

Table of Content

1. Customizing Jira for Your Projects: 3
2. Jira Workflows: 57
3. User Management: 106
4. Jira Security: 162



Customizing Jira for Your Projects

Setting up different issue types for projects

- Jira comes with a number of issue types out of the box that are designed for software project management.
- However, over time, you might find that these issue types do not apply to all of your projects, and you have added your own.

How to do it...

Modify Issue Type Scheme — HUM: Scrum Issue Type Scheme

USED BY 1 PROJECT

+ Add issue type

Scheme Name * HUM: Scrum Issue Type Scheme

Description

Default Issue Type Story

Change the order of the options by **dragging and dropping** the option into the desired order. Similarly, **drag and drop** the option from one list to the other to add or remove them.

| Issue Types for Current Scheme | Available Issue Types |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <p>Remove all</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Task<input type="checkbox"/> Sub-task (sub-task)<input type="checkbox"/> Story<input type="checkbox"/> Bug<input type="checkbox"/> Epic | <p>Add all</p> <ul style="list-style-type: none"><input type="checkbox"/> Story |

Save Reset Cancel

How to do it...

Having created your new issue type scheme, you now need to apply it to projects in which you want to restrict issue type selections:

1. Click on the Associate link for the new issue type scheme.
2. Select the project(s) you want to apply the scheme to.
3. Click on Associate to change the selected projects' issue type scheme.

Making a field required

- Required fields such as Summary and Issue Type have a little red asterisk next to them, which means they must have a value when you are creating or updating an issue.
- This is a great way to ensure that users do not skip filling in important information.

How to do it...

Create Issue

Configure Fields ▾

Project* Project Hummingbirds (HUM)

Issue Type* Bug

Summary* You must specify a summary of the issue.

Reporter* Patrick Li Start typing to get a list of possible matches.

Due Date* Due Date is required.

Component/s None

Description

Style **B** *I* U A ^D _E ~~O~~ U

Create another **Create** Cancel

How it works...

- When a field is marked as required, Jira will check to make sure that the field has a value when you are making updates to the issue, such as an edit, or during a workflow transition.
- This validation is applied even if the field is not present on the screen, so make sure you do not make a field that is not required on screen; otherwise, users will not be able to complete the action.

Making the Assignee field required

- By default, the Assignee field has an unassigned option, which is equivalent to making the field optional.
- If you look at the field configuration, you will realize that you cannot make the Assignee field required, as there is no such option available.

Getting ready

Note the following conditions are required for your projects in order to disable the unassigned option for each individual project:

- Issues that are currently using that option for the Assignee field—you will need to change all issues with an Unassigned value for Assignee to something else.
- Projects that have Unassigned set as the default Assignee—you will need to change your project's Default Assignee setting in the Users and roles section.

How to do it...

Create Issue

Configure Fields ▾

Project* Project Hummingbirds (HUM)

Issue Type* Bug

Summary*

Reporter* Patrick Li

Start typing to get a list of possible matches.

Assignee* Automatic

Suggestions

Patrick Li - patrick@appfusions.com (patrick)

Start Typing for Users

All Users

John Clark - john@appfusions.com (john)

Component/s

Description

Create another Create Cancel

Hiding a field from view

- There will be times when a field is no longer needed.
- When this happens, instead of deleting the field, which would also remove all its data, you can choose to hide it.
- If you need the field again further down the track, you can simply unhide it and retain all the data.

How to do it...

1. Log into Jira with a user that has a Jira administrator's permission.
2. Navigate to Administration > Issues > Field Configurations.
3. Click on the Configure link for the field configuration used by the project and issue type.
4. Click on the Hide link for Priority and Due Date.

Creating a new field configuration

- You can configure a field's behavior with field configuration.
- Jira not only comes with a default field configuration that is applied to all project and issue types by default.
- But it also lets you create your own so that you can choose the projects and/or issue types to apply your field configuration to.

How to do it...

1. Log into Jira with a user that has a Jira administrator's permission.
2. Navigate to Administration > Issues > Field Configurations.
3. Click on the Add Field Configuration button, and name it HUM: Bug Field Configuration, and click on Add.
4. Click on the Required link for the Description and Assignee fields.

How to do it...

Associate an Issue Type with a Field Configuration

Issue Type

Bug



Field Configuration

HUM: Bug Field Configuration



Add

Cancel

How to do it...

1. Navigate to Administration > Projects.
2. Select a project from the list.
3. Select Fields from the left-hand panel.
4. Navigate to Actions > Use a different scheme.
5. Select the new HUM: Bug Field Configuration Scheme option and click on Associate.

Setting up customized screens for your project

- Jira comes with three screens by default—the Default screen, the Resolve Issue screen, and the Workflow screen.
- We will look at how to create a new screen from scratch, and then make it appear when we are creating a new Task issue.

How to do it...

Configure Screen

This page shows the way the fields are organized on **HUM: Scrum Task Create Screen** screen.

Note: when the screen is shown to the user only non-hidden fields that the user has permissions to edit will be actually displayed.

III **Field Tab**  

| Field | Type |
|-------------|--------------|
| Summary | System field |
| Issue Type | System field |
| Assignee | System field |
| Description | System field |
| Priority | System field |

Field name  

| | |
|-------------------|--------------|
| Affects Version/s | System field |
| Attachment | System field |
| Component/s | System field |

How to do it...

Associate an issue operation with a screen

Issue Operation

Create Issue



Screen

HUM: Scrum Task Create Screen



The screen to show for the chosen issue operation.

Add

Cancel

How to do it...

Associate an Issue Type with a Screen Scheme

Issue Type

Task



Screen Scheme

HUM: Scrum Task Screen Scheme



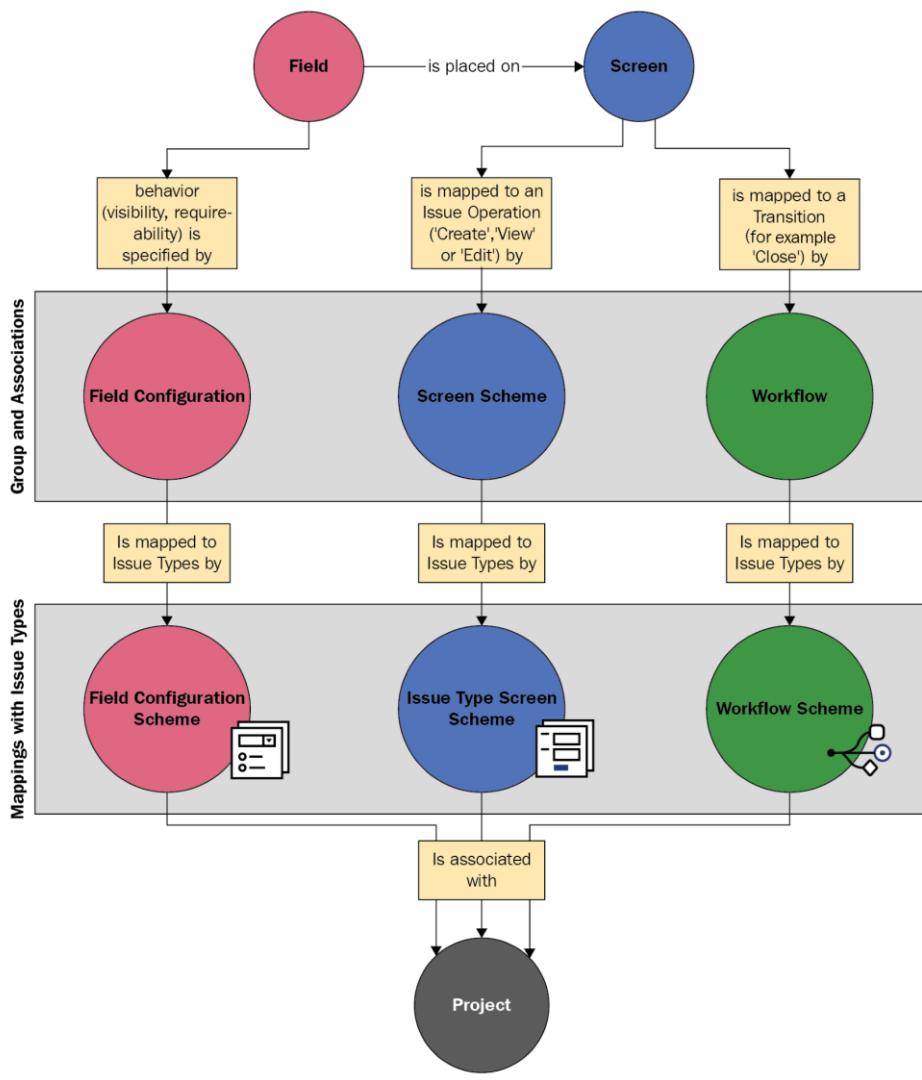
Add

Cancel

How to do it...

1. Navigate to Administration > Projects.
2. Select a project from the list.
3. Select Screens from the left-hand pane.
4. Navigate to Actions > Use a different scheme.
5. Select the new Task Issue Type Screen Scheme and click on Associate.

How it works...



Removing a select list's None option

- Custom field types such as select list (single and multi) come with the None option, and the only way to remove it is to make the field required.
- While this makes sense, it can be cumbersome to chase down every field and configuration.

How to do it...

1. Open the edit-select.vm file (located in the previously mentioned directory) in a text editor, and remove the following code snippet:

```
#if (!$fieldLayoutItem || $fieldLayoutItem.required == false)
<option value="-1">
    $i18n.getText("common.words.none")
</option>
#else
#if ( !$configs.default )
<option value="">
    $i18n.getText("common.words.none")
</option>
#end
#endif
```

2. Save the file and restart Jira. Make sure you do not change any other lines.

How it works...

- The Velocity .vm template files are what Jira uses to render the HTML for the custom fields.
- The code snippet we removed is what displays the None option.
- Note that, by changing the template, we are removing the None option for all single select custom fields in Jira.

Adding help tips to custom fields

- Users who are new to Jira often find it confusing when it comes to filling in fields, especially custom fields.
- Therefore, it is for you as the administrator to provide useful tips and descriptions to explain what some of the fields are for.

How to do it...

1. Log into Jira with a user that has a Jira administrator's permission.
2. Navigate to Administration > Issues > Custom Fields.
3. Click on the Edit link for the custom field.
4. Enter the following HTML snippet into the Description text box, and click on Update. You might want to substitute the href value to a real page containing help text:

Need help to work out assignment?

```
<a class="help-lnk" href="/secure>ShowConstantsHelp.jspa?decorator=popup#Teams"  
data-helplink="local" target="_blank">  
    <span class="aui-icon aui-icon-small aui-iconfont-help"></span>  
</a>
```

Create Issue

Configure Fields ▾

Project *  Project Hummingbirds (HUM) ▾

Issue Type *  Story ▾ ?

Summary *

Reporter *  Patrick Li

Start typing to get a list of possible matches.

