

Agile Project Management Using Jira Software



COURSE OVERVIEW



What you will learn

- Use lean and agile principles
- Differentiate kantan and scrum
- Configure Jira to match your team's current process

To succeed here, you need to have

- No knowledge of agile or Jira Is assumed

Live teach schedule (1 of 2)

1	Course Overview	20 minutes
2	Agile and Jira Overview	45 minutes
3	Visualize Work Using Project Boards	35 minutes
4	Enrich Issues	40 minutes
5	Kanban Method	30 minutes
6	Lean and Agile Principles	45 minutes
7	Scrum Overview I - Artifacts	15 minutes
8	Scrum Overview II – Roles and Events	35 minutes

Live teach schedule (2 of 2)

9	Quick Search and Basic Search	25 minutes
10	JQL	20 minutes
11	Filters	15 minutes
12	Epics	20 minutes
13	Dashboards	15 minutes
14	Putting it all Together	15 to 40 minutes*
	Total	~7 hours

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 ones in the Cloud version. The sidebar on the left includes additional options like 'Backlog' and 'Reports'.

Server /
Data Center

Jira Cloud- classic vs. next-gen projects

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

Classic project: A card with a blue folder icon and the text "Create project". It contains two sections: "Classic project" (blue background) and "Try a next-gen project" (white background). The "Classic project" section describes it as having "All the power and functionality you expect. Created and managed by your Jira admin." The "Try a next-gen project" section describes it as having "Easy setup and reimagined features. Created and managed by project team members."

Kanban board: A screenshot of the Jira Cloud classic Kanban board interface. The sidebar shows "Issues and filters" and "Project settings". The board has columns: "READY FOR TEST", "READY FOR DEVELOPMENT", "IN PROGRESS", and "DONE". Items include "add item 1" and "add item 2" under "READY FOR TEST", and "add item 3" under "IN PROGRESS".

PROJ board: A screenshot of the Jira Cloud next-gen PROJ board interface. The sidebar shows "Issues and filters" and "Project settings". The board has columns: "READY", "IN PROGRESS", "BLOCKED", and "DONE". Items include "add feature 1" and "add feature 2" under "READY", and "database refactor" under "BLOCKED".

Lab 1- Course Overview

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

Agile and Jira Overview



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

Topics

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



What is Agile?

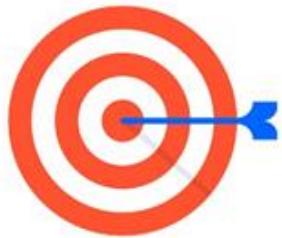
A way of getting things done.

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)



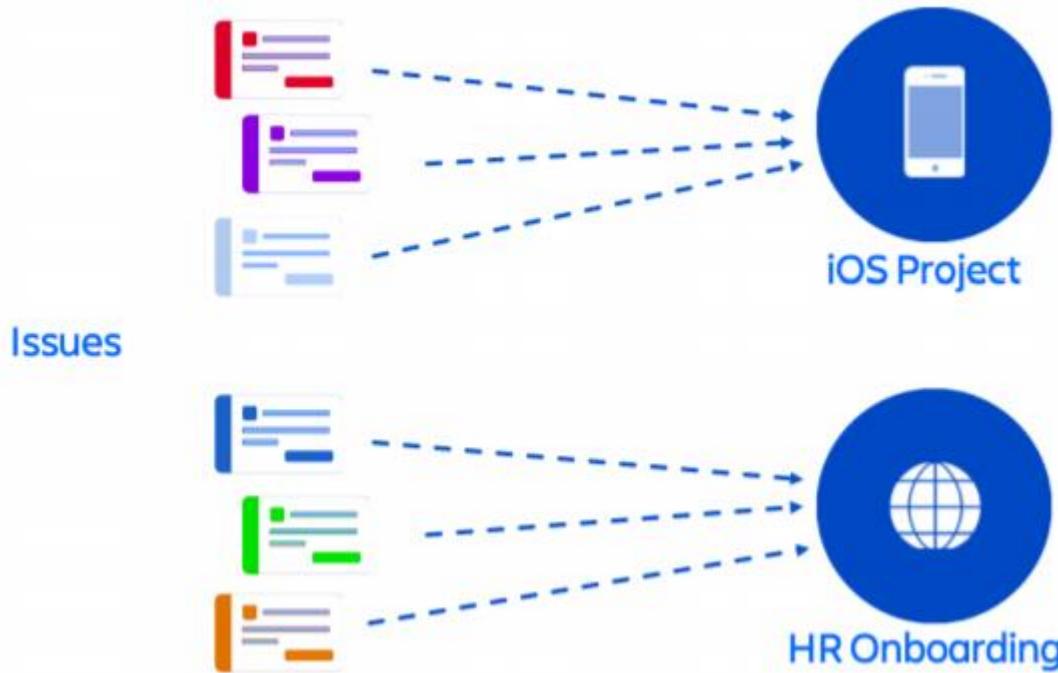
Issue Key



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

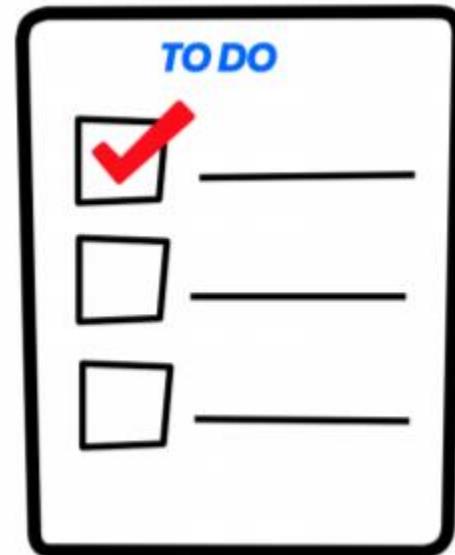
Topics

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



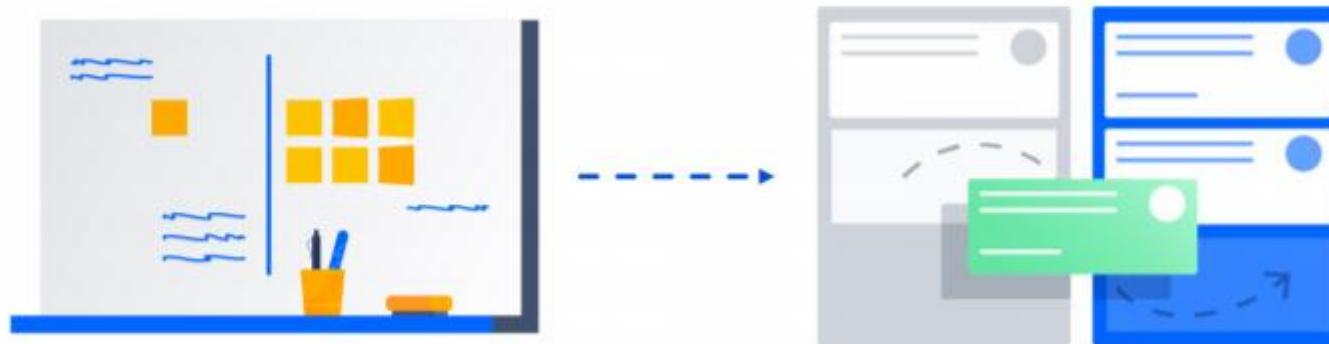
Virtualizing work: a “to do” list

- 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



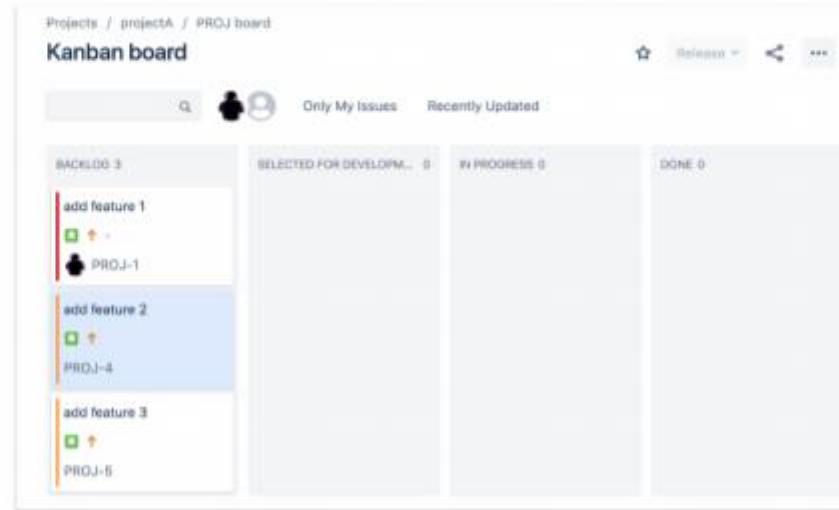
Topics

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

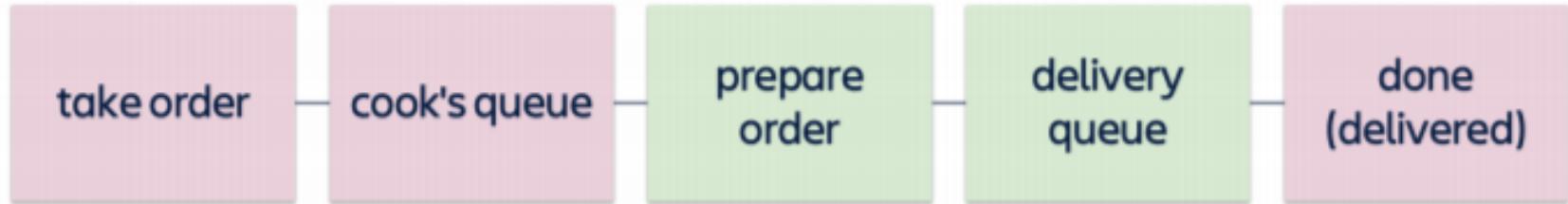


Workflows

- 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)

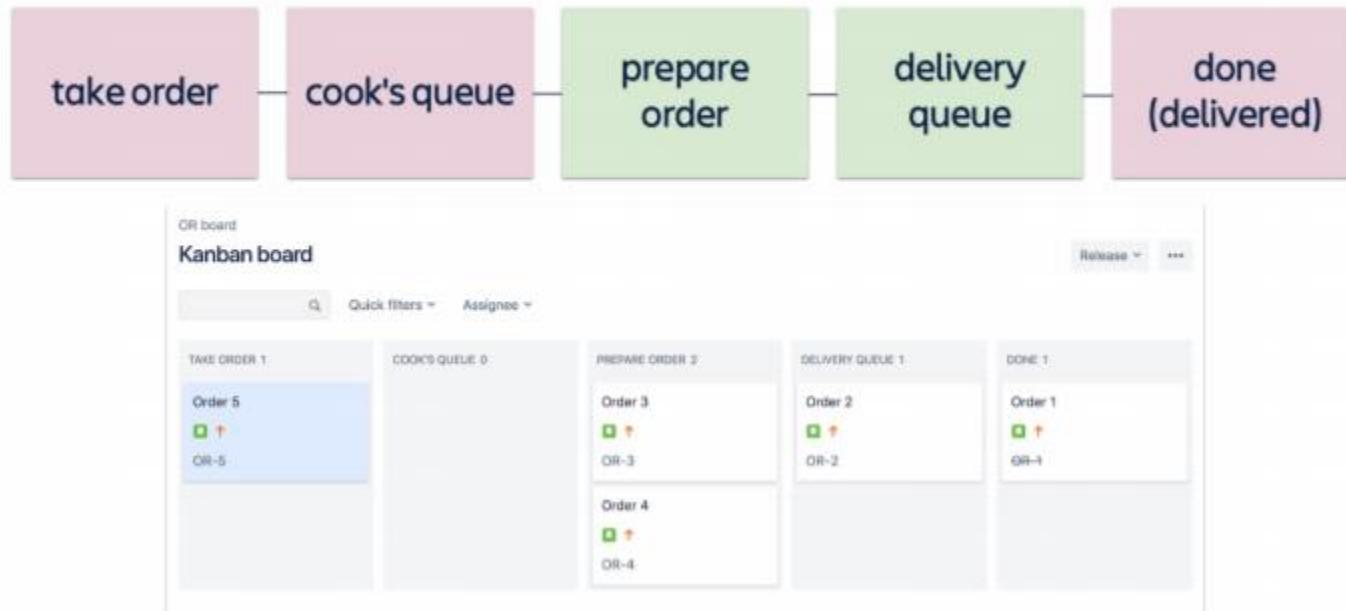


Example workflow: restaurant order and delivery



Boards VS Workflows

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



Topics

- Virtualizing 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 to the 'Create board' option in the context menu, which is open next to the 'Release' dropdown.

Projects / projectA / PROJ board

Kanban board

Only My Issues Recently Updated

BACKLOG 3 SELECTED FOR DEVELOPME... 0 IN PROGRESS 0

add item 1
PROJ-1

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

Attach Create subtask Link issue ...

Description

Add a description...

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

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



Adding a board column

Projects / projectA / PROJ board

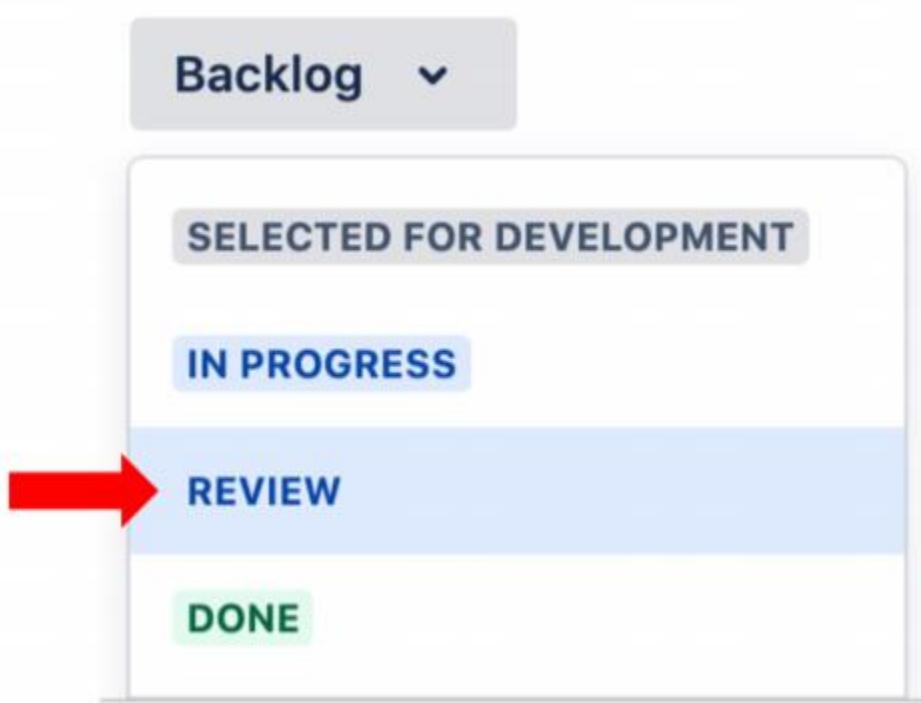
Kanban board

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

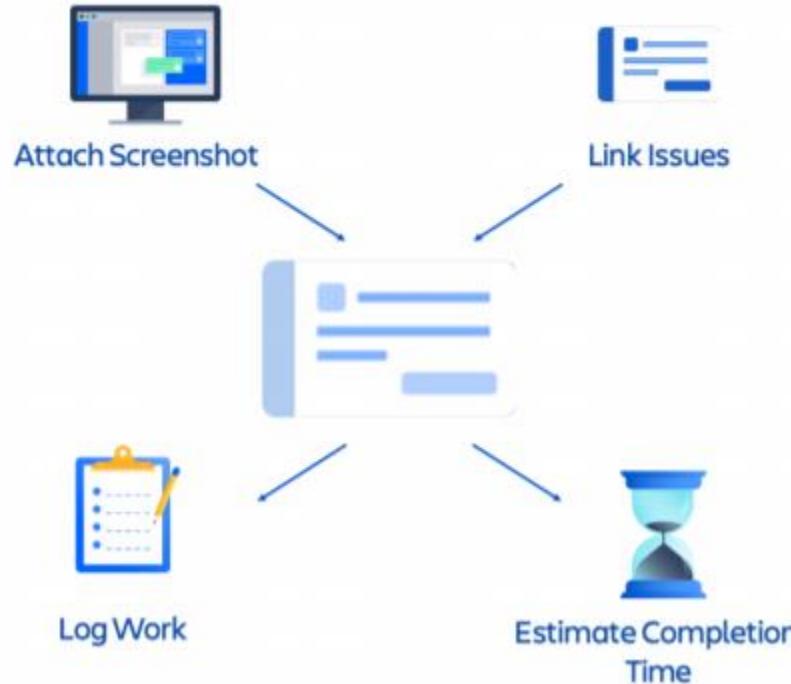
Reporter:



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)

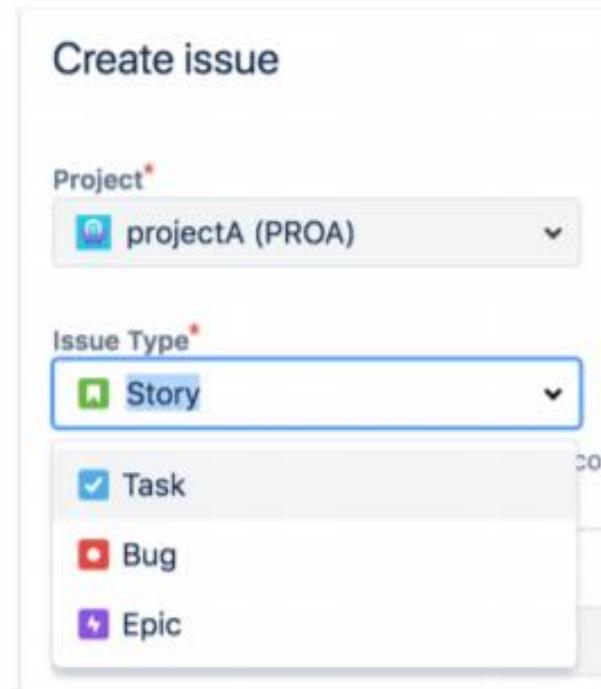
Issue Type*

Story

Task

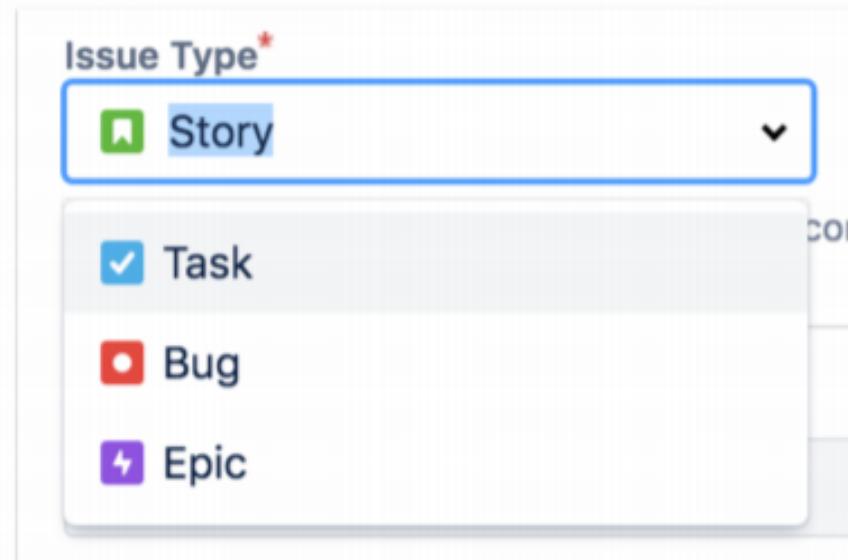
Bug

Epic

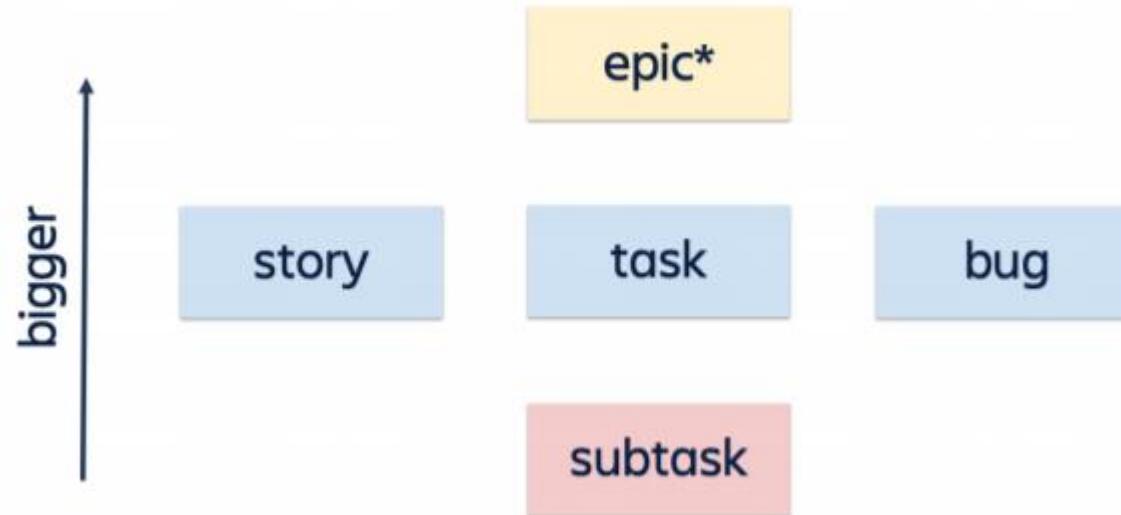


Why issue types

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



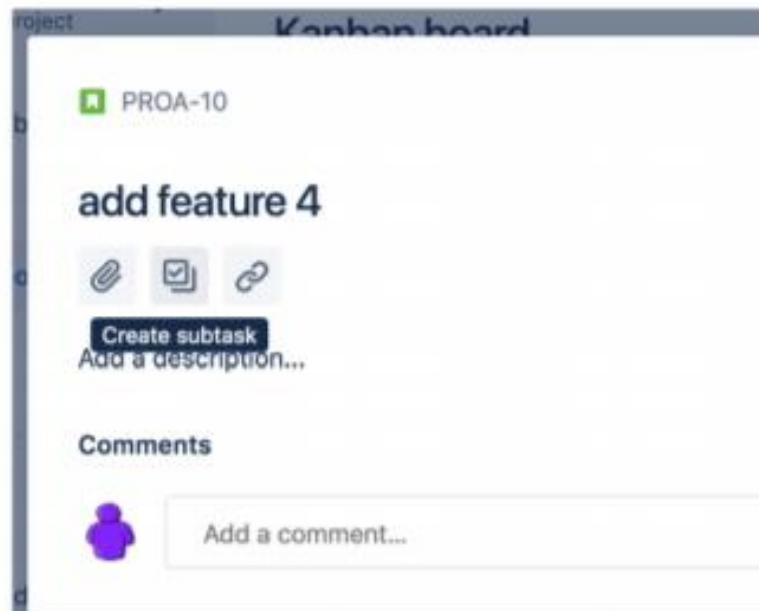
Jira's issue type hierarchy



* 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 includes fields for 'Status' (Selected for Development), 'Assignee' (Unassigned), 'Labels' (refactor, database), 'Priority' (Medium), 'Reporter' (Steve Byrnes), and creation/updating details. Two labels, 'refactor' and 'database', are highlighted with blue boxes and arrows pointing from the text 'Labels' on the right.

PROJ-6

complete task 1

Add a description...

Subtasks

PROJ-10: complete task 1a BACKLOG

Comments

Add a comment...

Status
Selected for Development

Assignee
Unassigned

Labels
refactor database

Priority
Medium

Reporter
Steve Byrnes

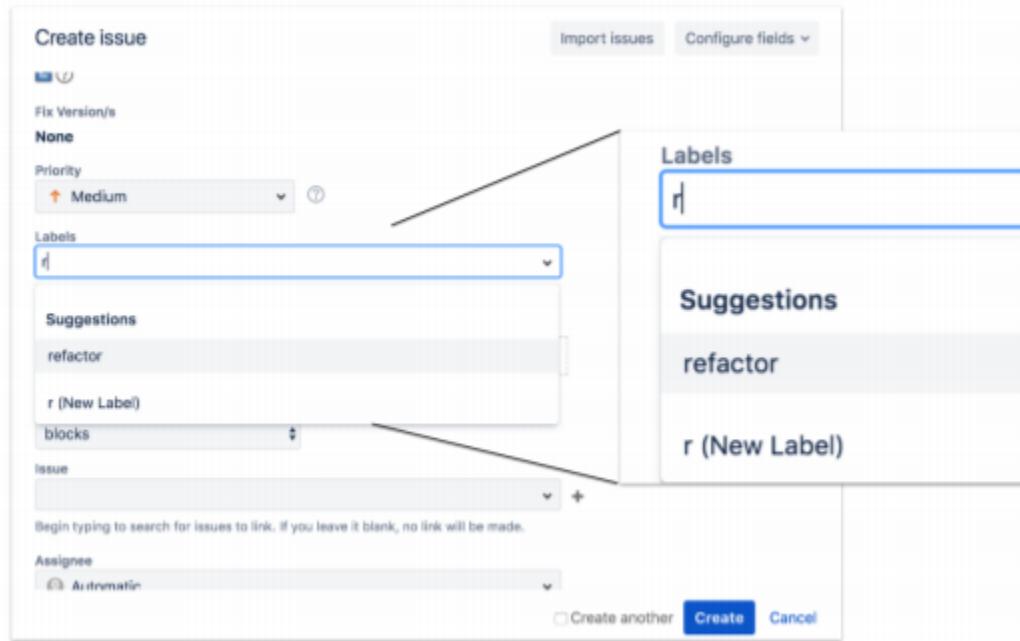
Show more

Created 6 days ago
Updated 1 hour ago

Labels

refactor database

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 Method



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

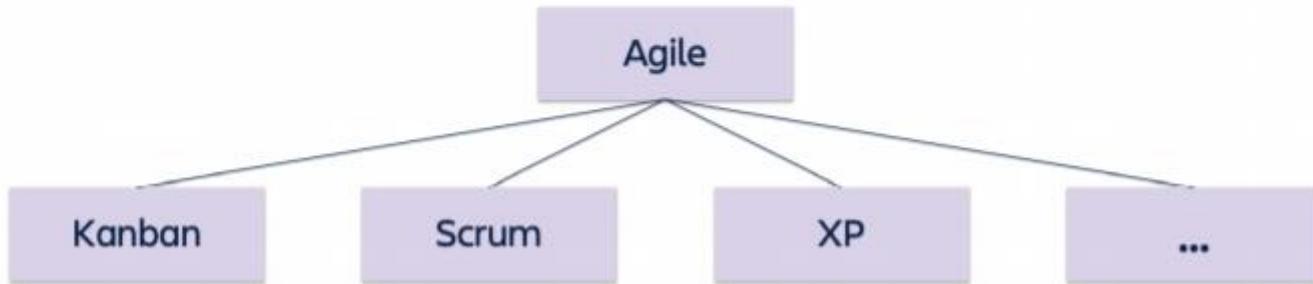
Topics

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



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/ smolt batch size
 - Pull work rather than push (where It makes sense)
 - Continuously prioritize work Items

Why choose the kanban method

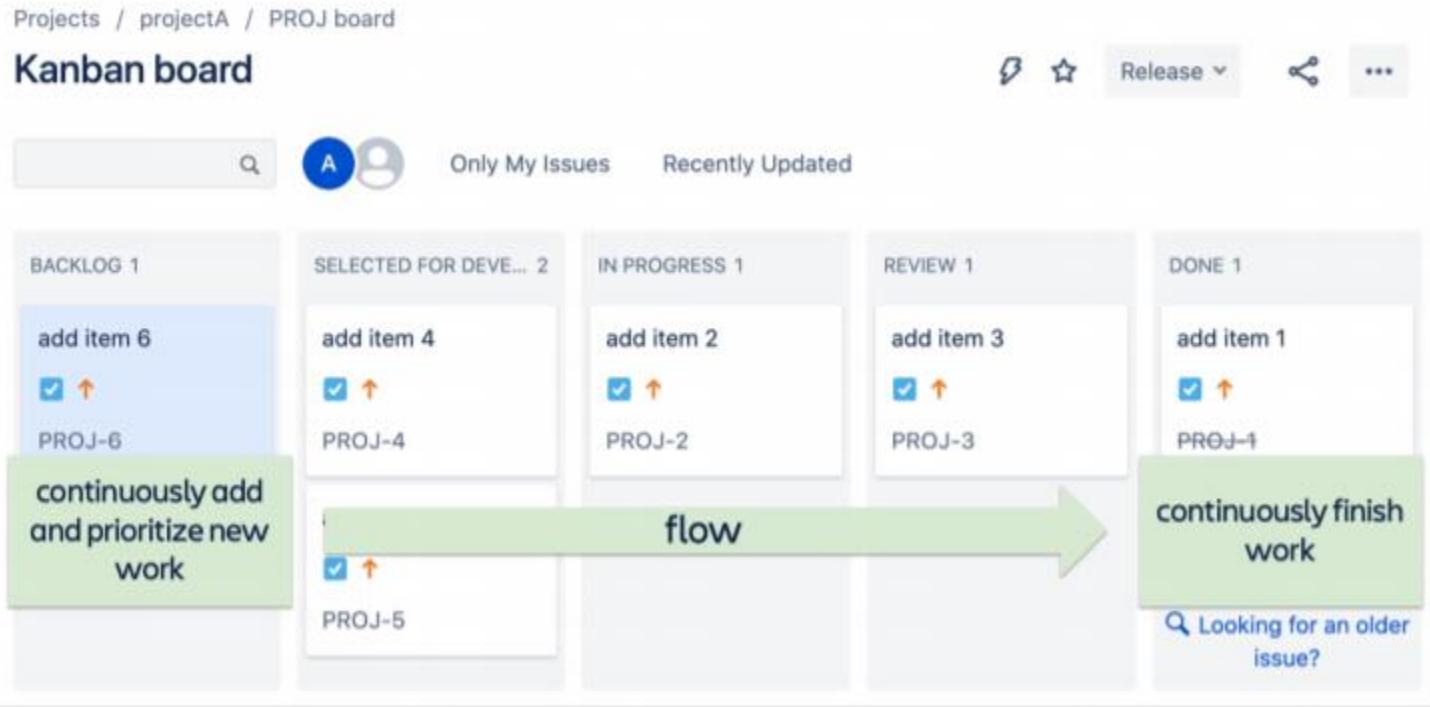
- Very Lightweight and efficient
- Evolutionary approach of transforming to agile
- Works well If the workflow is service-oriented
- operations
- support
- maintenance development
- new hire funnel
- Supports multi-team and multi-project workflows

Topics

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



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

Only My Issues Recently Updated

BACKLOG 1 SELECTED FOR DEVEL... 2 IN PROGRESS 0 REVIEW 3 Max 2 DONE 0

add item 6 add item 4 add item 1
PROJ-6 PROJ-4 PROJ-1

add item 5 add item 2
PROJ-5 PROJ-2

add item 3 add item 3
PROJ-3 PROJ-3

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

A red arrow points down to the REVIEW 3 column.

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

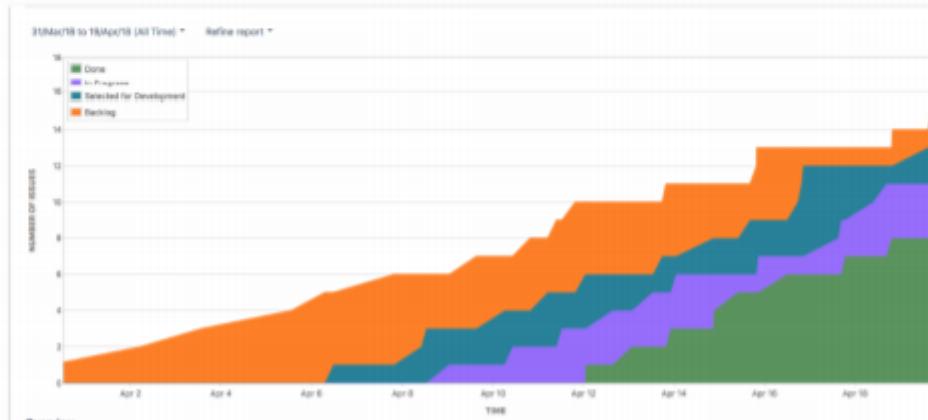
Kanban board

Only My Issues Recently Updated

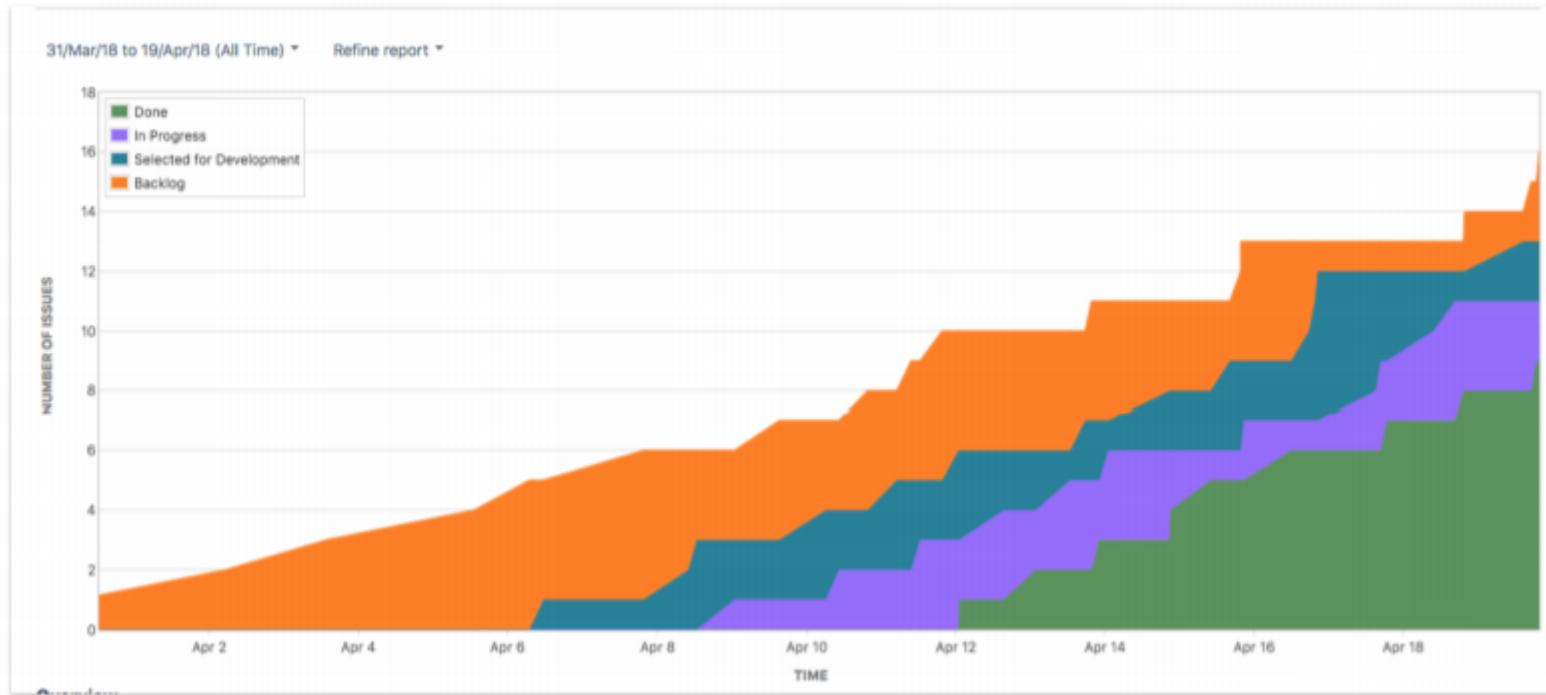
BACKLOG 1	SELECTED FOR DEVELOP... 2 Min 2	IN PROGRESS 1	REVIEW 2 Max 2	DONE 6
add item 6 PROJ-6	add item 4 PROJ-4	add item 3 PROJ-3	add item 1 PROJ-1	We're only showing recently modified issues. Looking for an older issue?
	add item 5 PROJ-5		add item 2 PROJ-2	