Team  None

Need help to work out assignment? ? [help tip](#)

Description

Rich text editor toolbar:

- Left/Right arrows
- Bold (B)
- Underline (U)
- Italic (I)
- Font color (A)
- Font style (Pencil)
- Styles (dropdown)
- Format (dropdown)
- Image (Image icon)
- Table (Table icon)
- Link (Link icon)
- Source (Source icon)
- Smiley face (Smiley icon)
- Code (Code icon)
- Checklist (List icon)
- Horizontal line (Line icon)
- Page break (Page Break icon)
- Page number (Page Number icon)
- Page orientation (Orientation icon)
- Page size (Page Size icon)
- Page margins (Margins icon)
- Page header (Header icon)
- Page footer (Footer icon)
- Page background (Background icon)
- Page watermark (Watermark icon)
- Page fit (Fit icon)
- Page zoom (Zoom icon)
- Page orientation (Orientation icon)
- Page size (Page Size icon)
- Page margins (Margins icon)
- Page header (Header icon)
- Page footer (Footer icon)
- Page background (Background icon)
- Page watermark (Watermark icon)
- Page fit (Fit icon)
- Page zoom (Zoom icon)

How to do it...

How it works...

- Jira allows us to use any valid HTML for custom field description, so we added some simple text and an anchor tag that links to an HTML page containing our help information.
- We also added a span tag with the proper style class in order to have the nice question mark icon used by the Issue Type and Priority fields.

There's more...

Normally, we put descriptions directly into the custom field's description textbox, as demonstrated. You can also put your descriptions into the field configuration settings, such as hiding a field. Doing so offers the following advantages:

- You can have different help text for different project/issue type contexts.
- You can set help text for fields that are not custom fields, such as Summary and Description.

Using JavaScript with custom fields

- JavaScript can be used to manipulate the HTML of custom fields.
- By adding JavaScript code in the custom field description and wrapping the code in `<script>` tags, Jira will execute the code when the custom field is rendered.

Getting ready

- We will need to use the custom field's ID in our script, so you will need to have that handy.
- You can find the ID by going to the Custom fields page, clicking on the Edit link of the target field, and clicking the number at the end of the URL, which is the field's ID.
- For example, the following URL shows a custom field with the ID 10103:

<http://jira.localhost.com:8080/secure/admin/EditCustomField!default.jspa?id=10103>

How to do it...

1. Log into Jira with a user that has a Jira administrator's permission.
2. Navigate to Administration > Issues > Custom Fields.
3. Click on the Edit link for the custom field.
4. Enter the following JavaScript snippet into the Description text box, and click on Update. You will need to substitute it in your custom field's ID:

```
<script>  
AJS.$('#customfield_10103 option[value="- 1"]').remove();  
</script>
```

Create Issue

Project* Project Hummingbirds (HUM)

Issue Type* Story

Summary*

Reporter* Patrick Li

Start typing to get a list of possible matches.

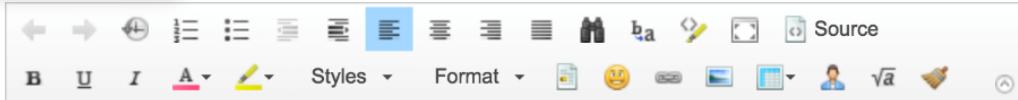
Team ✓ Ninja Team

Tiger Team

X-Force

out assignment? ?

Description



How it works...

- In our script, we use jQuery to select the Team custom field, based on its element ID, and remove the option with the value -1 (which is the None option) with the `#customfield_10103 option[value="-1"]` selector.
- We use the Atlassian JavaScript (AJS) namespace (AJS.\$), which is the recommended way to use jQuery in Jira.

Setting default values for fields

- As you add more and more fields onto your screens, sometimes it can be overwhelming and confusing for your end users, especially for free text fields such as Description.
- Users will often enter data however they like.
- To help with that, one way is to have instructions set as default values for some of these fields to help guide your users with their inputs.
- For those of you who have used other systems, such as GitHub, this is a common practice.

How to do it...

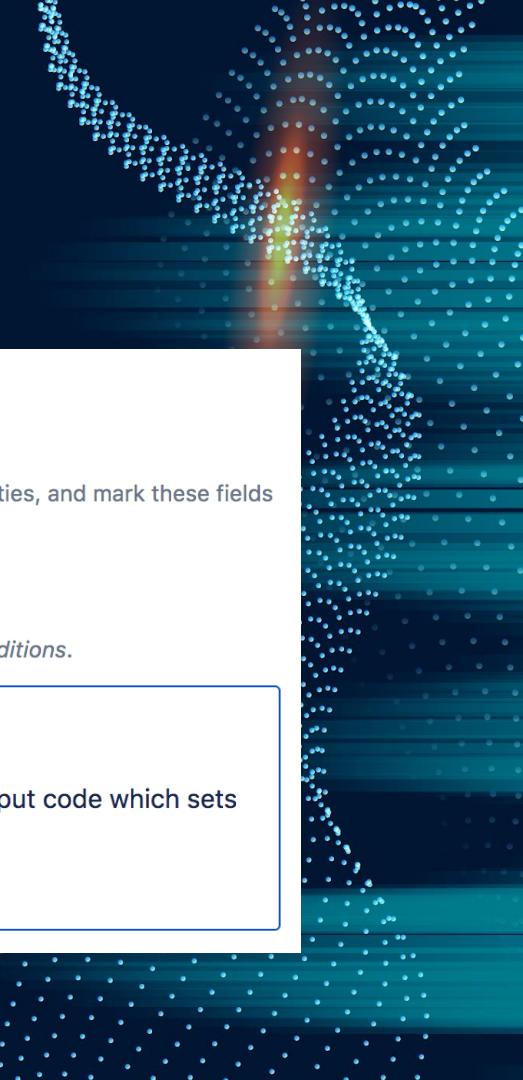
1. Log into Jira with a user that has a Jira administrator's permission.
2. Navigate to Administration > Manage apps > Behaviors.
3. Create a new behavior by entering a name for it and clicking the Add button.

How to do it...

1. Click on the Add Mapping link of the new behavior we have created.
2. Select All projects and Bug issue type for our mapping.
3. Click the Add Mapping button to save the setting.

How to do it...

1. Click on the Fields link of the new behavior we have created.
2. Click on the Create initialiser link:



Behaviour Settings

Use Validator Plugin

This will check your workflow for uses of the *Fields Required* validator in Jira Suite Utilities, and mark these fields as required.

Guide workflow

Use this setting to allow the tool to show you actions and state names when using *Conditions*.

Initialiser

i Initialiser Function

Initialisers run once when the form is loaded, and is a good place to put code which sets default values, or sets options for instance.

[Create initialiser.](#)

How to do it...

3. Enter the following code into the text editor:

```
def desc = getFieldById("description")
def defaultValue = ""
h2. Describe the problem
Tell us in details what the problem you are having.

h2. How to reproduce
List out the steps to reproduce the problem.
* step 1
* step 2

h2. Expected Result
Tell us what you think the correct outcome should be after completing the steps 13.      listed above.

h2. Actual Result
Tell us what you actually see after completing the steps listed above.""""

if (!underlyingIssue?.description) {
    desc.setFormValue(defaultValue)
}
```

4. Click on Save to save our changes.

How to do it...

Create Issue

Configure Fields ▾

Description

Style ▾ **B** *I* U ~~A~~ ^{A°} _{A°}

Describe the problem

Tell us in details what the problem you are having.

How to reproduce

List out the steps to reproduce the problem.

- step 1
- step 2

Expected Result

Tell us what you think the correct outcome should be after completing the steps listed above.

Actual Result

[Visual](#) [Text](#)

Priority Medium

Create another Create Cancel

TriveraTech
TECHNOLOGY TRAINING

How it works...

- Our code is written in Groovy, which is very similar to Java, the technology used by Jira.
- We first try to get hold of the Description field in line #1, and, from line #3 to #20, we use wiki markup, the markup language used by Jira's out-of-the-box editor, to create your default value in the form of instructions.
- On line #18, we check whether the issue already has a description value set, such as user-entered content, and, if not, we will apply our default value on line #19.

Adding permission to fields

- Out of the box, Jira comes with several levels of permissions, starting at the system level and going down to the issue level, allowing you to control who should have access to issues.
- While this is usually sufficient, you will find yourself needing to apply permissions to individual fields.
- For example, you may want a field such as Description to be read-only for everyone, but only editable by a select group of users.

How to do it...

1. Log into Jira with a user that has a Jira administrator's permission.
2. Navigate to **Administration** > **Manage apps** > **Behaviors**.
3. Create a new behavior by entering a name for it and clicking the **Add** button.

How to do it...

1. Click on the **Add Mapping** link of the new behavior we have created.
2. Select the projects and issue types you want to apply the behavior to. If you want this to be global, you can select the All projects and All issue types options.
3. Click the **Add Mapping** button to save the setting.

How to do it...

1. Click on the Fields link of the new behavior we have created.
2. Select the fields you want to apply field-level permissions to.
3. Toggle the available behavior options on and off, as shown here:

Fields

| Description | <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> | Optional <small>Click to make required</small> |
|---------------|---------------------------------------------------------------------------------------|------------------------------------------------|
| Delete | <input checked="" type="checkbox"/> | Readonly <small>Click to make writable</small> |
| | <input type="checkbox"/> | Shown <small>Click to make hidden</small> |

Conditions

- Except:
 -  Current user is current assignee
- [Add new condition...](#)

How to do it...

4. Click on the Add new condition link and select which users the permission should be applied to. In the following screenshot, the permission will be applied to everyone except users in the jira-administrators group:

Add condition

Add one or more conditions to the behaviour for field: **Description**.

When - behaviour will happen if condition is true
 Except - behaviour will **not** happen if condition is true

Current user is issue reporter
 Current user in group
 Current user is
 User in project role:
 Current user is project lead
 Current user is current assignee
 Current user is value of user field
 Current user member of group custom field value

jira-administrators

Select user(s)

Workflow Action: Enter a guide workflow before choosing actions or steps
 Workflow Step: Enter a guide workflow before choosing actions or steps

Add **Cancel**

How to do it...

Edit Issue : HUM-1

Configure Fields ▾

Assignee * Eric Lin

Assign to me

Description

Style **I** A ^{A^o} ~~D~~  About Scrum

Scrum is an iterative approach to Agile software development. The methodology has been around since the 1980s but was popularised by Jeff Sutherland and Ken Schwaber. Scrum breaks the development of a product down into discrete iterations (termed Sprints) that each deliver functionality that could potentially be shipped to users. The Scrum Alliance offers an excellent [introduction to Scrum](#) that provides an overview of key Scrum concepts, stakeholders, processes and artefacts.

Visual Text

Team None

Need help to work out assignment? [?](#)

Comment

Style **I** A ^{A^o} ~~D~~

Update Cancel

Creating your own custom field types

- All custom fields that come out-of-the-box with Jira have predefined purposes, such as the text field, which allows users to type in some simple text.
- It will often be useful to have a specialized custom field that does exactly what you need.
- Unfortunately, this often requires custom development efforts.
- However, there is an add-on that provides a custom field type that lets you use Groovy scripts to power its logic.

How to do it...

Select a Field Type

Search

All

Standard

Advanced



No field preview

Scripted Field
A calculated field whose value is calculated by running a groovy script. Search admin for "Script Fields" after adding to configure.

No field preview

Text Field (read only)
A read-only text label. Only possible to create values programmatically (Used internally for imports from Mantis). Maximum of 255 characters.

No field preview

User Picker (multiple users)
Choose multiple users from the user base via a popup picker window.

No field preview

Version Picker (multiple versions)
Choose from available versions in the project.