What agile reports

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



Cumulative flow



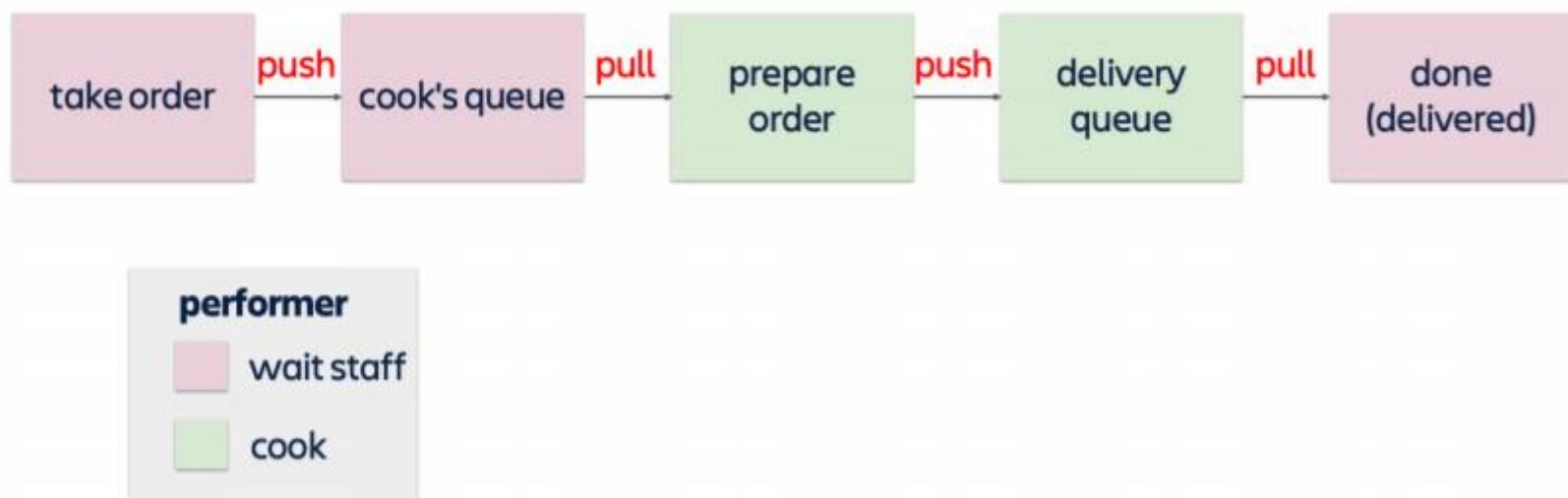
Topics

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



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

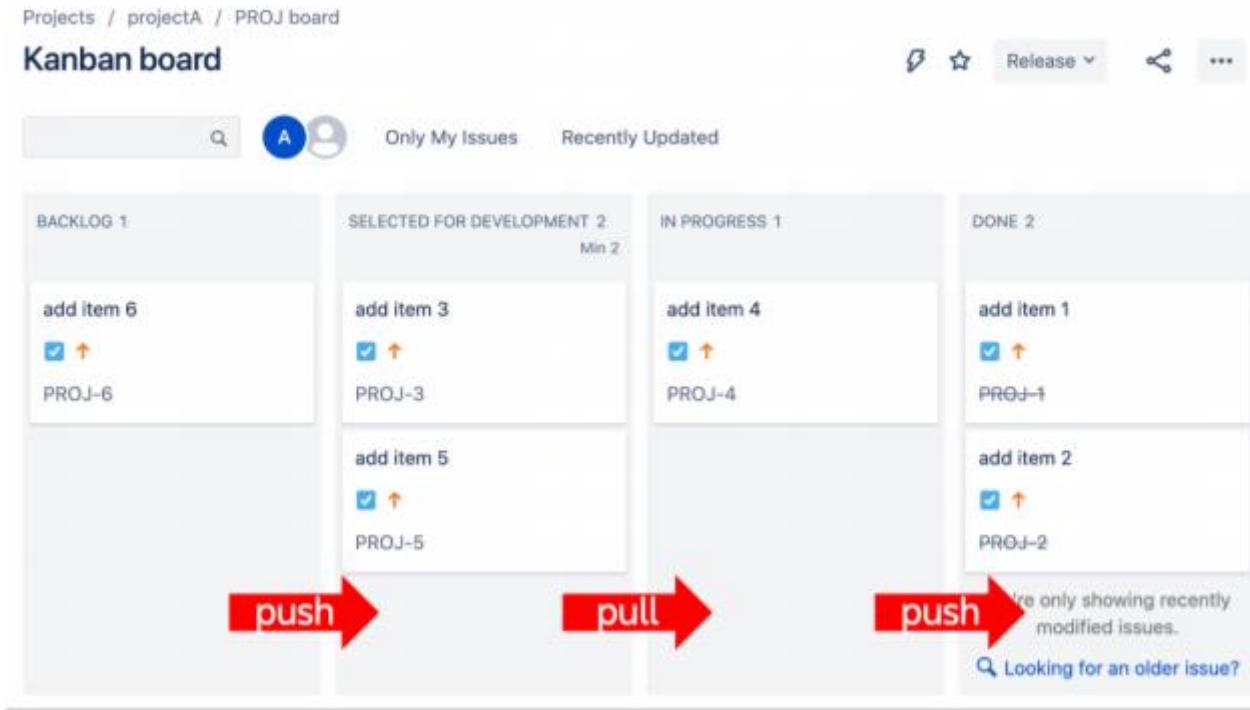
Only My Issues Recently Updated

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

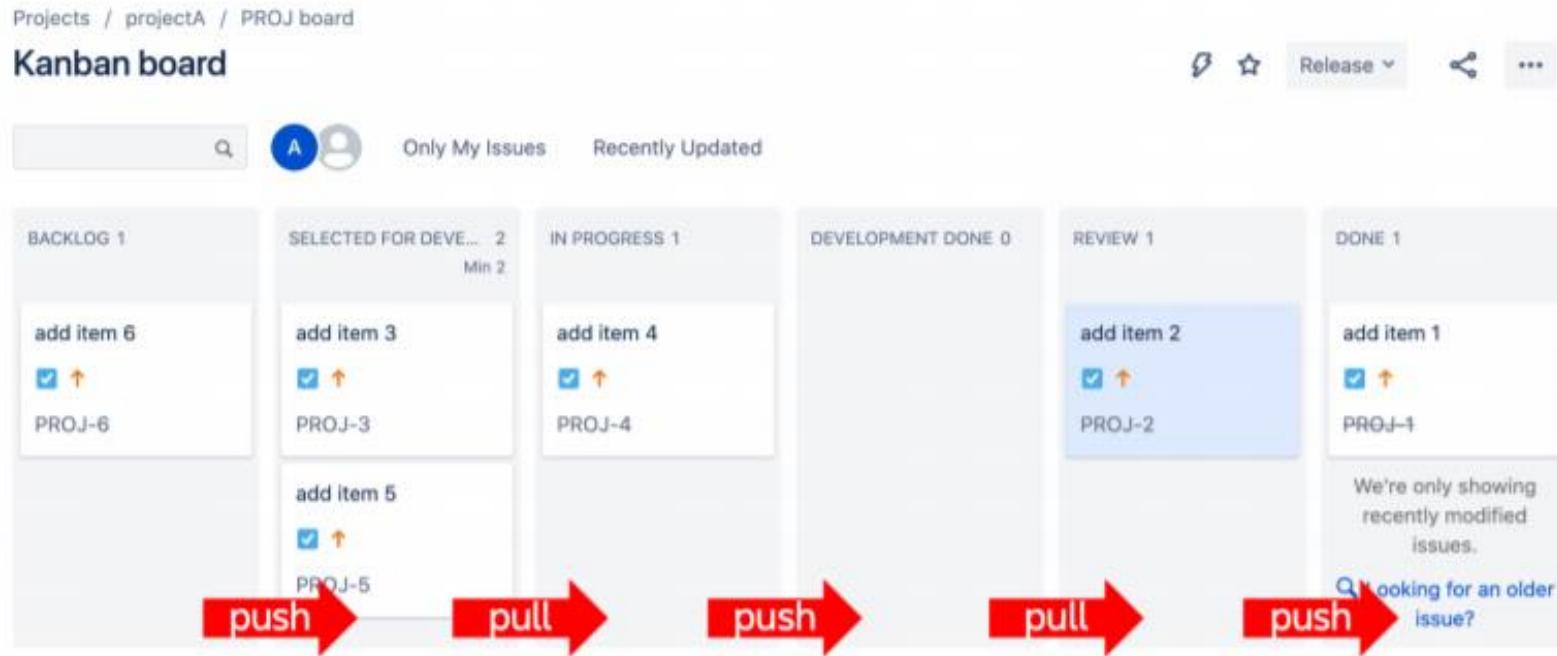
Release    

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 <input checked="" type="checkbox"/> PROJ-6	add item 3 <input checked="" type="checkbox"/> PROJ-3	add item 4 <input checked="" type="checkbox"/> PROJ-4	add item 1 <input checked="" type="checkbox"/> PROJ-1
	add item 5 <input checked="" type="checkbox"/> PROJ-5		add item 2 <input checked="" type="checkbox"/> 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 Backflow



Separate Backflow

Projects / projectA / PROJ board

Kanban board

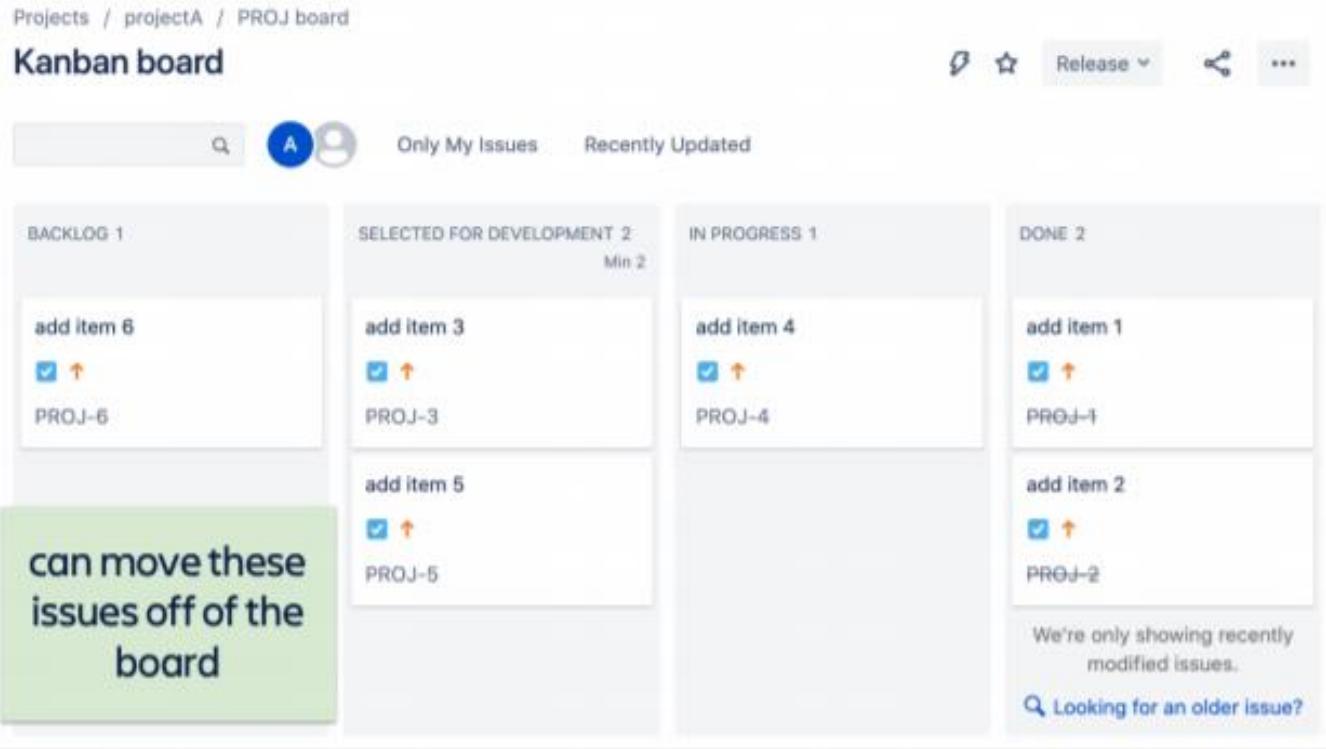
Only My Issues Recently Updated

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

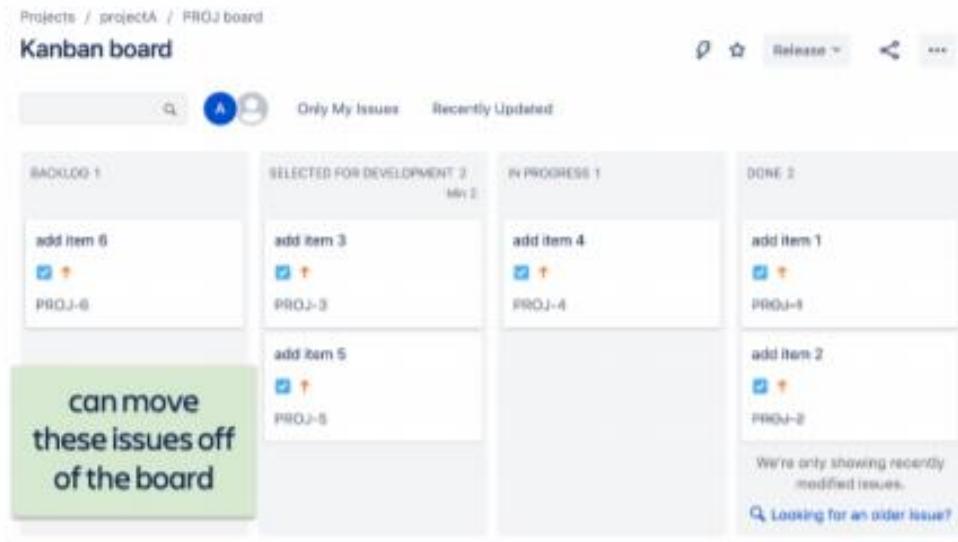
Only My Issues Recently Updated

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. 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, there is a '+ Create issue' button.

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

Lean and Agile Principals



What will you learn?

- Identify reasons the Toyota Production System Is studied today
- identify kanban objects
- Describe benefits of using kanban objects
- Identify kanban systems
- Describe Lean principles
- Describe agile principles
- Compare Lean and agile principles

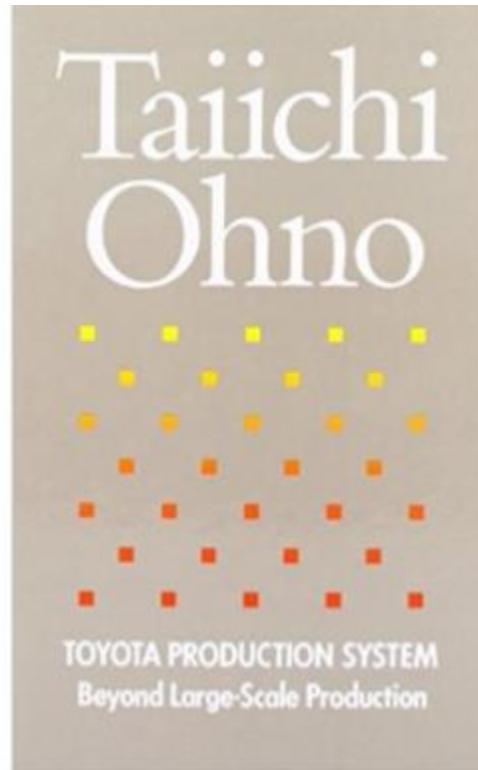
Topics

- Toyota production system
- Toyota kanban
- Lean principles
- Agile manifesto
- Lean vs Agile



Toyota production system

- Written in 1978, English translation in 1988
- Describes what is now sometimes called "lean thinking" or "lean management"
- Many agile principles are similar



Toyota simplified history

- "Catch up with America in three years."
- Focus was to eliminate waste and increase productivity
- Embraced ideas from Ford, but used a more "agile" approach

"I would like to emphasize that (the Toyota Production System) was realized because there were always clear purposes and needs."

Taichi Ohno

Toyota Production System: Beyond Large-Scale Production

Lean principles apply broadly

"I am confident (the Toyota Production System) will reveal its strength as a management system..."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production
preface to the English edition (1988)

Topics

- Toyota production system
- **Toyota kanban**
- Lean principles
- Agile manifesto
- Lean vs Agile



What is kanban?

- Kanban - an object that controls the flow of work
- The idea came to Toyota from supermarkets
 - Instead of push, order when inventory is low (pull)
 - This matches the supply and demand
 - An empty box is a "kanban" - a signal to order more

"From the supermarket, we got the idea of viewing the earlier process in a production line as a kind of store."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Other examples of kanbans



guest check



coffee cup



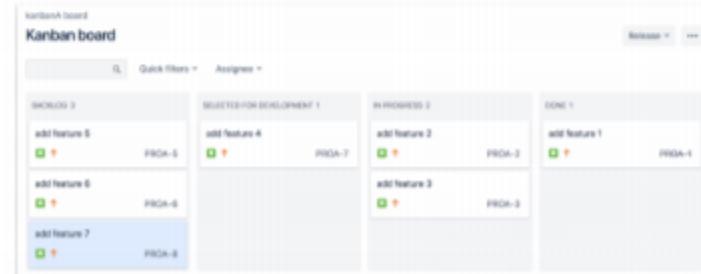
Jira issue

Kanban systems

"The Toyota Production System is the production method and the kanban system is the way it is managed."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production



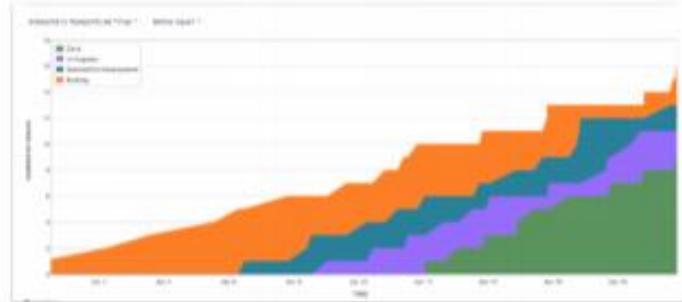
Benefits of kanban system

- Visualizes work
- Simple
- Reliable
- Efficient
- Eliminates waste
- Identifies bottlenecks/easy to improve



Summary – kanban defintions

- Kanban token - An object that controls the flow of work
- Kanban system - A system that controls the flow of work using kanbans
- Kanban method - A lightweight agile method



Topics

- Toyota production system
- Toyota kanban
- **Lean principles**
- Agile manifesto
- Lean vs Agile



Why principles?

"With a better tool, we can get wonderful results. But if we use it incorrectly, the tool can make things worse."

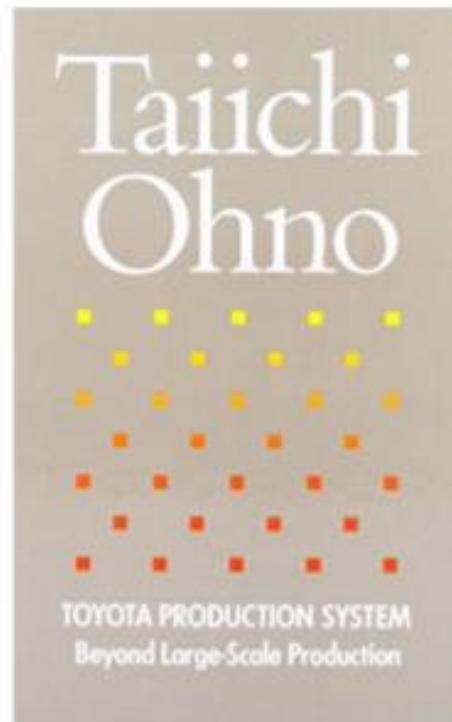
"We should not forget to always use the principles..."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Lean principles

1. Empower the team
2. Visualize work
3. Embrace the scientific method
4. Improve the "flow" of value
5. Build quality in



Empower the team

"...Operators acquire a **broad spectrum** of production skills... and **participate in** building up a total system in the production plant. In this way, the individual can **find value in working.**"

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Empower the team – teamwork

"A championship team combines good teamwork with individual skill."

"In modern industry, harmony among people in a group, as in teamwork, is in greater demand than the art of the individual craftsman."

Taichi Ohno

Toyota Production System: Beyond Large-Scale Production

Visualize work

- Visual control- Toyota uses kanbans to signal and control the work
- Andon board- An information board that shows any existing problems

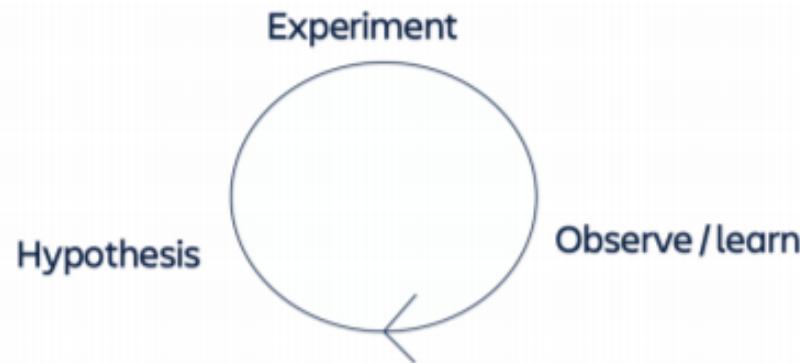
"When one looks up, the andon (the line stop indication board) comes into view, showing the location and nature of trouble situations at a glance."

Taichi Ohno

Toyota Production System: Beyond Large-Scale Production

Embrace the scientific method

1. Create a hypothesis
2. Build an experiment
3. Observe/learn from the results
4. Repeat/iterate



Embrace the scientific method

"The Toyota Production System has been built on the practice and evolution of this scientific approach."

"Progress can not be generated when we are satisfied with existing situations."

"... the new market demanded a constantly improving automobile."

Taiichi Ohno
Toyota Production System: Beyond Large-Scale Production

Embrace the scientific method-embrace change

"As long as we can not accurately predict the future,
our actions should change to suit changing situations.
In industry, it is important to enable people to cope
with change and think flexibly."

"Build a fine-tuning mechanism into the business so
that change will not be felt as change."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Improve the flow- limit work in progress / small batch size

"Reducing the number of kanban increases their sensitivity."

"People prefer working with large quantities. It is easier than having to work hard and learn from producing small quantities."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Improve the flow - map the value stream

- Draw the current state
- Draw the desired future state
- Iterate toward the future state

Improve the flow – pull work

"The conventional way was to supply materials from an earlier process to a later process. So, I tried thinking about the transfer of materials in the reverse direction."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Improve the flow – eliminate waste

"The basis of the Toyota Production System is the absolute elimination of waste."

"The vicious cycle of waste generating waste hides everywhere..."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Improve the flow – reduce setup times

"Our production slogan is 'small lot sizes and quick setups'."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Improve the flow – automate what should be automated

"With computers available, it is a waste to perform calculations by hand."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Summary – ways to improve the flow of value

1. Limit work in progress / small batch size
2. Map the value stream
3. Pull work
4. Eliminate waste
5. Reduce setup times
6. Automate what should be automated

"Look straight at the reality."

Taichi Ohno

Toyota Production System: Beyond Large-Scale Production

Build quality in

"...produce quality products 100 percent of the time..."

Taiichi Ohno

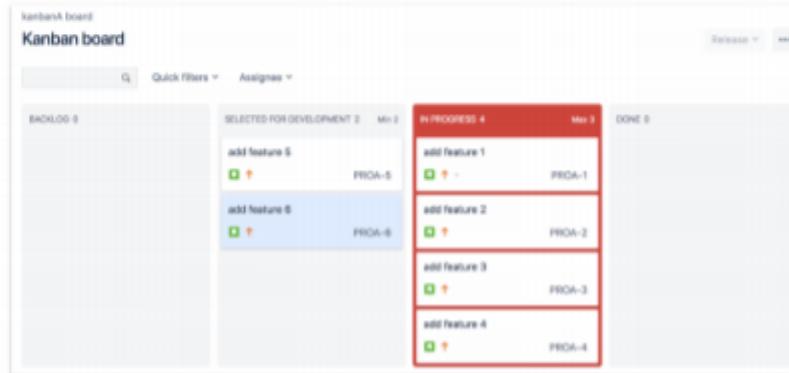
Toyota Production System: Beyond Large-Scale Production

Build quality in-the process should identify problems

"...distinctions between normal and abnormal operations must be clear and countermeasures (solutions) always taken to prevent recurrence."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production



Build quality in-fix problems when they are discovered

"Correct a mistake immediately-to rush and not to take time to correct a problem causes work loss later."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Example-fix problems when they are discovered



“Because a device that could distinguish between normal and abnormal conditions was built into the machine, defective products were not produced.”

Taiichi Ohno
Toyota Production System: Beyond Large-Scale Production

Build quality in- identify and fix the root cause

"By asking why five times and answering it each time, we can get to the real cause of the problem, which is often hidden behind more obvious solutions."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

Example: asking “why” to find the root cause and solution

Problem: The orbiter crashed into the planet.

Question: Why did the orbiter crash into the planet?

Answer: Because it didn't have the proper trajectory on approach.

Question: Why didn't it have the proper trajectory on approach?

Answer: Because the thrusters did not work properly.

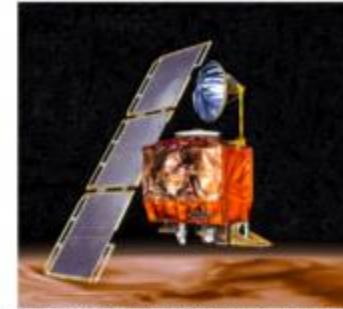
Question: Why didn't the thrusters work properly?

Answer: Because the acceleration data in the software was inaccurate.

Question: Why was the acceleration data inaccurate?

Answer: Because one team used metric units and the other used English units.

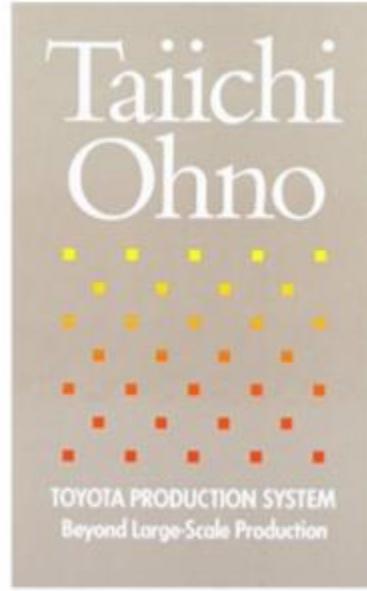
Solution: Standardize on a single system of measurement. Ensure pre-launch tests fail in similar circumstances.



<https://commons.wikimedia.org/w/index.php?curid=3>

Lean principles

1. Empower the team
2. Visualize work
3. Embrace the scientific method
 - a. Continuously learn and improve
 - b. Embrace change
4. Improve the "flow" of value
 - a. Limit work in progress / small batch size
 - b. Map the value stream
 - c. Pull work
 - d. Eliminate waste
 - e. Reduce setup times
 - f. Automate what should be automated
5. Build quality in
 - a. The process should identify problems
 - b. Fix problems when they are discovered
 - c. Identify and fix the root cause



Topics

- Toyota production system
- Toyota kanban
- Lean principles
- **Agile manifesto**
- Lean vs Agile



Manifesto for Agile software development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

© 2001, the above authors

this declaration may be freely copied in any form,
but only in its entirety through this notice.

<https://agilemanifesto.org>

Agile manifesto values statements-takeaway principles

1. Empower the team
 - "Individuals and interactions over processes and tools"
2. Embrace change
 - "Responding to change over following a plan"
3. Partner with the customer
 - "Customer collaboration over contract negotiation"
4. Plan, develop and deliver incrementally
 - "Working software over comprehensive documentation"

Principle behind the Agile Manefesto

Principles behind the Agile Manifesto

We follow these principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.

Principle and ideas from the Agile Manifesto

- 1. Empower the team**
 - a. Select motivated individuals
 - b. Teams should self-organize
 - c. Collaborate to create shared understanding
- 2. Embrace change**
 - a. Partner with the customer
 - b. Obtain fast feedback
 - c. Continuously inspect and adapt
- 3. Plan, develop and deliver incrementally**
 - a. Prefer conversations for conveying information
 - b. Continuously refactor to maintain agility
 - c. Maintain a sustainable pace
 - d. Completed work items are the primary measure of progress
- 4. Focus on value**
 - a. Eliminate waste
 - b. Continuously strive for simplicity
 - c. Don't compromise on quality

Topics

- Toyota production system
- Toyota kanban
- Lean principles
- Agile manifesto
- **Lean vs Agile**



Lean vs Agile

- **Lean**
 - Used at MIT by John Krafcik (1988)
 - Described the ideas of the Toyota Production System
 - Applies to any type of project
- **Agile**
 - Used by the participants who created the Agile Manifesto (2001)
 - Described a lightweight alternative to waterfall software development
 - Applies to any type of project
- The terms are often used interchangeably
- Can think of lean as more “foundational” than agile – you can be lean and use waterfall project management

Lean and Agile principles

Lean

1. Empower the team
2. Visualize work
3. Embrace the scientific method
 - a. Continuously learn and improve
 - b. Embrace change
4. Improve the "flow" of value
 - a. Limit work in progress / small batch size
 - b. Map the value stream
 - c. Pull work
 - d. Eliminate waste
 - e. Reduce setup times
 - f. Automate what should be automated
5. Build quality in
 - a. The process should identify problems
 - b. Fix problems when they are discovered
 - c. Identify and fix the root cause

Agile

1. Empower the team
 - a. Select motivated individuals
 - b. Teams should self-organize
 - c. Collaborate to create shared understanding
2. Embrace change
 - a. Partner with the customer
 - b. Obtain fast feedback
 - c. Continuously inspect and adapt
3. Plan, develop and deliver incrementally
 - a. Prefer conversations for conveying information
 - b. Continuously refactor to maintain agility
 - c. Maintain a sustainable pace
 - d. Completed work items are the primary measure of progress
4. Focus on value
 - a. Eliminate waste
 - b. Continuously strive for simplicity
 - c. Don't compromise on quality

Combined Lean and Agile principles

- 1. Empower the team**
 - a. Select motivated individuals
 - b. Teams should self-organize
 - c. Collaborate to create shared understanding
- 2. Visualize work**
- 3. Experiment using the scientific method**
 - a. Continuously learn and improve
 - b. Embrace change
 - c. Partner with the customer
 - d. Continuously inspect and adapt
- 4. Plan, develop and deliver incrementally**
 - a. Prefer conversations for conveying information
 - b. Continuously refactor to maintain agility
 - c. Maintain a sustainable pace
 - d. Completed work items are the primary measure of progress
 - e. Obtain fast feedback
- 5. Improve the "flow" of value**
 - a. Limit work in progress / small batch size
 - b. Map the value stream
 - c. Pull work
 - d. Eliminate waste
 - e. Reduce setup times
 - f. Automate what should be automated
 - g. Continuously strive for simplicity
- 6. Build quality in**
 - a. Don't compromise on quality
 - b. The process should identify problems
 - c. Fix problems when they are discovered
 - d. Identify and fix the root cause

Scrum Overview 1- Artifacts



What will you learn?

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

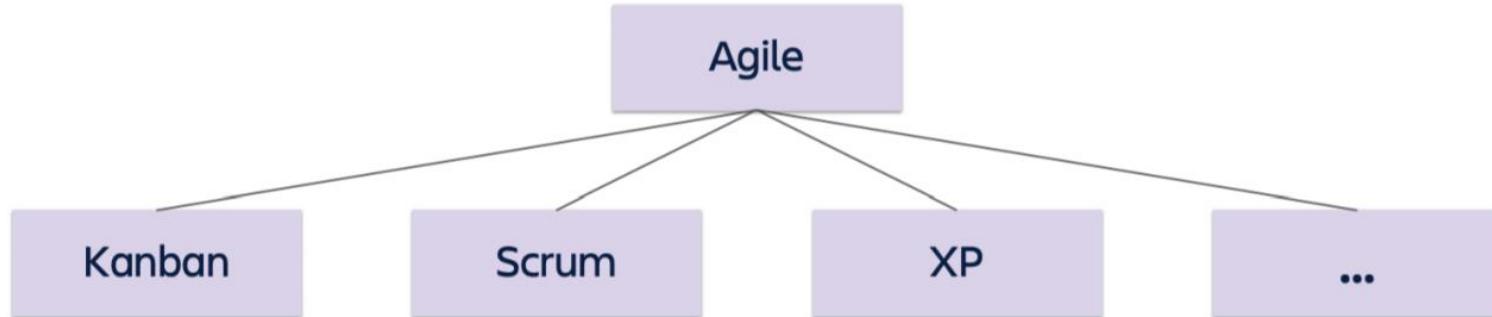
Topics

- What is scrum?
- Scrum Artifacts

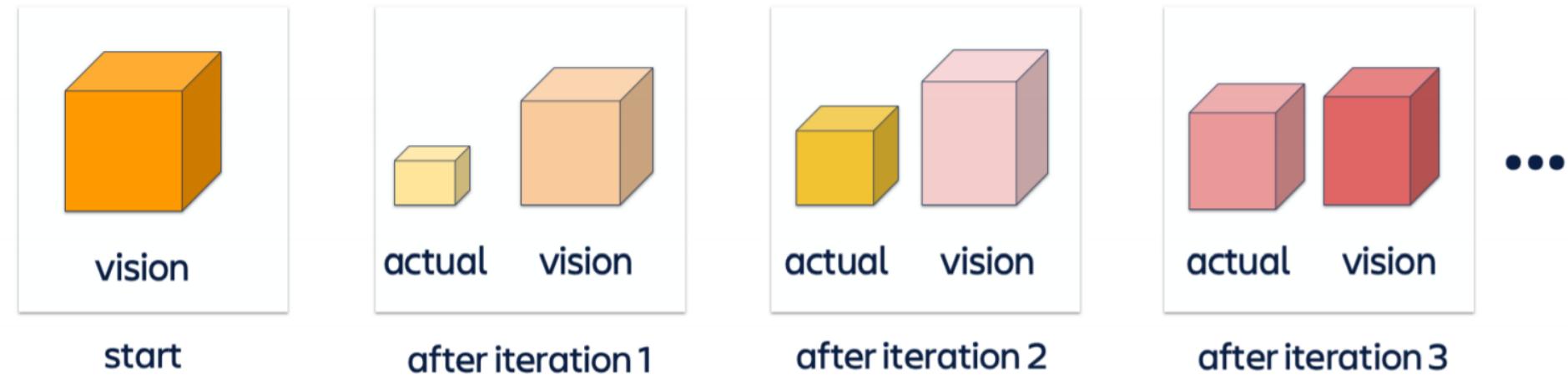


What is scrum?

- "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

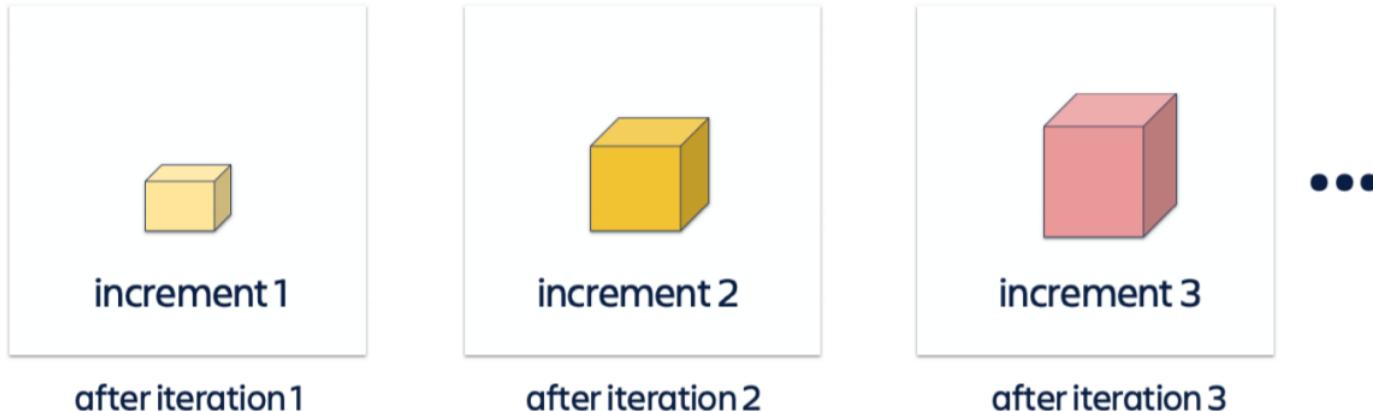


Continuous learning



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)



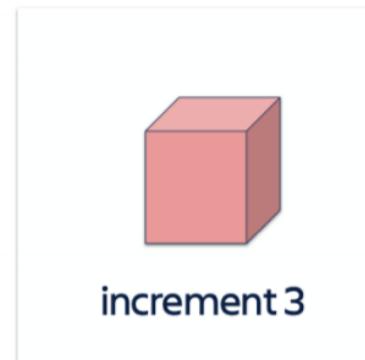
increment 1

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 icon and a label: '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	ISSUES	SPRINTS
	add item 1	PRJ-1 ↑ -
	add item 2	PRJ-2 ↑ -
EPICS	add item 3	PRJ-3 ↑ -

+ Create issue

Creating a sprint with Jira

Projects / projectB / PRJ board

Backlog

Share ...

A Only My Issues Recently Updated

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

Create sprint ...

Backlog 3 issues

Issue	Project	Priority
add item 1	PRJ-1	↑ -
add item 2	PRJ-2	↑ -
add item 3	PRJ-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

add item 1	PRJ-1 ↑ 1
add item 2	PRJ-2 ↑ 2

+ Create issue

2 issues Estimate 3

Backlog 1 issue

Create sprint ...

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 two sprints: 'PRJ Sprint 1' containing '2 issues' (estimated at 1 and 2 story points) and 'PRJ Sprint 2' containing '1 issue' (estimated at 3 story points). Below the sprints, there is a 'Backlog' section with one item ('add item 3') estimated at 3 story points. On the right side of the board, there is a sidebar titled 'Story Points' with a value of '1' displayed.

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

Projects / projectB / PRJ board

PRJ Sprint 1

Create the first product increment.

Only My Issues Recently Updated

TO DO IN PROGRESS DONE

Item	Status
add item 1	PRJ-1
add item 2	PRJ-2

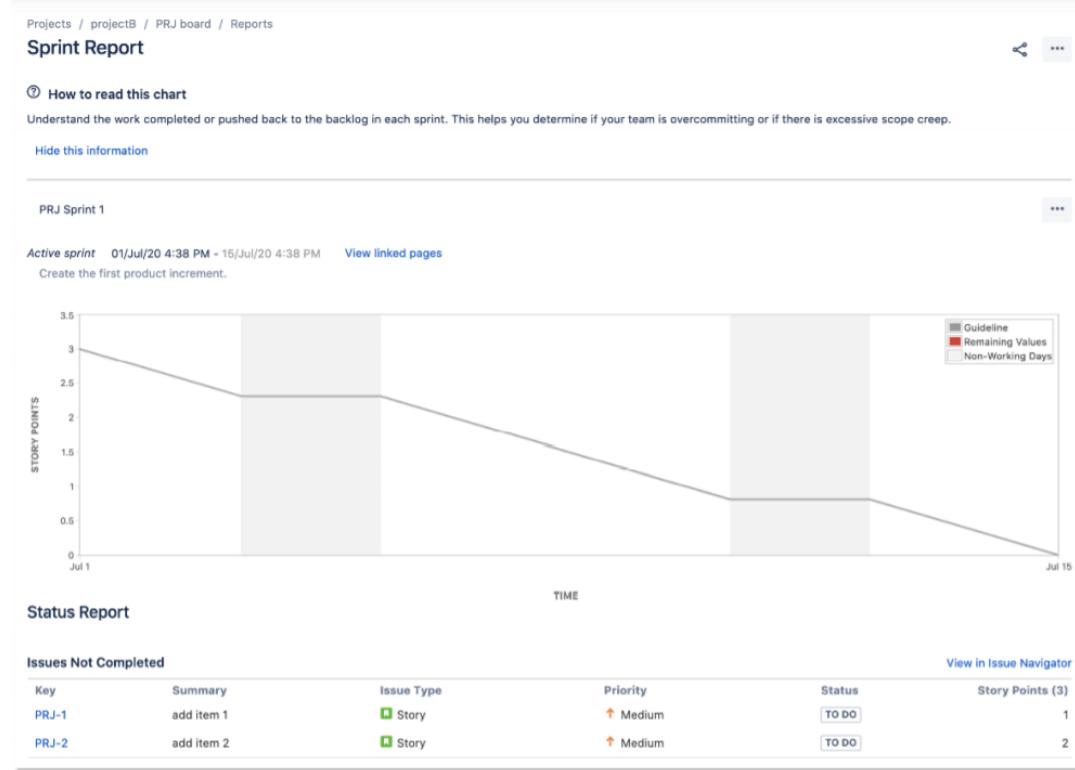
A sidebar on the left lists project management options:

- projectB (Classic software project)
- PRJ board (Board)
- Backlog
- Active sprints** (selected)
- Reports
- Releases
- Issues and filters
- Pages
- Components
- Add item
- Project settings

Scrum Reports – Burndown chart



Scrum Reports – Sprint Report



Velocity

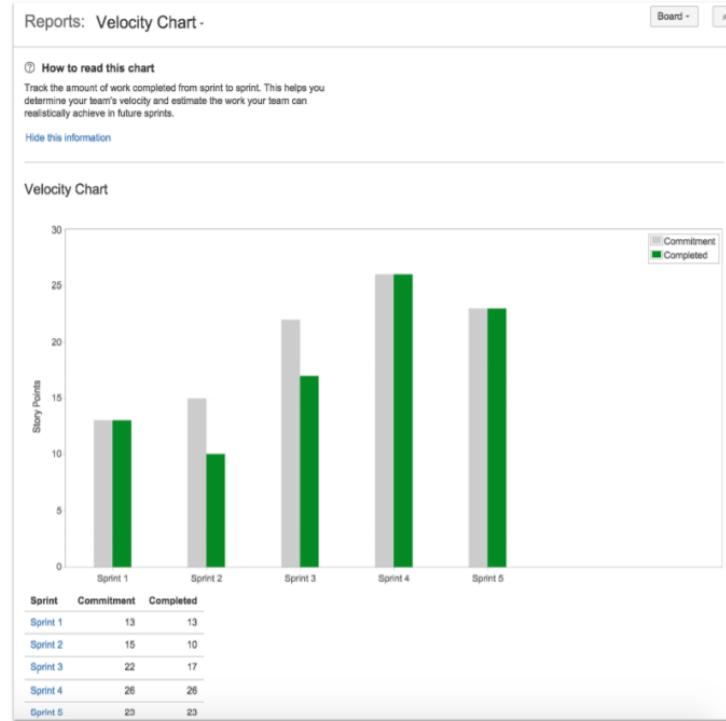
Represents the rate at which the team accomplishes work