No field preview

Version Picker (single version)
Choose a single version from available versions in the project.

 Find more custom fields

Next Cancel

How to do it...

```
import com.atlassian.jira.component.ComponentAccessor  
  
import com.atlassian.jira.issue.comments.CommentManager  
  
  
  
  
def commentManager = ComponentAccessor.getCommentManager()  
  
def numberOfComments = commentManager.getComments(issue).size()  
  
  
  
return numberOfComments ? numberOfComments as Double : null
```

How to do it...

Custom Script Field

Create your own custom scripted field.

Field Name Total Comments

The name of the custom field that you are creating

Field Description

The description of the custom field that you are creating

Note

An optional note, used only for your reference.

Template Number Field

Template to represent your field. Match to the output of your script.

Script file Start typing to search for files...

Path to the script accessible on the server

Inline script
2 import com.atlassian.jira.issue.comments.CommentManager

3
4 def commentManager = ComponentAccessor.getCommentManager()
5 def numberOfComments = commentManager.getComments(issue).size()

6
7 return numberOfComments ? numberOfComments as Double : null

8

Enter your script here   

Preview Issue Key

Issue key for preview. Only used for the preview function below.

Preview

Update

Cancel

How to do it...



Project Hummingbirds / HUM-1

As an Agile team, I'd like to learn about Scrum >> Click the "HUM-1" left of this row to see detail in the Description tab on the right

Edit

Comment

Assign

More ▾

To Do

In Progress

Workflow ▾

Admin ▾

Details

Type:

Story

Status:

TO DO (View Workflow)

Priority:

Medium

Resolution:

Unresolved

Affects Version/s:

None

Fix Version/s:

Version 2.0

Labels:

None

Total Comments:

3

How it works...

- The scripted field type is an example of what is called the calculated custom field type.
- The calculated custom field type is a special custom field that derives (calculates) its value based on some predefined logic, in this case, our Groovy script.
- Every time the field is displayed, Jira will recalculate the field's value so it is always kept up to date.

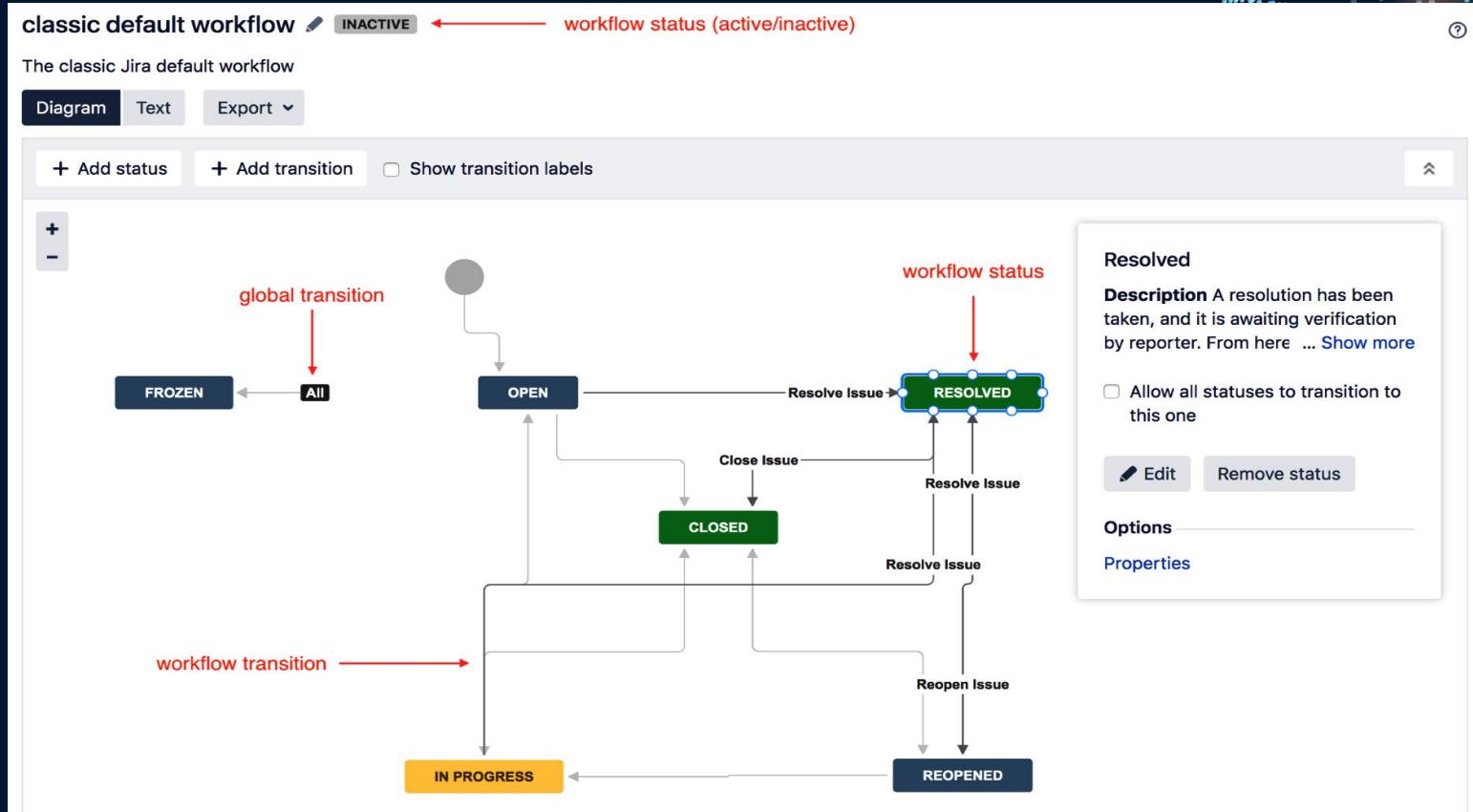


Jira Workflows

Setting up different workflows for your project

- A workflow is like a flowchart in which issues can go from one state to another by following the direction paths between the states.
- In Jira's workflow terminology, the states are called statuses, and the paths are called transitions.
- We will use these two major components when customizing a workflow.

How to do it...



How to do it...

1. Click on the Add status button.
2. Select the In Progress status from the list, and click on Add.
3. Repeat the steps to add the Closed status.

How to do it...

Workflows

Simple Workflow

INACTIVE

Diagram Text Export ▾

+ Add status + Add transition Show transition labels

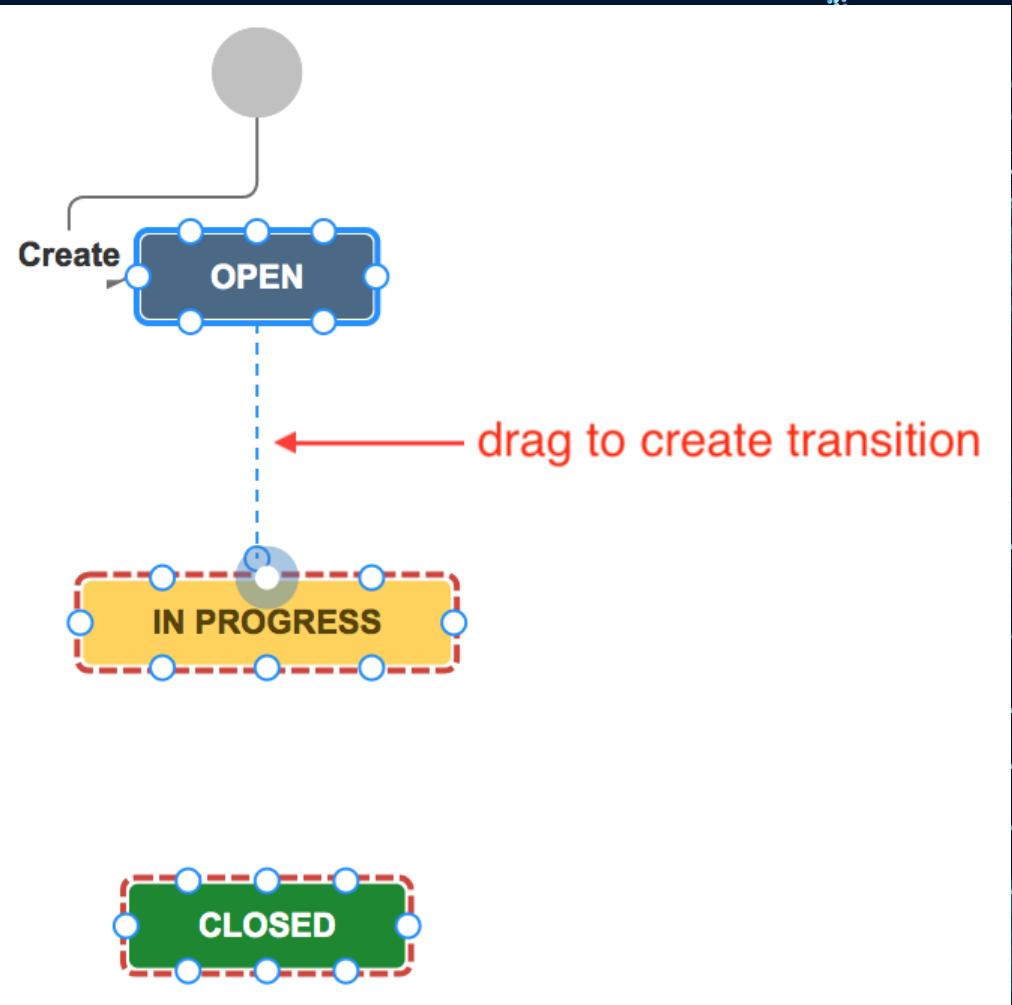
Frozen

Search for an existing status or name a new one.

Allow all statuses to transition to this one

Add Cancel

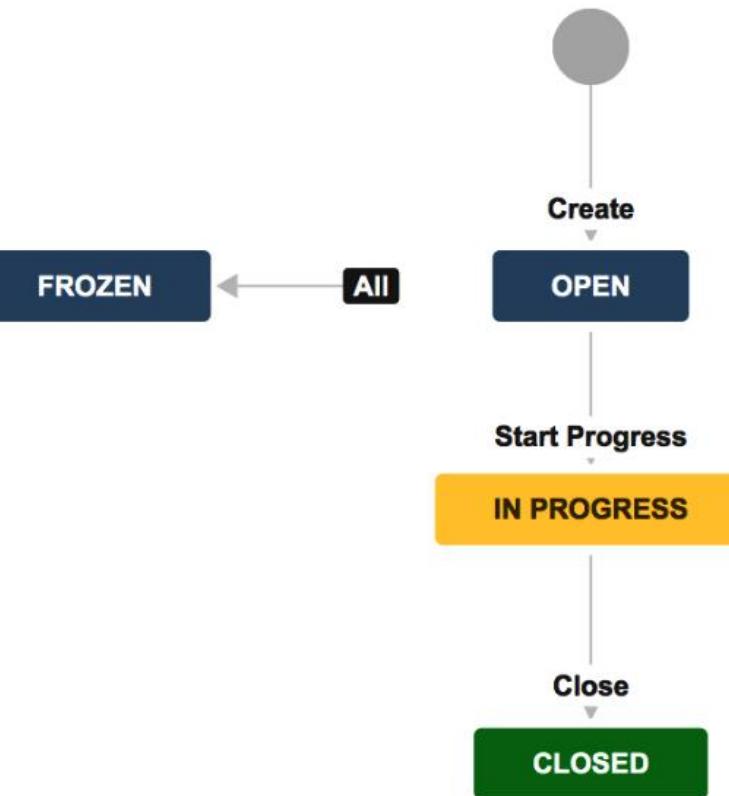
How to do it...