- 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 11- Roles and Events



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



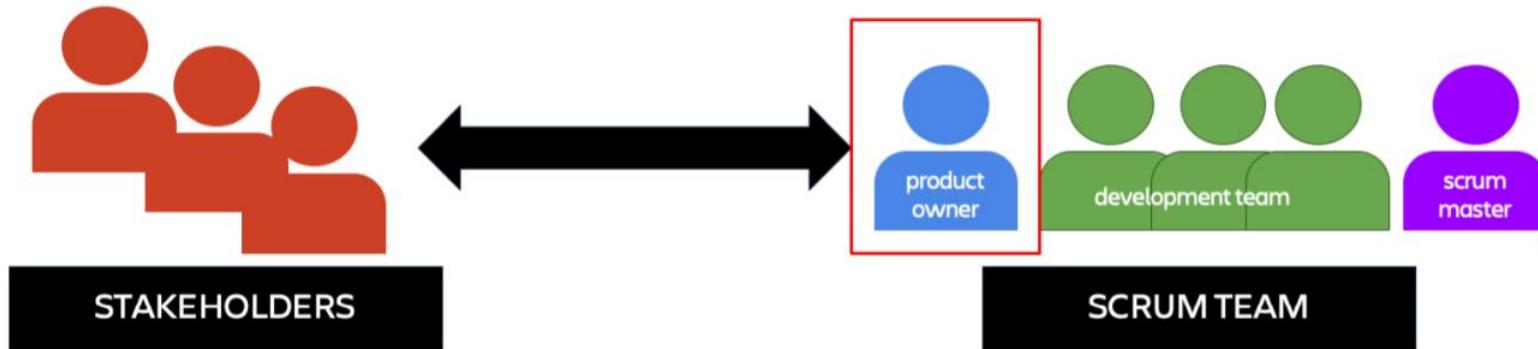
Stakeholders

- 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



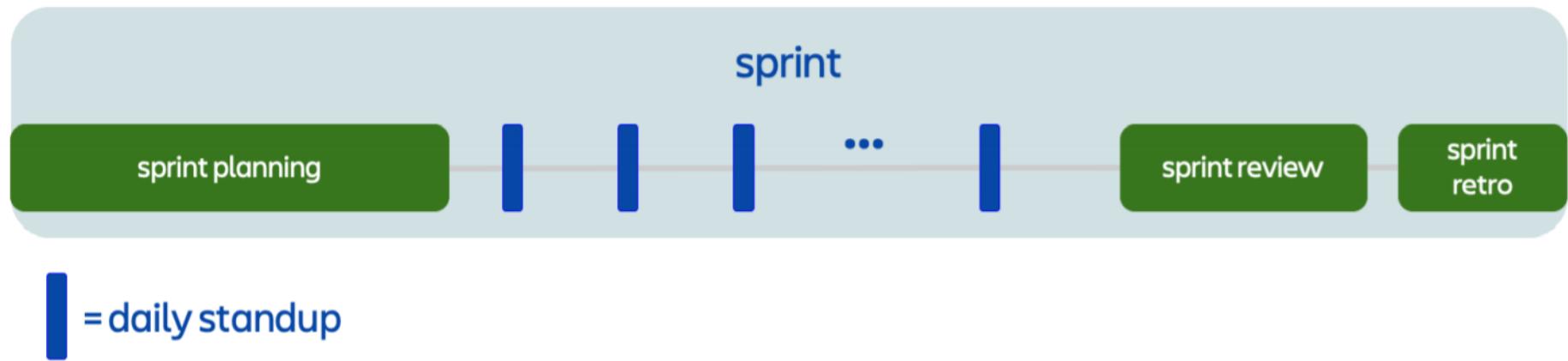
DEVELOPMENT TEAM

Topics

- Scrum roles
- Scrum Events

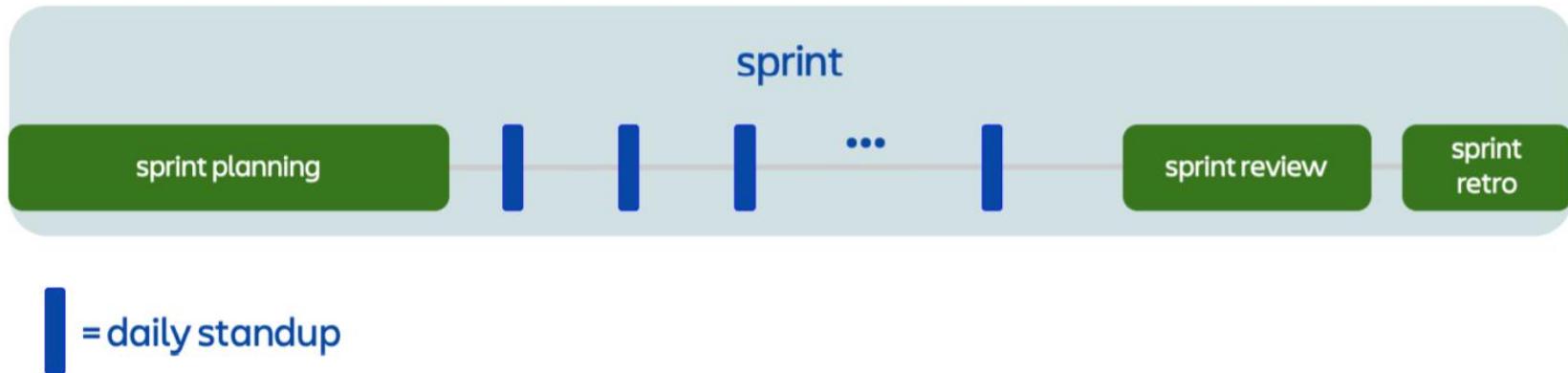


Scrum Events



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



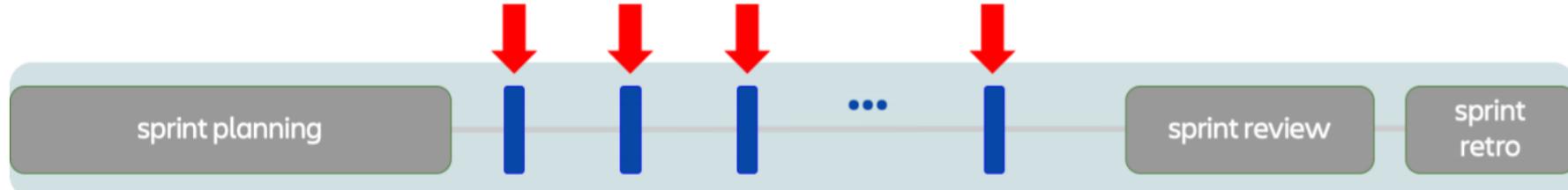
Sprint planning meeting

- **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



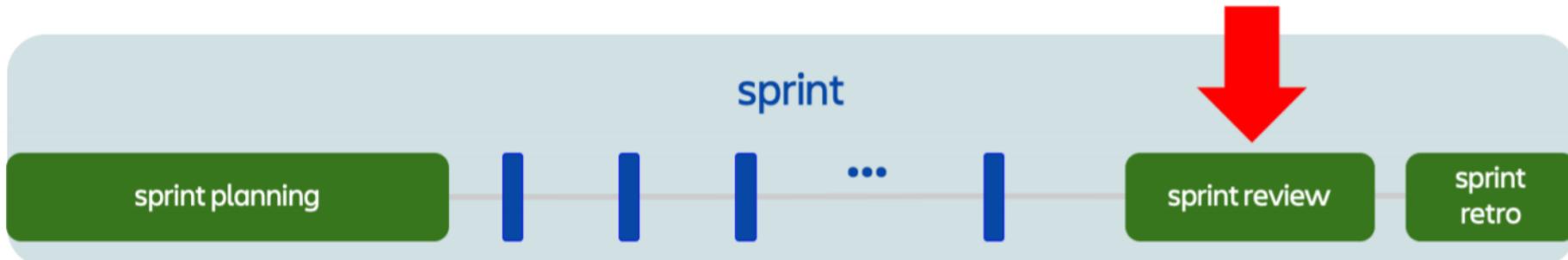
Daily Standup

- **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



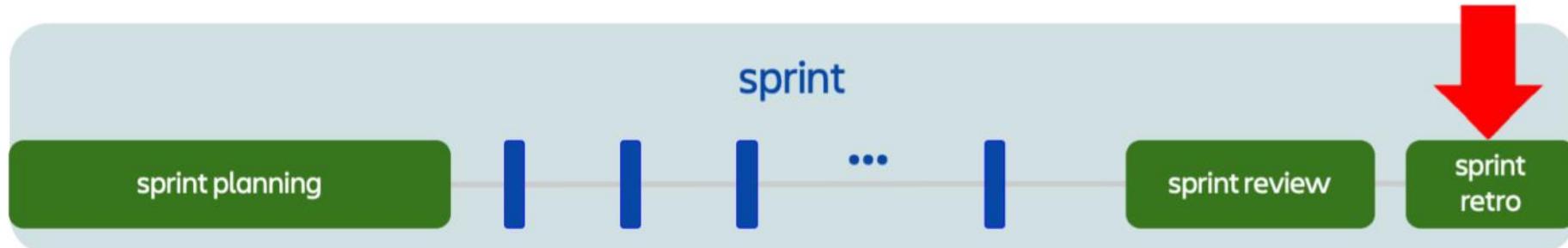
Sprint Review

- **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

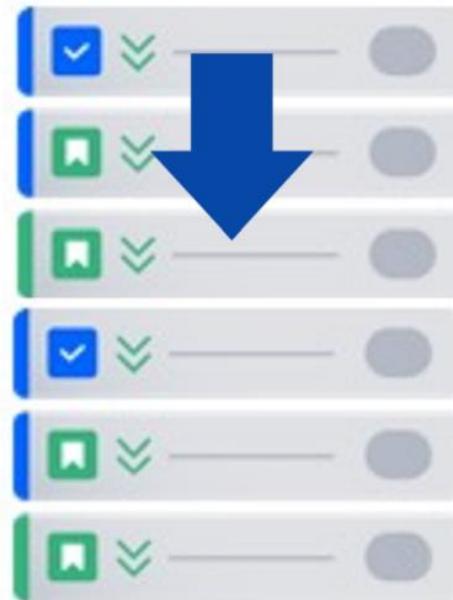


Scrum meetings - summary

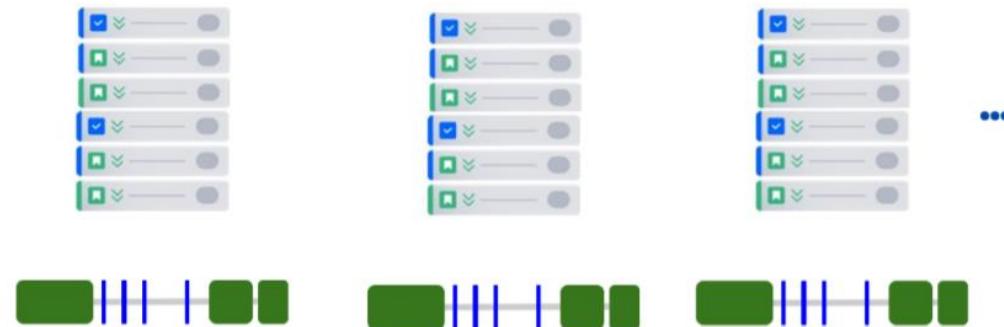
	Sprint Planning	Daily Standup	Sprint Review	Retrospective
Attendees	Scrum team	Development team (primarily)	Scrum team and stakeholders	Scrum team
Duration*	4 hours	15 minutes	2 hours	90 minutes
Purpose	Plan the work of the sprint	Inspect recent work, plan today	Inspect increment, brainstorm next sprint	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

Quick Search and Basic Search



What you will learn

- Identify the ways to search in Jira
- Use quick search
- Use basic search

Topics

- Searching overview
- Quick search
- Basic search



Viewing a project's progress



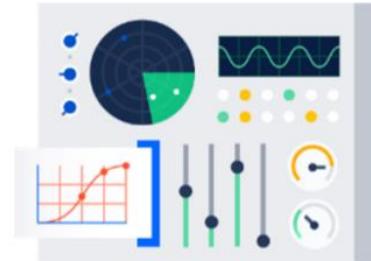
Boards



Search



Reports



Dashboards

Searching

1. Quick search

The screenshot shows the Jira search interface. At the top is a search bar with the placeholder "Search Jira". Below it is a section titled "Advanced search for issues". Underneath this are two sections: "RECENTLY VIEWED ISSUES" and "BOARDS, PROJECTS AND FILTERS".

- RECENTLY VIEWED ISSUES:**
 - PRJ-1 add item 1
 - PRJ-3 add item 3
 - PRJ-2 add item 2
- BOARDS, PROJECTS AND FILTERS:**
 - PROJ board
 - PRJ board
 - projectA (PROJ)
 - projectB (PRJ)

At the bottom of the sidebar, there are links to "Go to all:" followed by "Issues", "Boards", "Projects", "Filters", and "People".

Searching

1. Quick search
2. Basic search (user interface elements)

All issues [Save as](#)

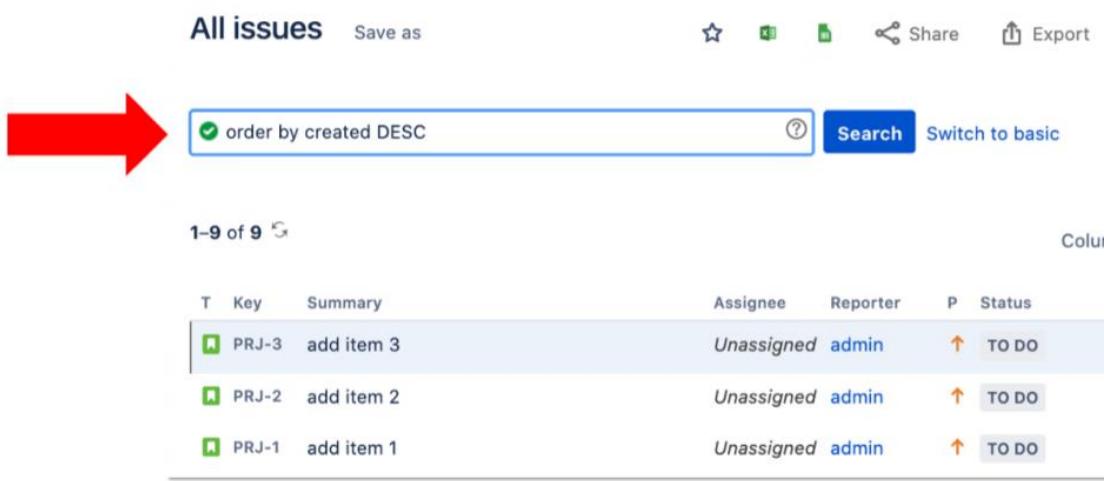
 Project: All ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ + More [Search](#) [Switch to JQL](#)

1–9 of 9 [S](#)

T	Key	Summary	Assignee	Reporter	P	Status
	PRJ-3	add item 3	Unassigned	admin		TO DO
	PRJ-2	add item 2	Unassigned	admin		TO DO
	PRJ-1	add item 1	Unassigned	admin		TO DO

Searching

1. Quick search
2. Basic search (user interface elements)
3. Advanced search (JQL)

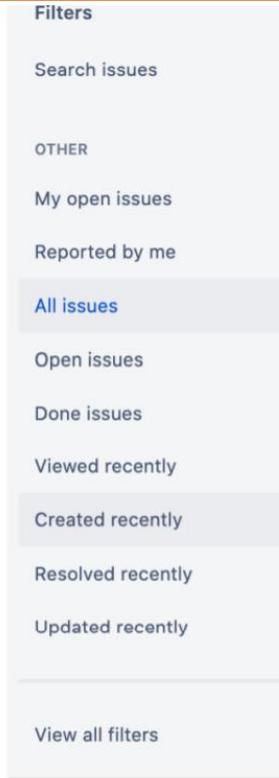


The screenshot shows a Jira search results page titled "All issues". At the top, there are several icons: a star, a green square, a blue square, a share button, and an export button. Below these are three buttons: "Save as", "Share", and "Export". A red arrow points to the search bar, which contains the text "order by created DESC". To the right of the search bar are a help icon, a "Search" button, and a "Switch to basic" link. The search results show 1-9 of 9 items. The columns are Key, Summary, Assignee, Reporter, and Status. The data is as follows:

T	Key	Summary	Assignee	Reporter	P	Status
	PRJ-3	add item 3	Unassigned	admin	↑	TO DO
	PRJ-2	add item 2	Unassigned	admin	↑	TO DO
	PRJ-1	add item 1	Unassigned	admin	↑	TO DO

Searching

1. Quick search
2. Basic search (user interface elements)
3. Advanced search (JQL)
4. Filters



Searching

1. Quick search
2. Basic search (user interface elements)
3. Advanced search (JQL)
4. Filters
5. Quick filters

Projects / projectA / PROJ board

Kanban board

The screenshot shows a Jira Kanban board for projectA. The board has three columns: "SELECTED FOR DEVELOPMENT 2", "IN PROGRESS 2", and "REVIEW 0". Each column contains two items, each with a title, a checkmark icon, and a "PROJ-X" label where X is a number. The top navigation bar is highlighted with a red box and contains a search bar, a user profile icon (labeled 'A'), and filter buttons for "Only My Issues" and "Recently Updated".

Column	Item 1	Item 2
SELECTED FOR DEVELOPMENT 2	add item 3 PROJ-3	add item 2 PROJ-2
IN PROGRESS 2	add item 5 PROJ-5	add item 4 PROJ-4
REVIEW 0		

Why is searching important?

Adapt your Jira experience to your team's processes

Projects / projectA / PROJ board

Kanban board

The screenshot shows a Jira Kanban board for projectA/PROJ board. The board has three columns: "SELECTED FOR DEVELOPMENT 2" (Min 2), "IN PROGRESS 2", and "REVIEW 0". Each column contains three items, each with a checkmark, up arrow, down arrow, and a person icon. The items are labeled "add item 3", "add item 2", "add item 4" and "add item 5", "PROJ-3", "PROJ-2", and "PROJ-4".

Column	Item 1	Item 2	Item 3
SELECTED FOR DEVELOPMENT 2 (Min 2)	add item 3 PROJ-3	add item 2 PROJ-2	add item 4 PROJ-4
IN PROGRESS 2			
REVIEW 0			

Topics

- Searching overview
- Quick search
- Basic search



Quick search

The image displays two side-by-side screenshots of the Jira interface, illustrating the differences between the Cloud and Server versions regarding the quick search feature.

Cloud Version (Left):

- Search Bar:** A large search bar at the top with the placeholder "Search Jira".
- Advanced Search:** A button labeled "Advanced search for issues".
- Recently Viewed Issues:** A list of recent items:
 - PRJ-1 add item 1
 - PRJ-3 add item 3
 - PRJ-2 add item 2
- Boards, Projects and Filters:** A list of boards and projects:
 - PROJ board
 - PRJ board
 - projectA (PROJ)
 - projectB (PRJ)
- Navigation:** A footer with links to "Go to all:" followed by "Issues", "Boards", "Projects", "Filters", and "People".

Server Version (Right):

- Search Bar:** A search bar at the top with the placeholder "Search".
- Issues:** A list of recent issues:
 - add item 5a
PROJ-7
 - add item 2
PROJ-2
 - add item 2
PROJF-2
 - add item 1
PROJF-1
 - initial release
PROJF-4
 - add item 3
PROJF-3
- View All Issues:** A link to view all issues.
- Projects:** A list of recent projects:
 - projectA (PROJ)
Software
 - projectfinal (PROJF)
Software
- View All Projects:** A link to view all projects.

Cloud

Server

Quick search- with search terms and keywords

feature

ISSUES

-  add feature 2 PROJ-2
-  add feature 3 PROJ-3
-  add feature 1 PROJ-1
-  add feature 2 PRJ-2
-  add feature 3 PRJ-3
-  add feature 1 PRJ-1

 View all matching issues

feature NOT 1

ISSUES

-  add feature 2 PROJ-2
-  add feature 3 PROJ-3
-  add feature 2 PRJ-2
-  add feature 3 PRJ-3

 View all matching issues

feature OR sample

ISSUES

-  add feature 3 PROJ-3
-  add feature 2 PROJ-2
-  add feature 3 PRJ-3
-  add feature 2 PRJ-2
-  Instructions for deleting this sample SAM-17
-  add feature 1 PROJ-1

Topics

- Searching overview
- Quick search
- Basic search



Basic search

Search Save as

Project: All ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ + More Contains text Search Switch to JQL

Search Save as

projectA ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ + More Contains text Search Switch to JQL Columns ▾

Find Projects...