+ Add status

+ Add transition

Show transition labels



How to do it...

1. Select the project to apply the workflow to.
2. Click on the Administration tab to go to the project administration page.
3. Select Workflows from the left-hand side of the page.
4. Click on Add Existing from the Add Workflow menu.
5. Select the new Simple Workflow from the dialog, and click on Next.
6. Choose the issue types to apply (for example, Bug) the workflow to, and click on Finish.

Capturing additional information during workflow transitions

- When users execute a workflow transition, we have an option to display an intermediate workflow screen.
- This is a very useful way of collecting some additional information from the user.
- For example, the default Jira workflow will display a screen for users to select the Resolution value when the issue is resolved.

How to do it...

1. Select the workflow to update, such as our Simple Workflow.
2. Click on the Edit button if the workflow is active. This will create a draft workflow for us to work on.
3. Select the Start Progress transition, and click on the Edit link from the panel on the right-hand side.
4. Select the screen you want to use, such as the workflow screen from the Screen drop-down menu, and click on Save.
5. Repeat step 3 and step 4 to add Resolve Issue Screen to the Close transition.

Using common transitions

- Often, you will have transitions that need to be made available from several different statuses in a workflow, such as the Resolve and Close transitions.
- In other words, these are transitions that have a common destination status but many different originating statuses.

How to do it...

Add Transition

New Transition Reuse a transition

You can reuse a transition provided the destination status is the same.

From status *

To status *

Transition to reuse *

Add **Cancel**

Using global transitions

- While a common transition is a great way to share transitions in a workflow and reduce the amount of management work that would otherwise be required.
- It has the limitation of having to manually create the transitions between the various statuses.
- As your workflow starts becoming more complicated, explicitly creating the transitions becomes a tedious job; this is where global transitions come in.

How to do it...

Frozen

There are no outgoing



transitions from this status.

[Add Transition](#)

- Allow all statuses to transition to this one



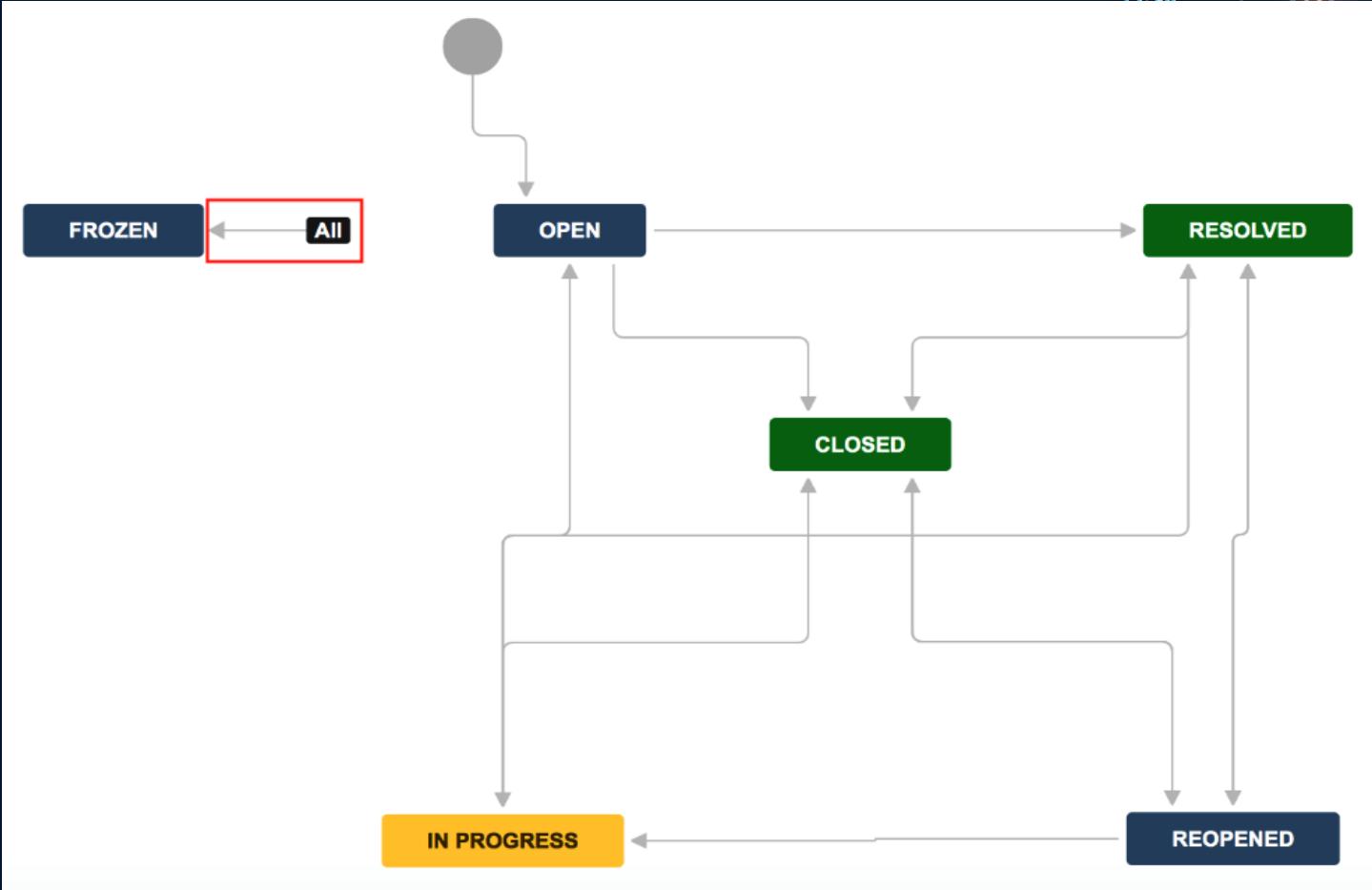
Edit

Remove status

Options

[Properties](#)

How to do it...



Restricting the availability of workflow transitions

- Workflow transitions, by default, are accessible to anyone who has access to the issue.
- There will be times when you want to restrict access to certain transitions.
- For example, you might want to restrict access to the Freeze Issue transition for the following reasons:
 - You want the transition to be available only to users in specific groups or project roles.
 - Since the transition is a global transition, it is available to all the workflow statuses, but it does not make sense to show the transition when the issue is already in the Frozen status.

How to do it...

1. Select and edit the workflow to configure.
2. Select Diagram mode.
3. Click on the Frozen global workflow transition.
4. Click on the Conditions link from the panel on the right-hand side.
5. Click on Add condition, select Value Field (JSU) from the list, and click on Add.
6. Configure the condition with the following parameters:

The Status field for Field

The not-equal sign, !=, for Condition

Frozen for Value

String for Comparison Type

How to do it...

Add Parameters To Condition

Add required parameters to the Condition.

Field: ▼

Choose the field that will be evaluated.

Condition: ▲

Choose a condition for the comparison.

Value:

Value with which the field will be compared to.

Comparison Type: ▲

Choose the type of comparison.

- ⓘ If you choose the comparison type **String**, only '=' and '!=' are valid options. Remember to use the appropriate **Value** to custom field type e.g. date **2014-11-28**, date time **2014-02-16 16:00**

For more information see the [JSU Documentation](#).

Add

Cancel

How it works...

- When a workflow transition option is to be rendered on the page, such as when viewing an issue, all its associated conditions are evaluated to determine if the option should be displayed.
- If there is more than one condition, the transition with conditions will only be displayed when one or all of the conditions pass, depending on the logic setting for the conditions.

There's more...

You can change this so that only one condition needs to pass for the transition to be available by changing the condition group logic from All of the following conditions to Any of the following conditions, as shown below:

Triggers 0 Conditions 2 Validators 0 Post Functions 5

✓ All of the following conditions
Any of the following conditions

value Field (JSU)

- Status != Frozen
- Comparison Type: String

Only users in project role **Developers** can execute this transition.

Validating user input in workflow transitions

- For workflow transitions that have transition screens, you can add validation logic to make sure what the users put in is what you are expecting.
- This is a great way to ensure data integrity, and we can do this with workflow validators.

How to do it...

1. Select and edit the workflow to configure.
2. Select the Diagram mode.
3. Select the Start Progress transition and click on the Validators link on the right-hand side.
4. Click on the Add validator link and select Date Compare (JSU) from the list.
5. Configure the validator with the following parameters:

The Start Date custom field for This date.

The greater-than sign, >, for Condition

Created for Compare with

An optional custom error message, or leave it blank and a default error message will be displayed if the validation fails

How to do it...

Add Parameters To Validator

Add required parameters to the Validator.

This date: ▼

Choose first date field.

Condition: ▲

Choose a condition for the comparison.

Compare with: ▼

Choose second date field.

Include time part:

Choose this option to include time part for the comparison. If the field doesn't have a time part, 00:00:00 will be used instead.

Error message (optional): ▼

Display a customized error message. A default error message will be displayed if nothing is specified.



For more information see the [JSU Documentation](#).

Add

Cancel

How to do it...

In Progress

Assignee*



Patrick Li



Start Date

2/Mar/19



Start date has to be after the issue's create date.

Comment

Style▼

B

I

U

A

▼

^D▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▲

Visual

Text



Viewable by All Users

In Progress

Cancel

Performing additional processing after a transition is executed

- Jira allows you to perform additional tasks as part of a workflow transition through the use of post functions.
- Jira makes heavy use of post functions internally.
- For example, in the case of an out-of-the-box workflow, the resolution field value is cleared automatically when you reopen an issue.

How to do it...

1. Select and edit the workflow to configure.
2. Select the Diagram mode.
3. Create a new workflow transition that will take the issue out of the Frozen status to another status.
4. Click on the Post functions link for the newly created transition.
5. Click on Add post function, select Clear Field Value (JSU) from the list, and click on Add.
6. Select the Reason for Freezing field from Field, and click on the Add button.
7. Click on Publish Draft to apply the change.

How it works...

Triggers 0

Conditions 0

Validators 0

Post Functions 6

The following will be processed after the transition occurs

Add post function

1. **Clear Field Value (JSU)**
 - Field: **Reason for Freezing**
2. Set issue status to the linked status of the destination workflow step.
3. Add a comment to an issue if one is entered during a transition.
4. Update change history for an issue and store the issue in the database.
5. Re-index an issue to keep indexes in sync with the database.
6. Fire a **Generic Event** event that can be processed by the listeners.

custom post function

system post functions



Reacting to events coming from outside of Jira

- When you are using Jira with other systems, especially applications also built by Atlassian, such as Bitbucket and Bamboo, you can get a lot more synergy with them working together.
- One example of such synergy is automatically executing a workflow transition when certain actions happen from the other application.

How to do it...

Add trigger



Pull request created

Automatically transitions the issue when a related pull request is created in a...



Pull request declined

Automatically transitions the issue when a related pull request is declined in a...



Branch created

Automatically transitions the issue when a related branch is created in a connected...



Pull request merged

Automatically transitions the issue when a related pull request is merged in a...