	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
<input checked="" type="checkbox"/> projectA (PROJ)	Ryan Lee	admin	↑	IN PROGRESS	Unresolved	01/Jul/20	02/Jul/20	
<input type="checkbox"/> projectB (PRJ)	Alana Grant	admin	↑	IN PROGRESS	Unresolved	01/Jul/20	02/Jul/20	
<input checked="" type="checkbox"/> PROJ-2 add item 2	Unassigned	admin	↑	SELECTED FOR DEV...	Unresolved	01/Jul/20	01/Jul/20	
<input checked="" type="checkbox"/> PROJ-3 add item 3	Unassigned	admin	↑	SELECTED FOR DEV...	Unresolved	01/Jul/20	01/Jul/20	
<input checked="" type="checkbox"/> PROJ-1 add item 1	Unassigned	admin	↑	DONE	Done	01/Jul/20	01/Jul/20	
<input checked="" type="checkbox"/> PROJ-6 add item 6	Unassigned	admin	↑	BACKLOG	Unresolved	01/Jul/20	01/Jul/20	
<input checked="" type="checkbox"/> PROJ-5 add item 5	Unassigned	admin	↑	SELECTED FOR DEV...	Unresolved	01/Jul/20	01/Jul/20	

1-6 of 6

Contains Text

Search Save as

projectA ▼ Type: All ▼ Status: All ▼ Assignee: All ▼ + More item NOT 1 Search Switch to JQL

1-5 of 5 

T	Key	Summary	Assignee	Reporter	P	Status	Resolutio
<input checked="" type="checkbox"/>	PROJ-4	add item 4	Ryan Lee	admin	 IN PROGRESS	Unresol	
<input checked="" type="checkbox"/>	PROJ-2	add item 2	Alana Grant	admin	 IN PROGRESS	Unresol	
<input checked="" type="checkbox"/>	PROJ-3	add item 3	Unassigned	admin	 SELECTED FOR DEV...	Unresol	
<input checked="" type="checkbox"/>	PROJ-6	add item 6	Unassigned	admin	 BACKLOG	Unresol	
<input checked="" type="checkbox"/>	PROJ-5	add item 5	Unassigned	admin	 SELECTED FOR DEV...	Unresol	

1-5 of 5 



Searching more fields

A screenshot of a search interface, likely from a project management tool like Jira. The top navigation bar includes 'Search' and 'Save as' buttons, followed by dropdown filters for 'projectA', 'Type: All', 'Status: All', and 'Assignee: All'. To the right of these filters are buttons for '+ More', 'Contains text', 'Search', and 'Switch to JQL'. A large red arrow points downwards from the '+ More' button towards a detailed search criteria panel.

The main search results table shows 5 items:

T	Key	Summary	Assignee	Due Date	Status
<input checked="" type="checkbox"/>	PROJ-4	add item 4	Unassigned	2023-10-05	In Progress
<input checked="" type="checkbox"/>	PROJ-2	add item 2	Unassigned	2023-10-05	In Progress
<input checked="" type="checkbox"/>	PROJ-3	add item 3	Unassigned	2023-10-05	Approved
<input checked="" type="checkbox"/>	PROJ-6	add item 6	Unassigned	2023-10-05	Pending Review

Below the table, it says '1-5 of 5'.

The detailed search criteria panel, triggered by the '+ More' button, is titled 'All Criteria' and lists various search options:

- % Limits
- [CHART] Date of First Response
- [CHART] Time in Status
- Approvals
- Change completion date
- Change reason

At the bottom of the criteria panel, it says '...excluding 9 hidden'.

Example- Specifying a priority

Search Save as

projectA Type: All Status: All Assignee: All + More Contains text Search Switch to JQL

Priority: All ×

Find Priorities... ×

- ↑ Highest
- ↑ High
- ↑ Medium
- ↓ Low
- ↓ Lowest

	Assignee	Reporter	P	Status	Resolution
<input checked="" type="checkbox"/> PROJ-6 add item 6	Ryan Lee	admin	↑	IN PROGRESS	Unresolved
<input checked="" type="checkbox"/> PROJ-5 add item 5	Alana Grant	admin	↑	IN PROGRESS	Unresolved
			SELECTED FOR DEVELOPMENT	SELECTED FOR DEV...	Unresolved
			BACKLOG		Unresolved
			SELECTED FOR DEV...		Unresolved

1-5 of 5 ↶

The screenshot shows a search interface for a project named 'projectA'. The search bar includes filters for Type, Status, Assignee, and a 'Priority' dropdown set to 'All'. A modal window titled 'Find Priorities...' lists priority levels: Highest, High, Medium, Low, and Lowest, each with a checkbox and an upward or downward arrow icon. Below this is a table of five items. The first two items have checkboxes checked and are labeled 'PROJ-6 add item 6' and 'PROJ-5 add item 5'. The third item has a 'SELECTED FOR DEVELOPMENT' button. The fourth item has a 'BACKLOG' button. The fifth item has a 'SELECTED FOR DEV...' button. Columns include Assignee, Reporter, Priority (P), Status, and Resolution.

Example- Specifying an updated date

Search [Save as](#)

projectA Type: All Status: All Assignee: All + More Contains text [Search](#) [Switch to JQL](#)

Updated Date: All [X](#)

Within the last 3 days [▼](#)

More than minutes ago [▼](#)

Between 11-Jan-2019 and 30-Jan-2019 [▼](#)

In range -3w 4d to 3w 4d [▼](#)

[Update](#) [Close](#)

PROJ-5 add item 5

1-5 of 5 [G](#)

Assignee	Reporter	P	Status	Resolu
Ryan Lee	admin	↑	IN PROGRESS	Unres
Alana Grant	admin	↑	IN PROGRESS	Unres
Unassigned	admin	↑	SELECTED FOR DEV...	Unres
Unassigned	admin	↑	BACKLOG	Unres
Unassigned	admin	↑	SELECTED FOR DEV...	Unres

Takeaways

- Quick search can search the text of issues, board names, project names and filter names
- Basic search is a user-friendly way to search for issues

Lab 9 - Quick Search and Basic Search

- Perform quick searches
- Perform basic searches

JQL



What will you learn?

- Describe Jira Query Language (JQL)
- Write JQL using autocomplete
- Use functions in JQL queries

Topics

- JQL Overview
- Autocomplete
- Functions



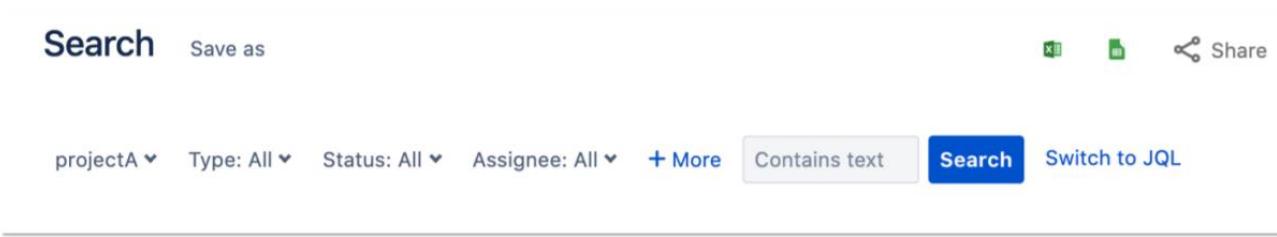
Basic and advanced/JQL Search

basic

Search Save as

X E Share

projectA Type: All Status: All Assignee: All + More Contains text Search Switch to JQL



advanced/JQL

Search Save as

E Share Export

project = PROJ order by lastViewed DESC

?

Search Switch to basic

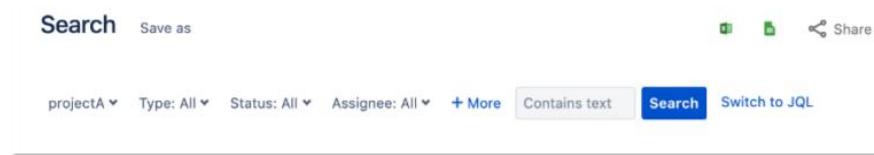


All issues Save as

order by created DESC

Jira Query Language (JQL)- Searches for issues and orders results

Basic vs. advanced/JQL Search



A screenshot of a search interface. At the top, there are buttons for 'Search' and 'Save as'. On the right, there are icons for file operations and a 'Share' button. Below these are dropdown menus for 'projectA', 'Type: All', 'Status: All', 'Assignee: All', and a '+ More' button. There is also a text input field labeled 'Contains text' and a blue 'Search' button. To the right of the search bar is a link 'Switch to JQL'.

Basic search

- User-friendly interface
- Queries can be complex, but there



Advanced/JQL search

- Uses JQL
- Most powerful search method
- JQL can be used in automation scripts

Writing “JQL” the easiest way

A screenshot of a search interface, likely from a project management tool. At the top left, there are dropdown filters for 'projectA' (selected), 'Type: All', 'Status: All', 'Assignee: All', and a '+ More' button. To the right are buttons for 'Contains text', 'Search', and 'Switch to JQL'. A large red arrow points from the 'Switch to JQL' button to a box on the right containing the generated JQL query. The query is: `project = PROJ ORDER BY created DESC`. Below the search bar, it says '1-6 of 6'.

projectA ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ + More Contains text Search Switch to JQL

1-6 of 6

project = PROJ ORDER BY created DESC

1-6 of 6

The two main parts of JQL query

 project = PROJ order by created DESC

search clause - selects a
subset of issues

order by clause - orders the
results

The Simplest JQL Query



All issues — Edited Save as ▾

1-29 of 29 ⏪ Columns ▾

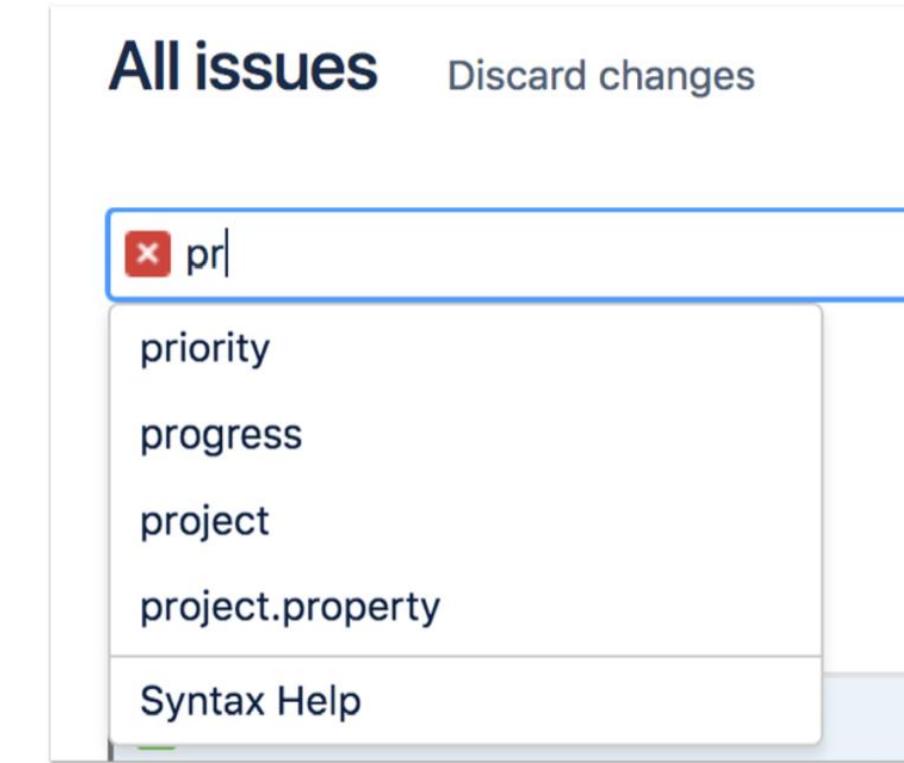
T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	SAM-23	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	09/Jun/18	...
	SAM-22	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	07/Jun/18	
	SAM-21	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	04/Jun/18	
	SAM-20	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	02/Jun/18	
	SAM-19	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	31/May/18	
	SAM-18	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	28/May/18	
	SAM-17	Instructions for deleting this sample board and project are in the description for this issue >> Click the "SAM-17" link and read the description tab of the detail view for more	Steve Byrnes	Steve Byrnes	↑	DONE	Done	10/Jun/18	14/Jun/18	

Topics

- JQL Overview
- Autocomplete
- Functions



JQL with autocomplete - fields



Advanced searching fields reference

The screenshot shows a blue header bar with the Jira Software Support logo, navigation links for Get started, Documentation (which is underlined), Knowledge base, Resources, a search icon, and a user profile icon. Below the header, a breadcrumb trail shows Atlassian Support / Jira Software / Documentation / ... / ... / Advanced searching. To the right are Cloud and Server dropdowns. The main content area has a title "Advanced searching - fields reference". A paragraph explains what JQL is and how clauses work. Below it is a section titled "Affected version" with a table of field details. To the right is a sidebar titled "On this page" listing various JQL fields.

This page describes information about fields that are used for advanced searching. A field in JQL is a word that represents a Jira field (or a custom field that has already been defined in your Jira applications). In a clause, a field is followed by an operator, which in turn is followed by one or more values (or functions). The operator compares the value of the field with one or more values or functions on the right, such that only true results are retrieved by the clause. Note: it is not possible to compare two fields in JQL.

Affected version

Search for issues that are assigned to a particular affects version(s). You can search by version name or version ID (i.e. the number that Jira automatically allocates to a version). Note, it is better to search by version ID than by version name. Different projects may have versions with the same name. It is also possible for your Jira administrator to change the name of a version, which could break any saved filters that rely on that name. Version IDs, however, are unique and cannot be changed.

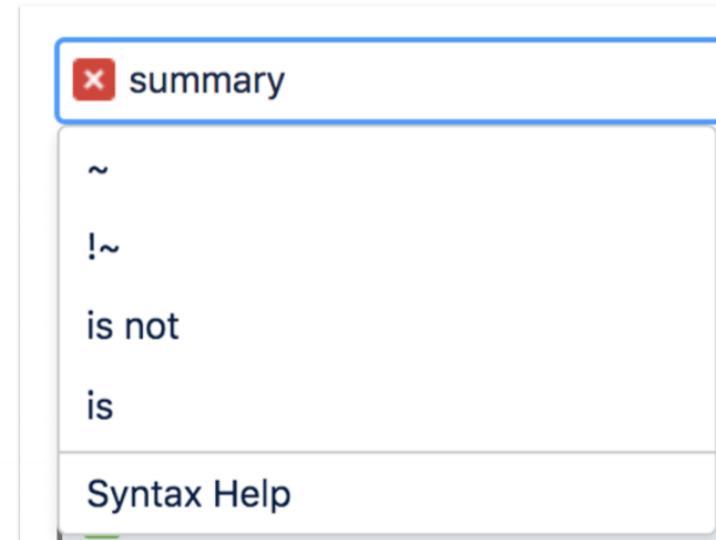
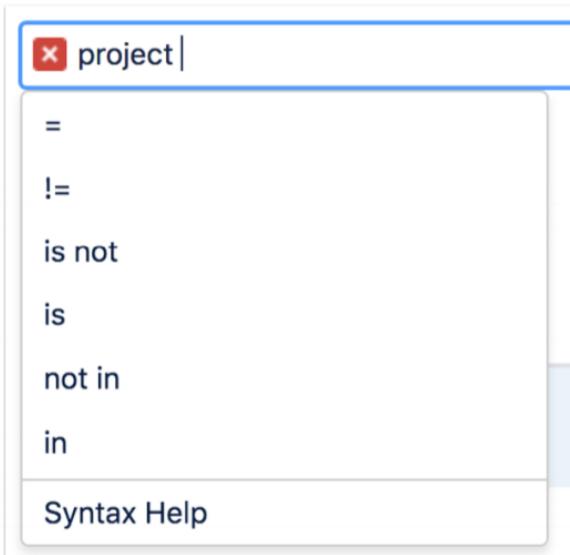
Syntax	affectedVersion
Field Type	VERSION
Auto-complete	Yes
Supported operators	= , != , > , >= , < , <= IS, IS NOT, IN, NOT IN Note that the comparison operators (e.g. ">") use the version order that has been set up by your project

On this page

- Affected version
- Approvals
- Assignee
- Attachments
- Category
- Comment
- Component
- Created
- Creator
- Custom field
- Customer Request Type
- Description
- Due
- Environment

Operator autocomplete

<field name> <operator> <value>
project = projectA



Boolean Operators

- **AND**
- **OR**
- **NOT**

```
assignee = currentUser() AND status = "In Progress"

status = "Selected for Development" OR status = "In Progress"
status in ("Selected for Development", "In Progress")

NOT status = Backlog
status != Backlog

find unresolved issues in all projects except SampleA
resolution = Unresolved AND NOT project = SampleA
```

Advanced searching – operators reference

The screenshot shows a web browser window with the Jira Core Support documentation. The title bar includes the Jira Core Support logo, a navigation menu with 'Documentation' selected, and a search icon. The main content area has a breadcrumb trail: Atlassian Support / Jira Core / Documentation / ... / ... / Advanced searching. Below this, there are two tabs: 'Cloud' and 'Server'. The main heading is 'Advanced searching - operators reference'. A sub-section titled 'EQUALS: =' is shown, explaining that it finds issues where a field exactly matches a value. It also notes that it cannot be used with text fields and instead uses the 'CONTAINS' operator. It provides examples for finding issues created by 'jsmith' and 'John Smith'. To the right, a sidebar lists other operators: NOT EQUALS: !=, GREATER THAN: >, GREATER THAN EQUALS: >=, LESS THAN: <, LESS THAN EQUALS: <=, IN, NOT IN, CONTAINS: ~, DOES NOT CONTAIN: !=~, IS, IS NOT, and WAS.

Advanced searching - operators reference

This page describes information about operators that are used for advanced searching.

An operator in JQL is one or more symbols or words, which compares the value of a field on its left with one or more values (or functions) on its right, such that only true results are retrieved by the clause. Some operators may use the NOT keyword.

EQUALS: =

The “=” operator is used to search for issues where the value of the specified field exactly matches the specified value. (Note: cannot be used with text fields; see the `CONTAINS` operator instead.)

To find issues where the value of a specified field exactly matches *multiple* values, use multiple “=” statements with the `AND` operator.

Examples

- Find all issues that were created by jsmith:

```
reporter = jsmith
```
- Find all issues that were created by John Smith:

```
reporter = "John Smith"
```

On this page

- EQUALS: =
- NOT EQUALS: !=
- GREATER THAN: >
- GREATER THAN EQUALS: >=
- LESS THAN: <
- LESS THAN EQUALS: <=
- IN
- NOT IN
- CONTAINS: ~
- DOES NOT CONTAIN: !=~
- IS
- IS NOT
- WAS

Executing the query

Search

Save as



Share



Export

project = projectA



Search

Switch to basic

1-6 of 6

Color

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated
<input checked="" type="checkbox"/>	PROJ-6	add item 6	Unassigned	admin	↑	BACKLOG	Unresolved	01/Jul/20	01/Jul
<input checked="" type="checkbox"/>	PROJ-5	add item 5	Unassigned	admin	↑	SELECTED FOR DEV...	Unresolved	01/Jul/20	01/Jul
<input checked="" type="checkbox"/>	PROJ-4	add item 4	Ryan Lee	admin	↑	IN PROGRESS	Unresolved	01/Jul/20	02/Ju

Search

Save as

The associated basic search

projectA ▾

Type: All ▾

Status: All ▾

Assignee: All ▾

+ More

Contains text

Search

Switch to JQL

Topics

- JQL Overview
- Autocomplete
- Functions



Search clauses

<field name> <operator> <**field value**>
project = projectA

<field name> <operator> <**function**>
assignee = currentUser()



Advanced searching - functions reference

This page describes information about functions that are used for advanced searching.

A function in JQL appears as a word followed by parentheses, which may contain one or more explicit values or Jira fields. In a clause, a function is preceded by an operator, which in turn is preceded by a field. A function performs a calculation on either specific Jira data or the function's content in parentheses, such that only true results are retrieved by the function, and then again by the clause in which the function is used.



Unless specified in the search query, note that JQL searches do not return empty fields in results. To include empty fields (e.g. unassigned issues) when searching for issues that are not assigned to the current user, you would enter `(assignee != currentUser() OR assignee is EMPTY)` to include unassigned issues in the list of results.

approved()

Only applicable if Jira Service Desk is installed and licensed.

Search for requests that required approval and have a final decision of approved.

On this page

- approved()
- approver()
- breached()
- cascadeOption()
- closedSprints()
- completed()
- componentsLeadByUser()
- currentLogin()
- currentUser()
- earliestUnreleasedVersion()
- elapsed()
- endOfDay()
- endOfMonth()

Time based Functions

- startOfDay()
- startOfWeek()
- startOfMonth()
- startOfYear()
- endOfDay()
- endOfWeek()
- endOfMonth()
- endOfYear()
- now()
- currentLogin()
- lastLogin()

Issues created since the start of today

created > startOfDay()

The unit qualifier

(+|-)nn(y|M|w|d|h|m)

Tip: Use basic search to create the query

created in the last 2 days (48 hours):

created > -2d

created since the start of day 2 days ago:

created > startOfDay (-2d)

created since the 15th of this month

created > startOfMonth (+14d)

Takeaways

- AJQL query is behind all basic and advanced searches
- Leverage basic queries and autocomplete to simplify creating JQL queries
- JQL queries may select subsets of issues and/or order query results
- Functions can be used to avoid hard-coding values in a search clause
- Time unit qualifiers (yMwdhm) can be used with date-related values

Lab10-JQL

- Create a basic search and view the JQL query
- Create JQL queries using autocomplete
- Use functions as values

Filters



What will you learn?

- Create filters
- Describe board filters
- Use quick filters

Topics

- Filters
- Board filters
- Quick filters



Filters

Filters

Search issues

OTHER

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

[View all filters](#)

My open issues

Save as

assignee = currentUser() AND resolution = Unresolved order by updated DESC

[?](#) [Search](#) [Switch to basic](#) [≡](#)

1–1 of 1 [S](#)

Columns [▼](#)

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated ▼	Due
	PRJ-1	add item 1	admin	admin	↑	TO DO	Unresolved	01/Jul/20	02/Jul/20	...

1–1 of 1 [S](#)

Saving a search creates a filter

The screenshot shows a Jira search interface. On the left, a sidebar lists various filters: Filters, Search issues, OTHER, My open issues (which is selected and highlighted in blue), Reported by me, All issues, Open issues, Done issues, Viewed recently, Created recently, Resolved recently, Updated recently, and View all filters. The main area is titled "My open issues". A red box highlights the "Save as" button in the top right corner of the title bar. Below the title bar, there is a search query: "assignee = currentUser() AND statusCategory = "In Progress" order by updated DESC". To the right of the query are "Search" and "Switch to basic" buttons. The results table shows one item: PROJ-2 add item 2, assigned to admin, reported by admin, in IN PROGRESS status, unresolved, created on 01/Jul/20, and updated on 02/Jul/20. The table has columns for T (Ticket), Key, Summary, Assignee, Reporter, P (Priority), Status, Resolution, Created, and Updated.

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated
<input checked="" type="checkbox"/>	PROJ-2	add item 2	admin	admin	↑	IN PROGRESS	Unresolved	01/Jul/20	02/Jul/20

Save the filters

The screenshot shows the Jira interface for managing issues. On the left, there's a sidebar with a 'Filters' section and a 'Search issues' field. Below these are several filter categories: 'OTHER' (selected), 'My open issues' (highlighted in blue), 'Reported by me', 'All issues', 'Open issues', 'Done issues', 'Viewed recently', 'Created recently', 'Resolved recently', 'Updated recently', and 'View all filters'. The main area displays a list titled 'My open issues' with the query: 'assignee = currentUser() AND statusCategory = "In Progress" order by updated DESC'. It shows 1-1 of 1 issue, with a key 'PROJ-2' and a status 'In Progress'. A modal window titled 'Save Filter' is open in the center. It has a 'Filter Name' input field containing 'My in progress', a note below it saying 'Enter a name for this Filter', and two buttons at the bottom: 'Submit' (highlighted in blue) and 'Cancel'.

The newly created filter

A screenshot of a Jira interface showing a newly created filter. On the left, a sidebar lists various filters, with 'My in progress' highlighted by a red arrow. The main area shows the filter details and a table of results.

Filters

- Search issues
- STARRED**
- My in progress** (highlighted with a red arrow)
- OTHER
- My open issues
- Reported by me
- All issues
- Open issues
- Done issues
- Viewed recently
- Created recently
- Resolved recently
- Updated recently
- [View all filters](#)

My in progress Save as Details

assignee = currentUser() AND statusCategory = "In Progress" ORDER BY updated DESC

1-1 of 1

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
<input checked="" type="checkbox"/>	PROJ-2	add item 2	admin	admin	↑	IN PROGRESS	Unresolved	01/Jul/20	02/Jul/20	***

1-1 of 1

Edit filter details

The screenshot shows the Jira interface for managing filters. On the left, there's a sidebar with various filter categories like 'Search issues', 'STARRED', and 'OTHER'. The 'My in progress' filter is selected. The main area displays the 'My in progress' filter details. A red arrow points to the 'Details' tab at the top of the filter view. The filter itself has one item: 'PROJ-2 add item 2'. To the right, there are two overlapping windows: 'Edit filter' and 'Filter Subscription'. The 'Edit filter' window shows the filter name 'My in progress', access set to 'Private', and a 'Save' button. The 'Filter Subscription' window shows subscription settings for 'Personal Subscription' with daily delivery at 9:00 am.

My in progress

Details

A Owned by admin

Permissions

This filter is only visible to you.

Edit permissions

Subscriptions

This filter has no subscriptions.

New subscription

IN PROGRESS

Filter Subscription

Recipients Personal Subscription

Schedule

Daily

Days per Week

Days per Month

Advanced

Interval

once per day

at 9 00 am

The timezone is the same as your profile's timezone - (GMT+00:00), GMT

Email this filter, even if there are no issues found

Subscribe Cancel

View all filters

Filters

Search issues

STARRED

My in progress

OTHER

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

[View all filters](#)

Filters

Name *	Owner	Project	Group	Starred by:
Filter for PRJ board	A admin	projectB, All roles	0 people	***
Filter for PROJ board	A admin	projectA, All roles	0 people	***
My in progress	A admin	Private	1 person	***

[Create filter](#)

[Manage subscriptions](#)

[Copy filter](#)

[Edit](#)

[Delete](#)

Edit filters query

Filters

Search issues

STARRED

My in progress

OTHER

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

[View all filters](#)

My in progress

Edited Save Details

assignee = currentUser() AND statusCategory = "In Progress" ORDER BY updated ASC

Search Switch to basic

1-1 of 1

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated ↑	Due
<input checked="" type="checkbox"/>	PROJ-2	add item 2	admin	admin		IN PROGRESS	Unresolved	01/Jul/20	02/Jul/20	...

1-1 of 1

Topics

- Filters
- Board filters
- Quick filters



Board filters

- Every board has a filter that defines the issues shown on the board
- You can edit the board's filter
- If you create a board, you must assign it a filter

Filter

Saved Filter

Filter for PRJCT board

[Edit Filter Query](#)

Filter Query

project = PRJCT ORDER BY Rank ASC

Example Board filter

A board filter can be used to show issues from multiple projects on a single board

Filter

- Saved Filter
- Two Projects
- Edit Filter Query

Shares

No shares

Edit Filter Shares

Filter Query

project in (PRJCT, PRJ) ORDER BY Rank ASC

Kanban board

The screenshot shows a Jira Kanban board with two columns: 'SELECTED FOR DEVELOPMENT 2' and 'IN PROGRESS'. In the first column, there are two items: 'add item 1' and 'add item 2'. Each item has a 'None' status and a 'More' options menu. A red arrow points to the 'PRJCT-1' label under the second item, and a blue arrow points to the 'PRJCT-5' label under the third item.

Topics

- Filters
- Board filters
- Quick filters



Filtering a board: quick filters

Further filters issues displayed on a board

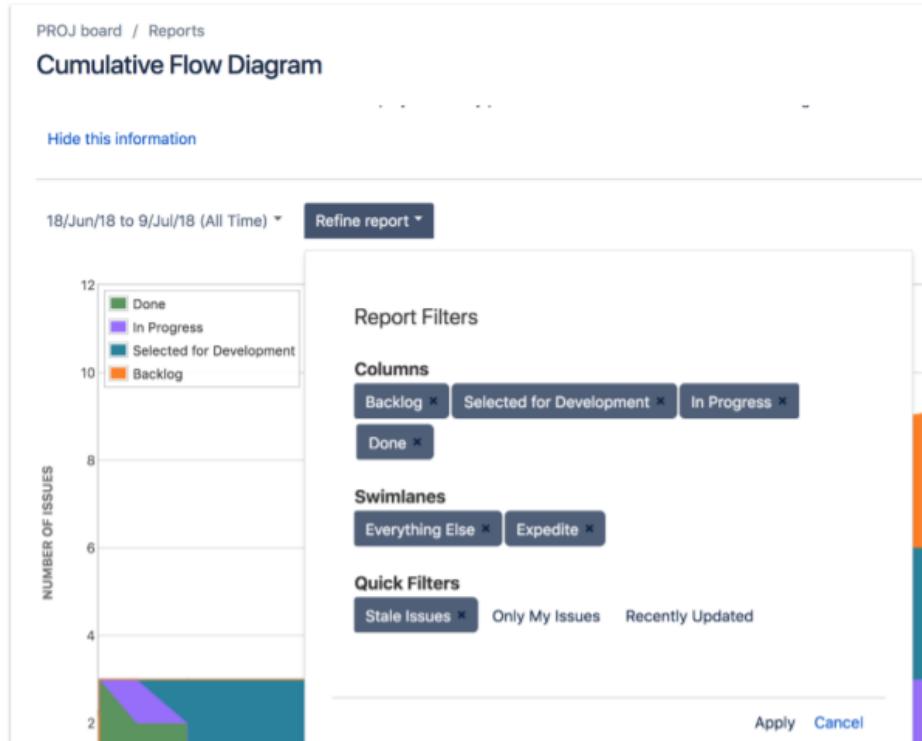
Projects / projectA / PROJ board

Kanban board

The screenshot shows a Jira Kanban board titled "Kanban board" under "Projects / projectA / PROJ board". The board has three columns: "SELECTED FOR DEVELOPMENT" (2 items), "IN PROGRESS" (2 items), and "REVIEW" (0 items). Each item card contains a title, a checkmark icon, a move-up/move-down icon, and a project identifier (PROJ-3, PROJ-2, PROJ-5, PROJ-4). At the top of the board, there is a search bar, a user profile icon (A), and three filter icons: "Only My Issues", "Recently Updated", and a magnifying glass icon. A red box highlights this top navigation area.

Column	Item 1	Item 2
SELECTED FOR DEVELOPMENT	add item 3 PROJ-3	add item 2 PROJ-2
IN PROGRESS	add item 5 PROJ-5	add item 4 PROJ-4
REVIEW		

Quick filters and report



Takeaways

- Filters are saved searches that can be exposed through user interface elements
- Every board has a filter that defines the issues shown on the board
- Quick filters are saved searches that are used to further limit the issues displayed on a board or in reports

Lab 11- Filters

- Explore default filter queries
- Create a starred filter
- Explore and create quick filters

Epics



What will you learn?

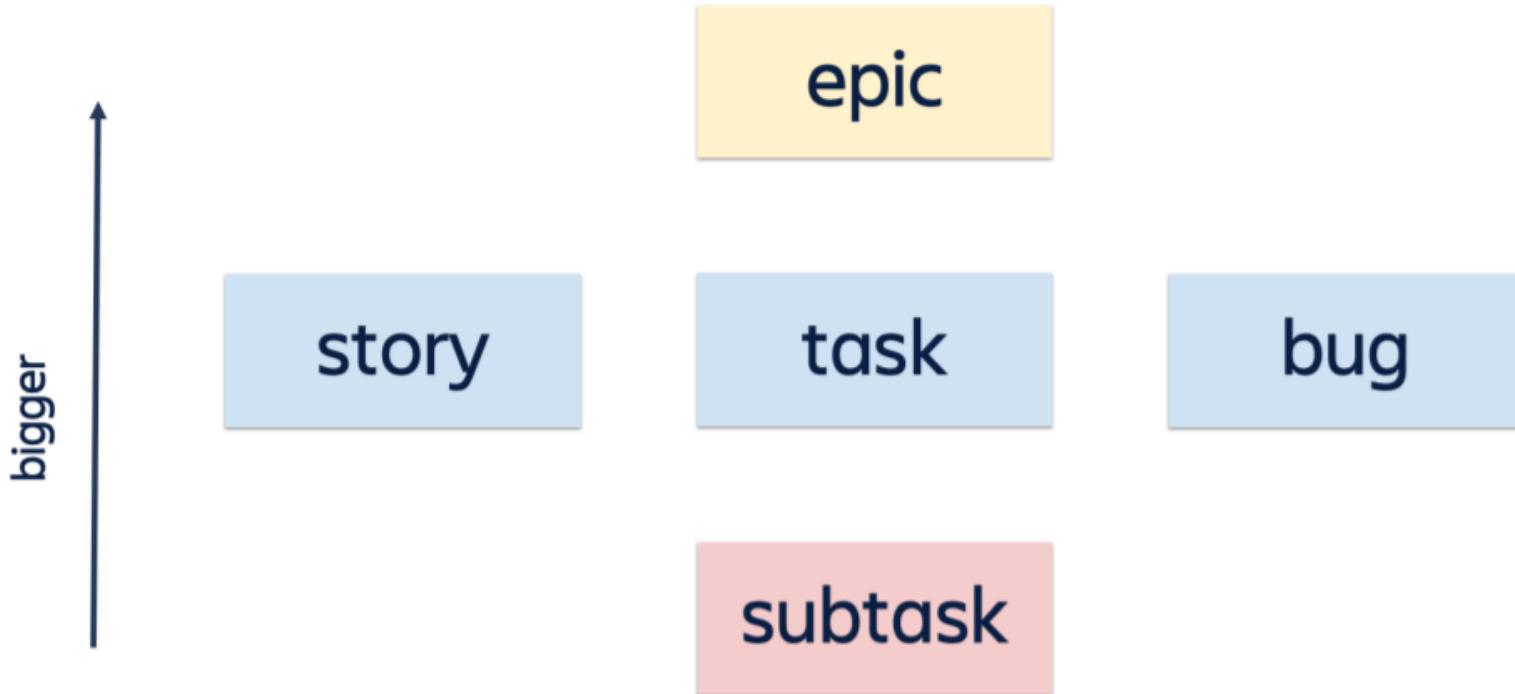
- Describe epics
- Work with epics
- Manage epics in the backlog

Topics

- Epics overview
- Working with epics
- Epics in the backlog



Jira's issue type hierarchy



Epics

- A large issue
- Can contain other issues
- Child issues can span multiple iterations, projects, teams and boards
- Can be a placeholder for many stories

Create issue

Import issues Configure fields ▾

Project* projectA (PROJ)

Issue Type* Epic

Some issue types are unavailable due to incompatible field configuration and/or workflow associations.

Epic Name* Big Feature A

Provide a short name to identify this epic.

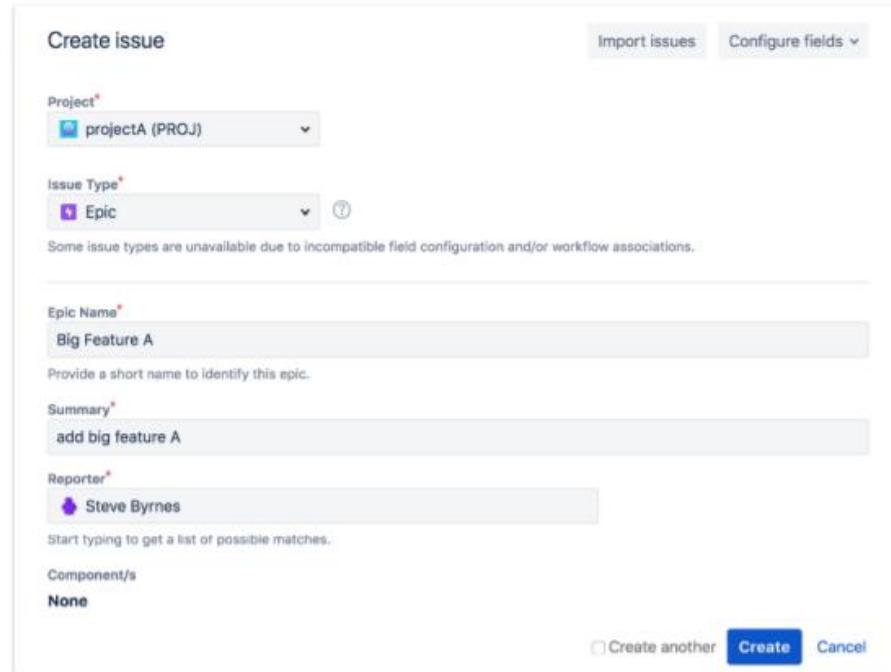
Summary* add big feature A

Reporter* Steve Byrnes

Start typing to get a list of possible matches.

Component/s None

Create another Create Cancel



Why Epics?

- Organization of work
- Span multiple iterations and projects
- Simplifies backlog (one issue)

Topics

- Epics overview
- **Working with epics**
- Epics in the backlog



Creating an epic

Create issue

[Import issues](#)
[Configure fields](#)

Project*

projectA (PROJ)

Issue Type*

Epic

Some issue types are unavailable due to incompatible field configuration and/or workflow associat

Epic Name*

Feature A

Provide a short name to identify this epic.

Summary*

add feature A

Provide a short name to identify this epic.

Components

None

Attachment

Drop files to attach, or browse.

Description

Style: **B** *I* U ~~A~~ ^A _A

Managing the issue of an epic

The screenshot shows a Jira interface for managing an epic issue. The epic is titled "add feature A".

Top Bar: PROJ-7, Give feedback, 1, Share, More, Close.

Epic Details:

- Description:** Add a description...
- Assignee:** Unassigned
- Reporter:** admin

Issues in this epic:

- PROJ-8 add feature A task 1 (checkbox checked, orange arrow pointing up, BACKLOG button)

Comment Section:

- Add a comment... (with a placeholder for pressing M to comment)

Feature A (Details):

- Story Points: None
- Original Estimate: 0m

Epic link

A field in the child issue pointing to the parent epic

The screenshot shows a Jira interface with two main components. On the left is a task card for "add feature A task 1" under "PROJ-7 / PROJ-B". The card includes fields for Description, Activity (Comments selected), and a comment input area. On the right is a larger epic card for "Feature A" under "Backlog". The epic card displays fields for Assignee (Unassigned), Reporter (admin), and Labels (None). A vertical line connects the "Link issue" button in the task card to the "Epic Link" section in the epic card, indicating the relationship.

PROJ-7 / PROJ-B

add feature A task 1

Attach Create subtask Link issue ...

Description
Add a description...

Activity
Show: Comments History Work log

Add a comment...
Pro tip: press M to comment

Backlog

Assignee
Unassigned

Reporter
admin

Labels
None

Epic Link

Feature A

Created 6 minutes ago Updated 6 minutes ago Configure

Adding an existing issue to an epic

Open the child issue and select the existing epic under Epic Link

The screenshot shows a Jira issue creation interface for 'add item 2'. At the top left is a checked checkbox labeled 'PROJ-2'. On the right are standard Jira navigation icons: 'Give feedback', a '1' in a circle, a share icon, three dots, and a close button.

The main area has a title 'add item 2' and several buttons: 'Attach', 'Create subtask', 'Link issue', and three dots. Below this is a 'Description' section with a text input placeholder 'Add a description...'. To the right of the description are status indicators: 'In Progress' (blue), 'Assignee' (admin), and 'Reporter' (admin). There is also a 'Labels' section showing 'None'.