Pull request reopened

Automatically transitions the issue when a related pull request is reopened in a...



Commit created

Automatically transitions the issue when a related commit is made in a connected...



Tell us what other triggers you'd like to see

We are very interested in how you'd like to automate your workflow.

Next

Cancel

Rearranging the workflow transition bar

- By default, workflow transitions are displayed based in the order in which they are defined in the workflow (as listed in text mode); the first two transitions will be shown as buttons, and the remaining transitions will be added to the Workflow menu.
- This sequence is determined by the order in which the transitions are added, so you cannot change that.
- But you can rearrange them by using the opsbar-sequence property.

How to do it...

1. Select and edit the workflow to configure.
2. Select the Frozen global transition.
3. Click on the Properties button.
4. Enter opsbar-sequence as the Property Key and the value 10 in Property Value and click on Add. Click on Publish Draft to apply the change.

How to do it...

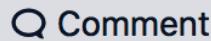


Project Hummingbirds / HUM-18

As a user, I'd like a historical story to show in reports



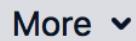
Edit



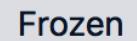
Comment



Assign



More



Frozen



To Do



Workflow



Admin

Details

| | | | |
|--------------------|-----------------|----------------|------------------------------------------------|
| Type: | Story | Status: | TO DO (View Workflow) |
| Priority: | Medium | Resolution: | Unresolved |
| Affects Version/s: | None | Fix Version/s: | Version 2.0 |
| Labels: | None | | |
| Sprint: | Sample Sprint 1 | | |
| Total Comments: | 1 | | |

How it works...

1. The opsbar-sequence property orders the workflow transitions numerically, from the smallest to the largest.
2. Its value needs to be a positive integer.
3. The smaller the number, the closer the transition will appear to the front.

There's more...

- Jira only displays the first two transitions as buttons. You can change this setting by editing the ops.bar.group.size.opsbar-transitions property in the jira-config.properties file located in your JIRA_HOME directory.
- All you have to do is edit the file, set the property to the desired number of transition buttons to display as shown (we are setting the number of transition buttons to 3), and restart Jira:

```
ops.bar.group.size.opsbar-transitions=3
```

There's more...



Project Hummingbirds / HUM-18

As a user, I'd like a historical story to show in reports

Edit

Comment

Assign

More ▾

Frozen

To Do

In Progress

Workflow ▾

Admin ▾

Details

3 transition buttons

Type:

Story

Status:

TO DO (View Workflow)

Priority:

Medium

Resolution:

Unresolved

Affects Version/s:

None

Fix Version/s:

Version 2.0

Labels:

None

Sprint:

Sample Sprint 1

Total Comments:

1

Restricting the resolution values in a transition

- Normally, issue resolution values such as **Fixed** and **Won't Fix** are global.
- So regardless of the project and issue type, the same set of values will be available.
- As you implement different workflows in Jira, you may find that certain resolutions are not relevant for a given project or issue type.

How to do it...

1. Select and edit the Simple Workflow.
2. Select the Close Issue workflow transition.
3. Click on the Properties option.
4. Enter `jira.field.resolution.include` for the Property Name and the IDs (comma separated) for resolutions we want to make available into Property Value. So, if we want to only include the Done, Won't Do, and Duplicate resolutions, we need to specify the 10000, 10001, and 10002 values as the property values, with no spaces in between.

How to do it...

Workflows / Simple Workflow / Close

Transition Properties

Property Key

Property Value

jira.field.resolution.include

10000,10001,10002

Add

jira.i18n.title

gh.workflow.preset.done

Delete

Preventing issue updates in selected statuses

- By default, when an issue is in the Closed status, it cannot be updated.
- It is a good practice to make issues read-only when they are in a status that signifies logical completion.

How to do it...

1. Select and edit the workflow to update.
2. Select the Frozen workflow status.
3. Click on the Properties link from the panel on the right-hand side.
4. Enter jira.issue.editable for Property Key and false into Property Value, and click on Add.
5. Click on Publish Draft to apply the change.

Making a field required during a workflow transition

- Using field configuration to make a field required will make the field required all the time.
- There are many use cases where you will only need the field to be required during certain workflow transitions.

How to do it...

Add Parameters To Validator

Add required parameters to the Validator.

Fields required:

Epic Name
Epic Status
Fix Version/s
issueFunction
Labels
Last Viewed
Original Estimate
Priority
Rank
Remaining Estimate

Available fields:

Required fields:

Reason for Freezing

Add >>

<< Remove

Error message (optional):

Display a customized error message. A default error message will be displayed if nothing is specified.

Ignore context:

If checked a field will be required by this checker, even if its context is not configured for the current issue.



For more information see the [JSU Documentation](#).

Add

Cancel

How to do it...

Frozen

Assignee*

 Patrick Li

Reason for Freezing*

You must provide a reason for freezing this issue.

Comment

Style ▾ B I U A A ^A _A C ▾ E E ☺ + ▾

Visual Text ↪ ↫ 🔍 Viewable by All Users

Frozen

Cancel

Creating custom workflow transition logic

- We will take a look at how to use scripts to define our own validation rules for a workflow validator.
- We will address a common use case, which is to make a field required during a workflow transition only when another field is set to a certain value.

How to do it...

1. Select and edit the Simple Workflow.
2. Select Diagram mode.
3. Click on the Done global workflow transition.
4. Click on the Validators link from the panel on the right-hand side.
5. Click on Add validator, select Script Validator [ScriptRunner] from the list, and click on Add.
6. Select the Simple scripted validator option.

How to do it...

7. Perform the following steps to set up a validator with custom-scripted logic

```
import com.opensymphony.workflow.InvalidInputException
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.issue.CustomFieldManager
import org.apache.commons.lang3.StringUtils

def customFieldManager = ComponentAccessor.getCustomFieldManager()
def solutionField = customFieldManager.getCustomFieldObjectByName("Solution Details")
def resolution = issue.getResolution().getName()
String solution = issue.getCustomFieldValue(solutionField)

if(resolution == "Done" && StringUtils.isBlank(solution)) {
    false
} else {
    true
}
```

How to do it...

Simple scripted validator [?](#)

Runs a simple embedded script to find out whether to allow the transition or not

Note

An optional note, used only for your reference.

Condition

```
1 import com.opensymphony.workflow.InvalidInputException
2 import com.atlassian.jira.component.ComponentAccessor
3 import com.atlassian.jira.issue.CustomFieldManager
4 import org.apache.commons.lang3.StringUtils
5
6 def customFieldManager = ComponentAccessor.getCustomFieldManager()
7 def solutionField = customFieldManager.getCustomFieldObjectByName("Solution")
8
9 def resolution = issue.getResolution().getName()
10 String solution = issue.getCustomFieldValue(solutionField)
11
12 if(resolution == "Done" && StringUtils.isBlank(solution)) {
13     false
14 } else {
15     true
16 }
17
```

Show examples 



Enter the condition for which this function will fire. Blank will evaluate to "true". You can click one of the examples below, or see the wiki page for further examples.

Error Message

You must provide the Solution Details if the Resolution is set to Done.

Error message that will be provided to the user if the condition is not true

Field

Solution Details

Form field that the error message will appear against. Leave blank for it to appear at the top.

Preview

Update

Cancel

How to do it...

Done

Resolution* Done ?

Assignee* Patrick Li

Solution Details

You must provide the Solution Details if the Resolution is set to Fixed.

Comment

Style B I U A ° ♂ ♀ ≡ ≡ 😊 + ✖

Visual Text ↶ ↷ 🔒 Viewable by All Users

Done Cancel

How it works...

- We first get the Solution Details custom field via its name, as shown in the following line of code. If you have more than one custom field with the same name, you need to use its ID instead of its name:

```
def solutionField = customFieldManager.getCustomFieldObjectByName("Solution Details")
```

- We then select the resolution value and obtain the value entered for Solution Details during the transition, as follows:

```
def resolution = issue.getResolutionObject().getName()
```

```
String solution = issue.getCustomFieldValue(solutionField)
```

User Management

Creating and importing multiple users

- As a Jira administrator, it is usually your responsibility to set up accounts for the new user whenever someone new joins the organization.
- This is usually fine on an ad hoc basis, but from time to time, you might be required to import many users at once.

How to do it...

| Username | Password | Email | Full name | Group A |
|-------------|----------|-------------------------|--------------|------------------------|
| tester 1 | XXXXX | tester1@example.co m | Test User | jira- softwareusers |

How to do it...

1. Unzip the CLI Client into a directory on your computer (for example, /opt/cli).
2. Copy the user's CSV file to a directory on your computer (for example, /tmp/users.csv).
3. Open a Command Prompt and navigate to the directory that contains the CLI Client—that is, the directory that contains the jira.sh or jira.bat file.
4. Make sure that the jira.shfile (Linux) or jira.bat (Windows) file is executable.
5. Run the following command to import users; make sure you substitute the administrator username and password in your Jira URL:

```
./jira.sh --action addUserWithFile --server
```

```
http://localhost:8080 --password <password>
```

```
--user <username> --file /tmp/users.csv
```

How to do it...

If everything runs fine, you will see an output similar to the following one on your console:

```
User: tester1 added with password: xxxxx. Full name is: Tester One. Email is: tester1@example.com.  
User: tester2 added with password: yyyyy. Full name is: Tester Two. Email is: tester2@example.com.  
User: tester3 added with password: zzzzz. Full name is: Tester Three. Email is: tester3@example.com.  
User: tester4 added with password: 9vwybjvmlbjs. Full name is: Tester Four. Email is: tester4@example.com.  
Successful adds: 4 errors: 0 already defined users: 0
```

How it works...

- The command-line client that we used to run the `addUserWithFile` command uses Jira's remote APIs to interact with Jira.
- Jira exposes many of its core functionalities via these APIs, such as the ability to create new users and issues.
- When we run the `addUserWithFile` command, we pass in the CSV file that contains our new users, formatted in a way that the command-line client is able to understand and make an API call to Jira to create those users for us.

Enabling public user signup

- In the previous recipe, we looked at how to manually create new user accounts and import users from a CSV file.
- These are the two options that the Jira administrators have when your Jira instance is used internally.

How to do it...

1. Navigate to Administration > System > General configuration.
2. Click on the Edit Settings button.
3. Set the Mode option to Public and click on Update.

How it works...

Login



Username

Password

Remember my login on this computer

Not a member? [Sign up for an account.](#)

[Log In](#)

[Can't access your account?](#)

There's more...

1. Navigate to Administration > System > General configuration.
2. Click on the Edit Settings button.
3. Set the CAPTCHA on signup option to On, and click on Update.

There's more...

Sign up

Email* E.g. charlie@atlassian.com

Full name* Your full name

Username* Desired username

Password* Password

Please enter the word as shown below



houters

Sign up

Cancel

Managing groups and group memberships

- In any information system, a common way of managing users is through the use of groups.
- Groups are based on positions and responsibilities within an organization; however, it is important to note that groups simply represent a collection of users.
- In Jira, groups provide an effective way to apply configuration settings, such as permissions and notifications, to users.

How to do it...

1. Navigate to Administration > User management > Groups.
2. Enter the new group's name under the Add group section.
3. Click on the Add group button.

How to do it...

Groups

Bulk edit group members

This page allows you to edit the user memberships for each group.

You can add to and remove users from multiple groups at a time. When selecting multiple groups please note:

- All the common users in the selected groups are displayed under the 'All' label and the remaining disparate users are displayed under the label with its group name.
- **Removing Users** - Removing user(s) listed in the 'All' section will remove the selected user(s) from all of the selected groups. However if user(s) are selected under a specific group name(s), the selected user(s) will be removed from the group its listed under.
- **Adding Users** - All user(s) to be added are added to all of the selected group(s).

Step 1: Select group(s) to edit and refresh the members list

Step 2: Select users to leave OR join the selected group(s) and click on the corresponding button

Selected 1 of 2 groups

jira-administrators ×

1 Group member(s)

jira-administrators
patrick

Add members to selected group(s)

christine, john



Begin typing to find users.

Add selected users

Remove selected users

There's more...

Manage user groups

Type to start searching.

jira-administrators ×

Join selected groups

Current Groups

jira-software-users

Leave selected groups

Cancel



Managing project roles

- Using groups is the default method of managing multiple users in Jira; however, there are some limitations with using groups.
- The first limitation is that groups are global in Jira.
- This means that if a user is in a group, then that user is included in all projects in that group.

How to do it...

Project Role Browser



You can use project roles to associate users and/or groups with specific projects. The table below shows all the project roles that are available in Jira. Use this screen to add, edit and delete project roles. You can also click 'View Usage' to see which projects, permission schemes and notification schemes are using project roles.

| Project Role Name | Description | Actions | | |
|-------------------|------------------------------------------------------------|----------------------------|----------------------|------------------------|
| Administrators | A project role that represents administrators in a project | View Usage | Edit | Delete |
| Developers | | View Usage | Edit | Delete |

Add Project Role

Name

Description

[Add Project Role](#)

How to do it...

Users and roles

Defaults

Project Lead  Patrick Li

Default Assignee Project Lead

Users by role

Roles: All ▾

 Search by name or email

Administrators Showing 1 of 1

Name

Username

 jira-administrators

Developers Showing 1 of 1

Name

Username

Add users to a role | Edit defaults

Add users to a role

Users or groups

 Search by name or email

 Christine Johnson

Role

Project Manager

Add

Managing default project role memberships

Project role memberships are defined per project; however, there are cases where certain users or groups need to be members of a given project role by default. In fact, Jira has the following default members out of the box:

- Administrators: All members of the jira-administrators group
- Developers: All members of the jira-developers group

How to do it...

Edit Default Members for Project Role: Project Manager



The table below shows the default members (i.e. users, groups) for a project role.

NOTE: When a new project is created, it will be assigned these 'default members' for the 'Project Manager' project role. Note that 'default members' apply only when a project is created. Changing the 'default members' for a project role will not affect role membership for existing projects.

- [Return to Project Role Browser](#)

Default Users

Christine Johnson [Edit](#)

Default Groups

project-owners [Edit](#)

How it works...

- Once you have assigned users and groups as the default members of a project role, any newly created project will have those users and groups added to the role.
- A good practice is to use groups for the default project role membership, as a user's role and responsibilities are likely to change over time.

Deactivating a user

- Once a user has created an issue or comment, Jira will not allow you to delete the user.
- In fact, deactivating a user is usually a better approach than deleting the user completely.
- Once the user is deactivated, the user cannot log in to Jira, and this will not count toward your license count.

How to do it...

1. Navigate to Administration > User management > Users.
2. Click on the Edit link for the user that is to be deactivated.
3. Uncheck the Active option.
4. Click on the Update button to deactivate the user.

Integrating and importing users from LDAP

- By default, Jira manages its users and groups internally.
- Most organizations today often use LDAP, such as Microsoft Active Directory (AD), for centralized user management, and you can integrate Jira with LDAP.
- Jira supports many different types of LDAP, including ADn, OpenLDAP and more.

Getting ready

- The hostname and port number of the LDAP server.
- The base Distinguished Name (DN) to search for users and groups.
- The credentials to access the LDAP server. If you want Jira to be able to make changes to LDAP, make sure that the credentials have write permissions.

How to do it...

1. Navigate to Administration > User management > User Directories.
2. Click on the Add Directory button and select either Microsoft Active Directory or LDAP for non-AD directories.
3. Enter the LDAP server, schema, and permission settings. Refer to the following table for more details.
4. Click on the Quick Test button to validate Jira's connectivity to LDAP.
5. Click on the Save and Test button if there are no issues with connecting to LDAP.
6. Type in a username and password to run a quick test. While doing this, make sure that Jira is able to connect to LDAP, to find the user and retrieve the user's group information, and to authenticate against LDAP.

How to do it...

| Server settings | Description |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | This is an identifier for the LDAP server. |
| Directory Type | This selects the type of the LDAP server—for example, Microsoft Active Directory. Jira automatically fills in the user and group schema details based on the type selected. |
| Hostname | This is the host of the LDAP server is hosted. |
| Port | This is the port that the LDAP server listens to for incoming connections. |
| Use SSL | This checks whether SSL is being used on LDAP. |
| Username | This is the user account that Jira uses to access LDAP. This should be a dedicated account for Jira. |
| Password | This is the password for the account. |

How to do it...

| LDAP schema | Description |
|---------------------|--------------------------------------------------------------------------|
| Base DN | This is the root node where Jira starts the search for users and groups. |
| Additional User DN | This is the additional DN to further restrict a user search. |
| Additional Group DN | This is the additional DN to further restrict a group search. |

How to do it...

| LDAP permission | Description |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Read Only | Select this option if you do not want Jira to make any changes to LDAP. This is the ideal option if everything, including the user's group memberships, is managed with LDAP. |
| Read Only, with Local Groups | This option is similar to the Read Only option but lets you manage group memberships locally within Jira. With this option, the group membership changes you make will remain in Jira only. This is the ideal option when you only need user information from LDAP and want to manage Jira-related groups locally. |
| Read/Write | Select this option if you want Jira to be able to make direct changes to LDAP, assuming that Jira's LDAP account has the write permission as well. |

How to do it...

Test Remote Directory Connection

Use this form to test the connection to Microsoft Active Directory (Read Only) directory 'Microsoft Directory server'.

For extended testing enter the credentials of a user in the remote directory.

 Test basic connection : Succeeded

 Test retrieve user : Succeeded

 Test get user's memberships : Succeeded, 16 groups retrieved

 Test retrieve group : Succeeded

 Test get group members : Succeeded, 1 users retrieved

 Test user can authenticate : Succeeded

User name

patrick

Password

[Test Settings](#)

[Edit Settings](#)

[Back to directory list](#)

How it works...

- All authentication will be delegated to LDAP. So, if a user's password is updated in LDAP, it will be immediately reflected when the user attempts to log in to Jira.
- It is important to note that, with LDAP, users must still be in the necessary groups (for example, jira-users, by default) in order to access Jira.
- So you need to make sure that you either create a group called jira-users in LDAP and add everyone to it or grant the application access to LDAP groups, such as a group called all-employees.

Integrating with LDAP for authentication only

- Sometimes, you might need LDAP only for authentication, and want to keep the group membership separate from LDAP for easy management.
- In this recipe, we will look at how to integrate Jira with LDAP only for authentication.

How to do it...

1. Navigate to Administration > User management > User Directories.
2. Click on the Add Directory button and select the Internal with LDAP Authentication option.
3. Enter the LDAP server and schema settings. Most of the parameters are identical to those you use when creating a normal LDAP connection, with a few exceptions. Refer to the following table for details.
4. Click on the Quick Test button to validate Jira connectivity to LDAP.
5. Click on the Save and Test button if there are no issues connecting to LDAP.

How to do it...

| Server settings | Description |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Copy User on Login | This automatically copies the user from LDAP into Jira when the user first successfully logs in to Jira. |
| Default Group Membership | This automatically adds the user into the groups specified here when the user first successfully logs in to Jira. This setting is not retrospectively applied to existing users. This is a useful feature to ensure that every user who can log in to Jira will be added to the necessary groups, such as jira-users . |
| Synchronize Group Memberships | This automatically copies the user's group membership to Jira when the user successfully logs in. |

How it works...

This authentication option is similar to the previous recipe, but with a number of key differences:

- LDAP is only used for authentication.
- Jira does not periodically synchronize the user and group information from LDAP after the initial user login.
- Jira has read-only access to LDAP.
- Group membership is managed inside Jira.

Integrating with Atlassian Crowd

- Crowd is a user-identity-management solution from Atlassian, and Jira supports Crowd integration out of the box.
- With Crowd, you can also set up a single sign-on option with other Crowd-enabled applications.

Getting ready

At a minimum, you will also need to have the following information:

- The Crowd server URL
- Credentials for the registered application in Crowd for Jira

How to do it...

1. Navigate to Administration > User management > User Directories.
2. Click on the Add Directory button and select the Atlassian Crowd option.
3. Enter the Crowd server settings. Refer to the following table for details.
4. Click on the Test Settings button to validate Jira's connectivity with Crowd.
5. Click on the Save and Test button if there are no issues connecting with Crowd

How to do it...

| Server settings | Description |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | This is an identifier for the Crowd server. |
| Server URL | This is the Crowd's server URL. |
| Application Name | This is the registered application name for Jira inside Crowd. |
| Application Password | This is the password for the registered application. |
| Crowd Permissions | This is the column header. |
| Read Only | Select this option if you do not want Jira to make any changes to Crowd. This is the ideal option if everything, including the user's group membership, is managed with Crowd. |
| Read/Write | Select this option to let Jira synchronize any changes back to Crowd. |

How to do it...

| Advanced settings | Description |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Enable Nested Groups | This allows groups to contain other groups as members. |
| Enable Incremental Synchronization | This will only synchronize deltas. Enabling this option can help improve performance. |
| Synchronization Interval | This determines how often (in minutes) Jira should synchronize with Crowd for changes. Shorter intervals may cause performance issues. |

Setting up a single sign-on functionality with Crowd

- Web-based applications integrated with Crowd are able to participate in an SSO environment
- So when a user is logged in to one application, he/she will be automatically logged in to all other applications.

Getting ready

If you have already integrated Jira with Crowd, you will need to have the following information:

- The application name assigned to Jira in Crowd
- The password for Jira to access Crowd
- A copy of the crowd.properties file from the CROWD_INSTALL/client/conf directory

How to do it...

1. Shut down Jira if it is running.
2. Open the seraph-config.xml file located in the JIRA_INSTALL/atlassian-jira/WEB-INF/classes directory in a text editor.
3. Locate the line that contains
com.atlassian.jira.security.login.JiraSeraphAuthenticator. Comment it out so that it looks like the following:

```
<!--  
<authenticator class="com.atlassian.jira  
.security.login.JiraSeraphAuthenticator"/>  
-->
```

How to do it...

4. Locate the line that contains com.atlassian.jira.security.login.SSOSeraphAuthenticator. Uncomment it so that it looks like the following:

```
<authenticator class="com.atlassian.jira  
.security.login.SSOSeraphAuthenticator"/>
```

How to do it...

| Parameter | Value |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>application.name</code> | This is the application name configured in Crowd for Jira. |
| <code>application.password</code> | This is the password for the application. |
| <code>application.login.url</code> | This is Jira's base URL (you can get this from Jira's general configurations). |
| <code>crowd.base.url</code> | This is Crowd's base URL. |
| <code>session.validationinterval</code> | This is the duration (in minutes) that a Crowd SSO session will remain valid. Setting this to <code>0</code> will invalidate the session immediately, and will have a performance penalty. It is recommended that you set this at a higher value. |

Setting up a single sign-on with Google

- If your organization uses Google to manage user details, or you simply want to allow people with valid Google accounts to be able to log into your Jira instance (especially if it is a public instance).
- You can integrate Jira with Google so that users can log into Jira with their Google account details with a single click.

How to do it...

1. Browse to Google API Console at
<https://console.developers.google.com/apis/credentials>.
2. Click on the Create credentials drop-down menu.
3. Select OAuth client ID and then Web application.
4. Enter a name for the API credential and your Jira's URL for authorized JavaScript origins.
5. Create the new API credential and note down the Client ID and Client Secret: we will need them both later.

How to do it...

Configuration License

⚠ Web Sudo warning!

Because web sudo requires you to re-enter a password when accessing administration functions. Admin users using Google SSO will be unable to access functions behind web sudo protected prompts. We recommend disabling this option.

Client ID*

The client id assigned by Google upon project client registration

Client Secret*

The client secret assigned by Google upon project client registration

Sign-in button position

- Before the host application login fields
- After the host application login fields
- A custom JQuery Selector to determine login button position

User creation policy

- Application users must be created manually by an administrator
- Create application users automatically upon first successful Google sign-in

Save **Cancel**

How to do it...



Welcome to Jira

Username

Password

Remember my login on this computer

Not a member? [Sign up](#) for an account.

[Log In](#) [Can't access your account?](#)

 [Sign in with Google](#)

Setting up a Windows domain single sign-on

- If your organization is running a Windows domain, you can configure Jira so that users are automatically logged in when they log in to the domain with their workstations.

How to do it...

1. Shut down Jira if it is running.
2. Copy login.conf, krb5.conf, and spnego-exclusion.properties to the JIRA_INSTALL/atlassian-jira/WEB-INF/classes directory.
3. Copy appfusions-jira-seraph-4.0.0.jar and appfusions-spnego-r7_3.jar to the JIRA_INSTALL/atlassian-jira/WEB-INF/lib directory.
4. Open the web.xml file located in the JIRA_INSTALL/atlassian-jira/WEB-INF directory in a text editor.

How to do it...

5. Add the following XML snippet before the THIS MUST BE THE LAST FILTER IN THE DEFINED CHAIN entry. Make sure you update the values for the following parameters:

For spnego.krb5.conf, use the full path to the spnego.krb5.conf file.

For spnego.login.conf, use the full path to the spnego.login.conf file.

For spnego.prauth.username, use the username of the service account.

For spnego.prauth.password, use the password of the service account:

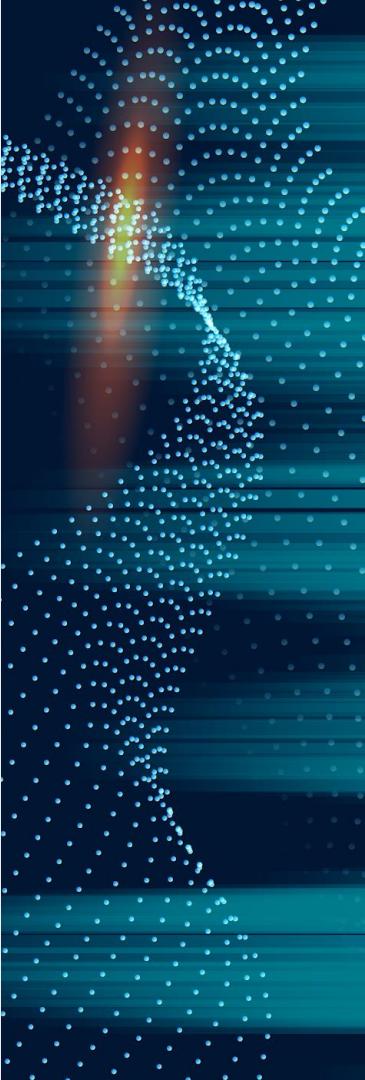
```
<filter>
  <filter-name>SpnegoHttpFilter</filter-name>

  <filter-class>net.sourceforge.spnego
    .SpnegoHttpFilter</filter-class>

  <init-param>
    <param-name>spnego.allow.basic</param-name>
    <param-value>true</param-value>
  </init-param>

  <init-param>
    <param-name>spnego.allow.localhost
    </param-name>
    <param-value>true</param-value>
  </init-param>

  <init-param>
    <param-name>spnego.allow.unsecure.basic
    </param-name>
```



```
      <param-value>true</param-value>
    </init-param>

    <init-param>
      <param-name>spnego.login.client.module
      </param-name>
      <param-value>spnego-client</param-value>
    </init-param>

    <init-param>
      <param-name>spnego.krb5.conf</param-name>
      <param-value>FULL_PATH/krb5.conf
      </param-value>
    </init-param>

    <init-param>
      <param-name>spnego.login.conf</param-name>
```

```
<param-value>FULL_PATH/login.conf
</param-value>

</init-param>

<init-param>

<param-name>spnego.preauth.username
</param-name>

<param-value>SPN_USERNAME</param-value>

</init-param>

<init-param>

<param-name>spnego.preauth.password
</param-name>

<param-value>SPN_PASSWORD</param-value>

</init-param>

<init-param>

<param-name>spnego.login.server.module
</param-name>

<param-value>spnego-server</param-value>

</init-param>

<init-param>

<param-name>spnego.prompt.ntlm</param-name>

<param-value>true</param-value>

</init-param>

<init-param>

<param-name>spnego.logger.level</param-name>

<param-value>1</param-value>

</init-param>

<init-param>

<param-name>spnego.skip.client.internet
</param-name>

<param-value>false</param-value>

</init-param>

</filter>
```

How to do it...

6. Add the following XML snippet before the login entry:

```
<filter-mapping>  
  <filter-name>SpnegoHttpFilter</filter-name>  
  <url-pattern>*</url-pattern>  
</filter-mapping>
```

7. Open the seraph-config.xml file located in the JIRA_INSTALL/atlassian-jira/WEB-INF/classes directory in a text editor.

How to do it...

8. Locate the line that contains com.atlassian.jira.security.login.JiraSeraphAuthenticator. Comment it out so that it looks like the following:

```
<!--  
<authenticator class=  
"com.atlassian.jira.security  
.login.JiraSeraphAuthenticator"/>
```

```
-->
```

9. Add the following XML snippet under the line that's been commented out:

```
<authenticator  
class="com.appfusions.jira.SeraphAuthenticator"
```



Jira Security

Granting access to Jira

- Since Jira is now a platform that includes Jira Core, Jira Software, Jira Service Desk, and other third-party applications,
- You can have multiple applications running on the same platform instance.
- This is because user access is granted on a per-application level.
- We will look at managing access to applications in Jira.

How to do it...

Application access

A user must belong to a group assigned to an application to be able to log in and access that application. When you create a user for a Jira application, that user is automatically added to the application's default group. Additional permissions can be assigned to a group via [global permissions](#).

JIRA Service Desk Unlimited agent (1 used) ⓘ

| Name | Default | |
|------------------------------------------------|-------------------------------------|------------------------|
| jira-servicedesk-users (1 user) | <input checked="" type="checkbox"/> | Remove |
| <input type="button" value="Select group..."/> | | |
| external-contractors | | |
| jira-administrators | | |
| jira-software-users | | |
| project-owners | | |

default groups new users will be added to

| Name | Default | |
|------------------------------------------------|-------------------------------------|------------------------|
| jira-administrators (1 user) | <input type="checkbox"/> ADMIN | Remove |
| jira-software-users (4 users) | <input checked="" type="checkbox"/> | Remove |
| <input type="button" value="Select group..."/> | | |
| Select group to add it to the application | | |

How it works...

- Starting with JIRA 7, Atlassian introduced the new concept of applications.
- This turns Jira into a platform, and major features, such as JIRA Agile (now called Jira Software) and Jira Service Desk, are now separate applications that run on the Jira platform.
- These changes mean that you can now control user access to each of the applications individually.
- Instead of using permissions to control who can access Jira, you can assign access rights based on the application.

Granting Jira System Administrator access

- We will take a look at how to grant administrative access to users.
- In the same way as granting user access, you can only grant administrative access to a group of users.

How to do it...

Jira Permissions



Permissions

Users / Groups

Jira System Administrators

Ability to perform all administration functions. There must be at least one group with this permission.

- jira-administrators
[View Users](#) [Delete](#)

Jira Administrators

Ability to perform most administration functions (excluding Import & Export, SMTP Configuration, etc.).

- jira-administrators
[View Users](#) [Delete](#)

Add Permission

Permission

Jira System Administrators



Group

jira-administrators



Add

How it works...

- There are two levels of administrative access in Jira: Jira Administrator and Jira System Administrator.
- For the most part, they have identical functions when it comes to Jira configurations, such as custom fields and workflows.
- Jira System Administrators have additional access to system-wide application configurations, such as the SMTP mail server configuration, installing apps, and updating Jira licenses.

Controlling access to a project

- In the previous recipes, we looked at how to use global permissions to control Jira access and administrator-level access.
- In this recipe, we will demonstrate how to control project-level permissions, starting with access to projects.

How to do it...

Grant permission

Permission Browse Projects x Add Comments x

Granted to Project Role
 Application access
 Group
jira-software-users ▼

Reporter
 Single user
 Project lead
 Current assignee
 Group custom field value

[Show less](#)

Grant Cancel

How to do it...

Project Permissions

Permissions define who can access a project, and what they can do within the project, and with the issues within.

The permission scheme defines how the permissions are set up for this project. To change the permissions, you can use a different permission scheme, or modify the current scheme.

[Learn more about project permissions](#)

i Scheme used by this project: Marketing project permission scheme USED BY 1 PROJECT

Permission helper

Actions ▾

Edit permissions

Use a different scheme

How it works...

- Permission schemes define project-level permissions.
- Unlike global permissions, which can only be granted to groups, these can be granted to specific users, groups, project roles, and more.
- The option you choose here will depend on your use case.
- Generally speaking, groups will be the most straightforward option, as users often belong to one or more groups and you can easily model your permission requirements based on groups.

Controlling access to Jira issue operations

- In this recipe, we will take a look at permissions that control issue operations.
- These are the operations that your end users will perform on a daily basis, including create, edit, delete, comment, and more.

How to do it...

Project Permissions

Permission helper

Actions ▾

Permissions define who can access a project, and what they can do within the project, and with the issue types.

The permission scheme defines how the permissions are set up for this project. To change the permission scheme, use a different scheme or modify the current scheme.

[Learn more about project permissions](#)

[Edit permissions](#)

[Use a different scheme](#)

i Scheme used by this project: Marketing project permission scheme USED BY 1 PROJECT

Project permissions

| Permission | Granted to |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Administer Projects Ability to administer a project in Jira. Extended project administration ENABLED Grant extended project administration permissions. | Project role <ul style="list-style-type: none">Administrators |
| Grant extended project administration permissions. | |

Permission helper

Discover why a user does or does not have certain permissions...

User

Patrick Li

Begin typing to find a user, leave blank for Anonymous user

Issue

DEMO-7 - How does Jira permissions work?

Begin typing to find an issue

Permission

Edit Issues

Begin typing to find a permission or press down to see all



Permission name: Edit Issues

User: Patrick Li

Project: Demonstration Project

Permission scheme: Default Permission Scheme

Issue: DEMO-7

Status: ✖ Patrick Li does not have the 'Edit Issues' permission

Status

Summary

Details



Issue Security

Patrick Li is not in security level Internal Only



Application Access

Any logged in user has this permission

Submit

There's more...

Allowing users to control permissions

- When you have a mixed group of users (such as internal employees and outside consultants) working on the same Jira project, there will be instances regarding sensitive information that should only be viewed by internal employees.
- In these cases, you will want to mark those instances as internal only, so that other people cannot see them.

How to do it...

1. Navigate to Administration > Issues > Issue security schemes.
2. Click on the Add issue security scheme link.
3. Enter a name for the new scheme and then click on Add.

How to do it...

1. Click on the Security Levels link for our new issue security scheme.
2. Enter the name for each security level and click on the Add Security Level button.

The following screenshot shows the three existing security levels:

Edit Issue Security Levels

On this page you can create and delete the issue security levels for the "Internal and External Users" issue security scheme. Each security level can have users/groups assigned to them.

An issue can then be assigned a Security Level. This ensures only users who are assigned to this security level may view the issue. Once you have set up some Security Levels, be sure to grant the "Set Issue Security" permission to relevant users.

▪ [View all Issue Security schemes](#)

| Security Level | Users / Groups / Project Roles | Actions |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| Internal Users Only | • Group (jira-software-users) (Delete) | Add Default Edit Delete |
| Internal and External Users | • Group (jira-software-users) (Delete) • Group (external-contractors) (Delete) | Add Default Edit Delete |
| Reporter and Assignee Only | • Reporter (Delete) • Current assignee (Delete) | Add Default Edit Delete |

Add Security Level

Add a new security level by entering a name and description below.

Name

Description

[Add Security Level](#)

How to do it...

1. Click on the Add link for the security level you want to set up the user access for.
2. Select the permission option and click on the Add button.

How to do it...

Add User/Group/Project Role to Issue Security Level

Issue Security Scheme: **Internal and External Users**

Issue Security Level: **Internal Users Only**

Please select a user or group to add to this security level.

This will enable the specific users/groups to view issues for projects that:

- are associated with this Issue Security Scheme and
- have their security level set to **Internal Users Only**

| | |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input type="radio"/> Application access | <input type="text" value="Any logged in user"/>  |
| <input checked="" type="radio"/> Reporter | |
| <input type="radio"/> Group | <input type="text" value="Anyone"/>  |
| <input type="radio"/> Single user | <input type="text"/>  Start typing to get a list of possible matches. |
| <input type="radio"/> Project lead | |
| <input type="radio"/> Current assignee | |
| <input type="radio"/> User custom field value | <input type="text" value="Choose a custom field"/>  |
| <input type="radio"/> Project Role | <input type="text" value="Choose a project role"/>  |
| <input type="radio"/> Group custom field value | <input type="text" value="Choose a custom field"/>  |

Add

Cancel

How to do it...



Associate Issue Security Scheme to Project

Step 2 of 2: Associate any issues in this project that previously had their security level set, with a security level from the new scheme.

Selecting a new level will change the security level of all the affected issues to be the newly selected security level

There are no previous secured issues.

[Associate](#)

[Cancel](#)



How it works...

Edit Issue : HUM-18

Configure Fields ▾

Summary * As a user, I'd like a historical story to show in reports

Issue Type * Story

Security Level

- None
- Internal Users Only
- Internal and External Users
- Reporter and Assignee Only

Assignee *

Description

Style **I** A ^{A²} _{S₂}

Tell us in details what the problem you are having.

How to reproduce

List out the steps to reproduce the problem.

- step 1
- step 2

Expected Result

Tell us what you think the correct outcome should be after completing the steps listed

Update Cancel

How it works...



You can't view this issue

It may have been deleted or you don't have permission to view it.

Delegating administrator permissions

- Most customization and configuration options, such as workflows and screens, are managed by Jira Administrators.
- While this model may work well for a small organization, it often causes a bottleneck for larger organizations.
- Where many customizations to these core Jira functions are required and there are only a few Jira Administrators who can actually make the necessary changes.

How to do it...

Project permissions

| Permission | Granted to | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------|------------------------|
| Administer Projects Ability to administer a project in Jira. <input checked="" type="checkbox"/> Extended project administration Grant extended project administration permissions . | Project role <ul style="list-style-type: none">Administrators | Edit | Remove |
| Browse Projects Ability to browse projects and the issues within them. | Application access <ul style="list-style-type: none">Any logged in user | Edit | Remove |

How it works...

- Workflows cannot be used by other projects.
- Only statuses that already exist in Jira can be added to workflows. Project administrators cannot create new statuses.
- The existing status in a workflow can be deleted only if no issues in the project are currently in the status.
- Workflow transitions can be deleted from the workflow, but details such as properties, validators, conditions, post-functions, and screens cannot be changed.
- Only non-system screens, that is, screens created by Jira Administrators, can be updated.
- Screens cannot be used by other projects.
- Only existing fields, both custom and system, can be added to screens.

Restricting access to projects based on reporter permissions

- As you have seen in one of the previous recipes, the Browse Projects permission controls who can access a project in Jira.
- In this recipe, we will set up permissions so that users can only see projects they can create issues in and not the projects in which they cannot.

How to do it...

```
<!-- Uncomment & use this permission to show only projects  
where the user has create permission and issues  
within that where they are the reporter. -->  
  
<!-- This permission type should only ever be assigned to  
the "Browse Projects" permission. -->  
  
<!-- Other permissions can use the "reporter" or "create"  
permission type as appropriate. -->  
  
<!-- <type id="reportercreate" enterprise="true">  
    <class>com.atlassian.jira.security.type  
  
        .CurrentReporterHasCreatePermission</class>  
  
    </type>  
  
-->
```

How to do it...

Grant permission

Permission Administer Projects

Granted to

- Project Role
- Application access
- Group
- Reporter
- Single user
- Project lead
- Current assignee
- Group custom field value
- Reporter (show only projects with create permission)
- Assignee (show only projects with assignable permission)

[Show less](#) new permission types

Grant Cancel

How it works...

- The reportercreate permission type checks whether the current user has permission to create issues in a given project.
- This is different than the default reporter or the current reporter permission type, which make the project visible to all users.

Setting up password policies

- By default, Jira allows you to create a password of any combination and length.
- For security, organizations often need to have password policies such as password length and complexity to strengthen the passwords and make them difficult to guess.

How to do it...

Password Policy

i The password policy is currently disabled.

- Disabled:** Allow all passwords
- Basic:** Reject very weak passwords ?
- Secure:** Require stronger passwords ?
- Custom:** Use your own settings

pre-defined policies

Password Length ?

Minimum Length

Maximum Length

Character Variety ?

Minimum Uppercase

Minimum Lowercase

Minimum Digits

Minimum Special ?

Combination ?

Similarity Checks ?

Old Password

User Information

Update

How it works...

Change Password

Current Password*

New Password*

The new password must satisfy the password policy.

- The password must have at least 10 characters.
- The password must contain at least 1 special character, such as &, %, ™, or É.
- The password must contain at least 3 different kinds of characters, such as uppercase letters, lowercase letters, numeric digits, and punctuation marks.

Confirm Password*

Update Cancel

There's more...

Password Policy Configuration

General

Password Complexity

Additional Options

User Locking

Expiration Notification

Set Password Complexity Requirements

Refer to [documentation](#) for more info.

Enable password characters rules:

Passwords like name allowed:

Number of character characteristics: 3

Rule for determining if a password contains the desired mix of character types. In order to meet the criteria of this rule, passwords must meet any number of supplied below character rules.

Digit character rule: 1

Rule for determining if a password contains the correct number of digit characters. Number 0 disables rule.

Non-alphanumeric characters: 1

Rule for determining if a password contains the correct number of non-alphanumeric characters. Number 0 disables rule.

Uppercase characters: 1

Rule for determining if a password contains the correct number of uppercase characters. Number 0 disables rule.

Lowercase characters: 1

Rule for determining if a password contains the correct number of lowercase characters. Number 0 disables rule.

Capturing electronic signatures for changes

- Organizations that have strict regulatory requirements often need to capture electronic signatures (or e-signatures) as issues move along the workflow, for future auditing purposes. This is often a part of the CFR Part 11 compliance.
- In this recipe, we will look at how to enforce and capture e-signatures when someone tries to transition an issue through the workflow.

How to do it...

Select a Field Type

Search

All

Standard

Advanced

No field preview

Electronic Signature
Captures and validates user electronic signature.

No field preview

Global Rank
Global rank field for Jira Software use only. MANAGED

No field preview

Group Picker (multiple groups)
Choose multiple user groups using a popup picker window.

No field preview

Group Picker (single group)
Choose a user group using a popup picker window.

No field preview

Hidden Job Switch
Hidden switch programmatically set whether or not to create a Perforce job.

 Find more custom fields

Next Cancel

How it works...

Done

Resolution* Done

Assignee* Patrick Li

E-Signature* patrick Username
..... Password

The supplied credentials are incorrect.

capturing and verifying electronic signature

Comment

Style **I** U A ^A _A *O*     

 Visual  Text

Viewable by All Users

Done Cancel

How it works...

Activity

All Comments Work Log History Activity E-Signatures Transitions Source

| Signature | User ID | Timestamp | Old Status | New Status |
|------------|---------|-------------------|-------------|-------------|
| Patrick Li | patrick | 07/Apr/19 6:27 PM | To Do | In Progress |
| John Clark | john | 07/Apr/19 6:27 PM | In Progress | Testing |
| Eric Lin | eric | 07/Apr/19 6:27 PM | Testing | In QA |
| Patrick Li | patrick | 07/Apr/19 6:29 PM | In QA | Done |

Changing the duration of the remember me cookies

- When a user selects the Remember my login on this computer checkbox, the user doesn't need to re-enter their credentials again from the same browser, unless they are explicitly logged out.
- In addition to this, by default, this feature lasts for two weeks.

How to do it...

1. Open the seraph-config.xml file from the JIRA_INSTALL/atlassian-jira/WEB-INF/classes directory in a text editor.
2. Locate the autologin.cookie.age line, and change the value of param-value to the desired number in seconds:

```
<init-param>  
    <param-name>autologin.cookie.age</param-name>  
    <param-value>1209600</param-value>  
</init-param>
```

3. Restart Jira for the changes to apply.

How it works...

- Jira uses the Seraph framework to manage its HTTP session cookies. When the Remember me option is checked, it creates seraph.rememberme.cookie.
- The seraph-config.xml file is used to configure the Seraph framework and the autologin.cookie.age parameter is used to set the maximum age for the cookie.

Changing the default session timeout

- By default, each active user session lasts for five hours (or 300 minutes) of idle time.
- This means that a user can log in and leave the computer for up to five hours and their browser session will still remain active.

How to do it...

1. Open the web.xml file from the JIRA_INSTALL/atlassian-jira/WEB-INF directory in a text editor.
2. Locate the <session-config> line, and change the value of session-timeout to the desired number in minutes:

```
<session-config>  
    <session-timeout>300</session-timeout>  
</session-config>
```

3. Restart Jira for the changes to apply.