The 'Activity' section includes tabs for 'Comments' (selected), 'History', and 'Work log'. A comment input field says 'Add a comment...' with a 'Pro tip: press M to comment' note. A large red arrow points from the text above to the 'Epic Link' section.

The 'Epic Link' section contains a blue-bordered input field labeled 'Select Epic' and a card below it for 'Feature A' (PROJ-7).

Searching for issues of an epic

Search

Save as

 "Epic Link" = PROJ-7

Feature A - (PROJ-7)

Feature A - (PROJ-7)

Syntax Help

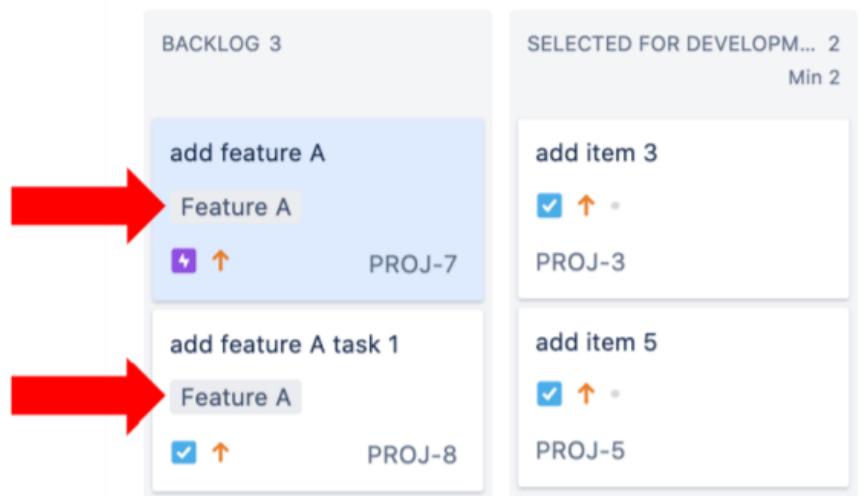
T	Key	Summary	Assignee	Reporter
<input checked="" type="checkbox"/>	PROJ-8	add feature A task 1	Unassigned	admin

1-1 of 1 

Epics labels on a board

Projects / projectA / PROJ board

Kanban board



Epic based swimlanes

Projects / projectA / PROJ board

Kanban board

Only My Issues Recently Updated

BACKLOG 3

SELECTED FOR DEVELOPM... 2

IN PROGRESS 2

▼ Feature A 1 issue

add feature A task 1

Feature A



PROJ-B

▼ Issues without epics 6 issues

add item 6



PROJ-6

add item 3



PROJ-3

add item 2



PROJ-2

add item 5

add item 4



Topics

- Epics overview
- Working with epics
- **Epics in the backlog**



Creating an epic from a backlog

Projects / projectB / PRJ board

Backlog

The screenshot shows a Jira backlog interface. At the top, there is a navigation bar with 'Projects / projectB / PRJ board' and a search bar. Below the navigation bar, the word 'Backlog' is displayed. In the center, there is a modal dialog box titled 'Create epic'. The dialog has two tabs: 'EPICS' (which is currently selected) and 'All issues'. A large red arrow points upwards towards the 'EPICS' tab. To the right of the tabs, there are filter options 'Only My Issues' and 'Recently Updated'. The main content area shows a section for 'PRJ Sprint 1' which contains 2 issues. It includes a description: 'Create the first product increment.' and a date range: '01/Jul/20 4:38 PM - 15/Jul/20 4:38 PM'. Below this, there is a list of items with green add icons: 'add item 1' and 'add item 2'. Further down, there is a section for 'Backlog' which contains 1 issue, with an 'add item 3' button below it. At the bottom of the dialog, there is a '+ Create issue' button.

EPICS Create epic X

VERSIONS

All issues

PRJ Sprint 1 2 issues

Create the first product increment.
01/Jul/20 4:38 PM - 15/Jul/20 4:38 PM

A ...

add item 1

add item 2

Backlog 1 issue

add item 3

+ Create issue

Epics panel

Can monitor the status of issues in the epic

Projects / projectB / PRJ board

Backlog

The screenshot shows the Jira Backlog interface for the 'PRJ board' under 'projectB'. At the top, there are filters for 'Only My Issues' and 'Recently Updated'. A red box highlights the 'EPICS' section on the left, which contains a search bar, user profile, and buttons for 'Create epic' and 'X'. Below this, it lists 'All issues' and a specific epic named 'feature B' (PRJ-4 add feature B). The epic details show 3 issues, 2 completed, 0 unestimated, and an estimate of 6. It also includes links to 'Create issue in epic' and 'View linked pages'. To the right, the backlog is organized into sprints: 'PRJ Sprint 1' (2 issues), 'PRJ Sprint 2' (1 issue), and 'Backlog' (1 issue). Each sprint has a summary card with an 'add item' button, the epic name, the sprint name, and the count of items.

EPICS Create epic X

All issues

feature B
PRJ-4 add feature B

Issues 3

Completed 2

Unestimated 0

Estimate 6

Create issue in epic

View linked pages

Issues without epics

PRJ Sprint 1 2 issues

0 0 3 Linked pages 0 ...

Create the first product increment.
01/Jul/20 4:38 PM × 15/Jul/20 4:38 PM

A ...

add item 1 feature B PRJ-1 ↑ 1

add item 2 feature B PRJ-2 ↑ 2

Backlog 1 issue Create sprint ...

add item 3 feature B PRJ-3 ↑ 3

+ Create issue

Mark an epic as done

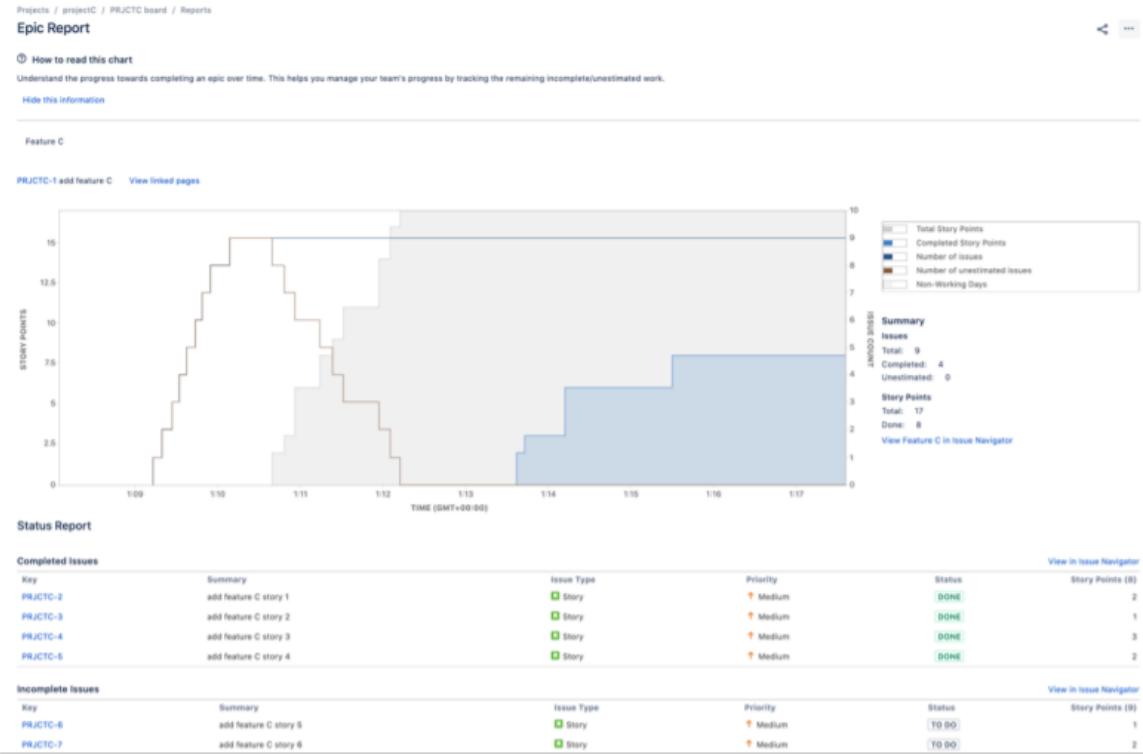
An epic marked as done will be removed from the epics panel

Projects / projectB / PRJ board

Backlog

The screenshot shows the Jira Backlog interface for the 'PRJ board'. A modal window is open over the backlog, specifically for an epic titled 'feature B'. The modal has tabs for 'EPICS' (selected), 'Create epic', and 'x'. It displays statistics: 'Issues' (3), 'Completed' (3), 'Unestimated' (0), and 'Estimate' (6). Below these are links to 'Create issue in epic' and 'View linked pages'. At the bottom of the modal, a button labeled 'Mark as Done' is highlighted with a red box. To the right of the modal, there's a 'Color' palette and options to 'Edit name' and 'View epic details'.

Epics Report



Scrum - Epics burndown report



Takeaways

- An epic is a large issue of issue type "epic" that may contain other issues
- The "Epic Link" field is used to associate Issues with an epic
- Epics can be shown on boards or in backlogs

Lab12- Epics

- Create an issue of type
- Add issues to the epic
- View swimlanes by epic
- View the epic in the kanban backlog
- Complete an epic

Dashboards



What will you learn?

- Describe dashboards
- Configure a dashboard
- Display a dashboard as a wallboard

Visualizing work



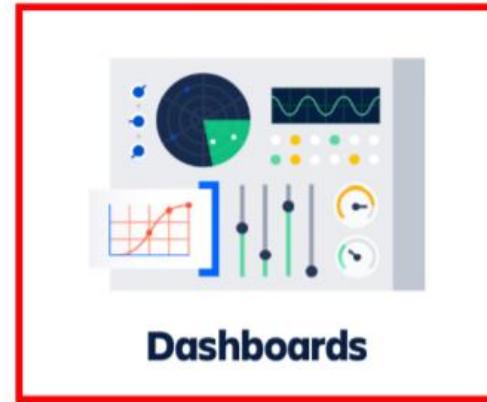
Boards



Reports



Search



Dashboards

Dashboards

The dashboard consists of several gadgets:

- Sprint Health Gadget:** Displays overall sprint progress (Story Points) at 5, with 5 days left. Breakdown: Time elapsed (0%), Work complete (0%), Scope change (0%), Blocker (0%), and Flagged (0%).
- Sprint Burndown Gadget:** A burndown chart from Nov 3 to Nov 9. The Y-axis is Story Points (0 to 6). The X-axis is TIME. A legend indicates Guideline (blue), Remaining Values (red), and Non-Working Days (grey).
- Activity Stream:** Shows a feed of recent activity from "Your Company JIRA". Items include:
 - Alana Grant updated the Sprint of PROJ-B-6 - add item 8 (1 minute ago)
 - Alana Grant updated the Sprint of PROJ-B-7 - add item 7 (1 minute ago)
 - Alana Grant updated the Story Points of PROJ-B-7 - add item 9 (1 minute ago)
 - Alana Grant created PROJ-B-7 - add item 9 (2 minutes ago)
 - Alana Grant updated the Story Points of PROJ-B-6 - add item 8 (2 minutes ago)
- Assigned to Me:** A list of tasks assigned to the user:
 - TIS-57 Buttons need to be red
 - TIS-58 Add Features for Flight School Teachers
 - TIS-59 Enhancements for Flight School
 - PROJ-B fix bug 1
 - PROJ-11 add big feature A
 - PROJ-F-1 add item 1
 - PROJ-F-2 Create initial release
 - PROJ-F-3 add item 6
 - PROJ-B-5 add item 7

- Configurable view of the work of one or more projects
- Can be personal or shared
- Contains gadgets

Types of Gadgets

Add a gadget

Search

CATEGORIES

All	31
Charts	8
Jira	27
Other	3
Wallboard	7

Discover powerful apps compatible with your JIRA version via the Atlassian Marketplace.

The screenshot shows the Atlassian Marketplace interface. At the top, there's a search bar and dropdown menus for 'Staff-picked', 'Dashboard gad...', and 'All paid & free'. A red box highlights the 'Dashboard gad...' dropdown. Below the search bar, there are two app cards:

- Xray Test Management for Jira** by Xpand IT (Top Vendor). It has a 4-star rating (231 reviews), 6,562 installations, and a price of \$2500. It's categorized under CUSTOM FIELDS, DASHBOARD GADGETS, JIRA SERVICE DESK, TESTING & QA, and REPORTS. The description states: "Xray is a Jira App for managing quality assurance tests which integrates seamlessly with Jira. Xray supports manual and automated tests and provides reports to track test coverage of your Jira projects. Xray includes a RESTful API out of the box."
- Arsenale Dataplane - Jira Reports** by Arsenale (Top Vendor). It has a 3-star rating (55 reviews), 1,142 installations, and a price of \$1250. It's categorized under CHARTS & DIAGRAMMING, DASHBOARD GADGETS, PROJECT MANAGEMENT, and REPORTS. The description states: "Dataplane delivers powerful, intuitive Jira reports and metrics for managers. Dozens of built-in reports; save, share and subscribe to reports; display on dashboards; export to PDF and Excel."

- Contains more dashboard gadgets
- Select “Dashboard gadgets”

Adding gadgets

Add gadget Edit layout ...

Add a gadget

Search

CATEGORIES

- All 31
- Charts 8
- Jira 27
- Other 3
- Wallboard 7

Activity Stream
By Atlassian + Local
Lists recent activity in a single project, or in all projects.
[Show XML link](#) [Add gadget](#)

Agile Wallboard Gadget
By Atlassian + Local
Displays a board as a Wallboard gadget.
[Show XML link](#) [Add gadget](#)

Assigned to Me
By Atlassian + Local
Displays all unresolved issues assigned to me
[Show XML link](#) [Add gadget](#)

Average Age Chart
By Atlassian + Local
Displays the average number of days issues have been unresolved.
[Show XML link](#) [Add gadget](#)

Configuring a gadget

The screenshot shows a configuration dialog for a 'Sprint Health Gadget'. At the top, it says 'Sprint Health Gadget'. Below that, there are dropdown menus for 'Board' (set to 'PROJB board') and 'Sprint' (set to 'Next Sprint Due (auto)'). Under these, there are three checked checkboxes: 'Show board name', 'Show sprint name', and 'Show assignees'. At the bottom, there is an 'Auto refresh' section with a checkbox for 'Update every 15 minutes'. At the very bottom are 'Save' and 'Cancel' buttons.

A context menu is displayed above the configuration dialog, showing options: 'Minimize', 'Delete', and 'Edit'. A red arrow points to the 'Edit' option, indicating it is the target of the configuration action.

Sharing Dashboards

The screenshot shows a user interface for managing a dashboard. At the top right, a context menu is open with options: Add gadget, Edit layout, Copy Dashboard, Edit Dashboard, Share Dashboard (which is highlighted in blue), Delete Dashboard, Find Dashboards, Create Dashboard, View as wallboard, View wallboard slideshow, and Set up wallboard slideshow.

The main area contains the following fields:

- Name: Team Sprint Dashboard
- Description: (empty)
- Favorite: (no star icon)
- Shares: Not shared
- Add Shares: Project (dropdown), Teams in Space (dropdown), All (selected, dropdown), + Add (button)

A tooltip for the 'All' share option lists the following roles:

- Project Roles
- Administrators
- Developers
- Users

Wallboards

- Turn any dashboard into a wallboard
- Acts as an information radiator

The screenshot shows a Jira dashboard titled "Sample Sprint 2" with the subtitle "0 days left". The dashboard features a Gantt chart at the top left, followed by four large progress bars: "Time elapsed" (100%), "Work complete" (43%), "Scope change" (55%), and "Blockers" (0). Below these are two tables: "Stories to be triaged" and "Untriaged bugs". The "Stories to be triaged" table lists various story types with their respective counts and percentages. The "Untriaged bugs" table does the same for bug types. To the right of the dashboard is a context menu with the following options: Add gadget, Edit layout, ..., Copy Dashboard, Edit Dashboard, Share Dashboard, Delete Dashboard, Find Dashboards, Create Dashboard, View as wallboard (which is highlighted in blue), View wallboard slideshow, and Set up wallboard slideshow.

Takeaways

- Dashboards display the work of projects
- Dashboards can be shared or used personally
- Gadgets display a portion of a dashboard
- Dashboards can be shown as a wallboard to radiate information

Lab 13 - Dashboards

- Create a dashboard
- Display a dashboard as a wallboard

Putting it all together



Topics

- Course overview
- Jira family
- Wrap up



Combined lean and agile principles

1. Empower the team

- Select motivated individuals
- Teams should self-organize
- Collaborate to create shared understanding

2. Visualize work

3. Experiment using the scientific method

- Continuously learn and improve
- Embrace change
- Partner with the customer
- Continuously inspect and adapt

4. Plan, develop and deliver incrementally

- Prefer conversations for conveying information
- Continuously refactor to maintain agility
- Maintain a sustainable pace
- Completed work items are the primary measure of progress
- Obtain fast feedback

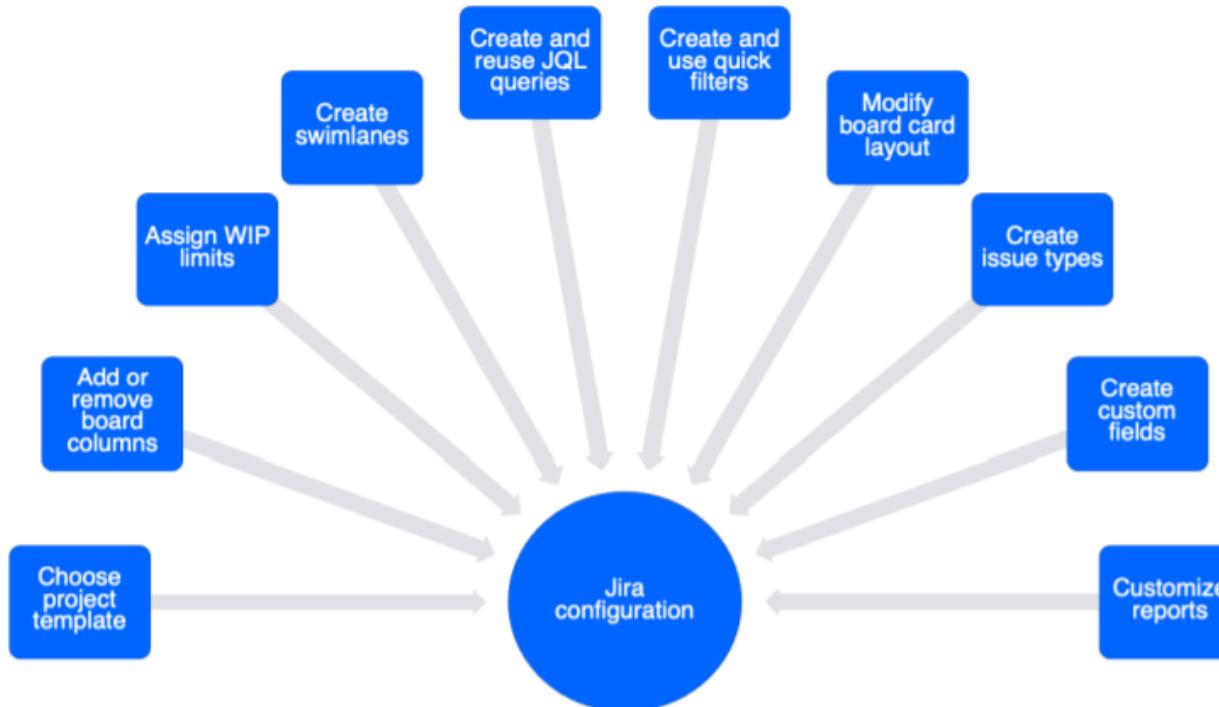
5. Improve the "flow" of value

- Limit work in progress
- Map the value stream
- Pull work
- Eliminate waste
- Reduce setup times
- Automate what should be automated
- Continuously strive for simplicity

6. Build quality in

- Don't compromise on quality
- The process should identify problems
- Fix problems when they are discovered
- Identify and fix the root cause

Ways to configure jira to match your team's process



Topics

- Course overview
- Jira family
- Wrap up



Jira family of products



Jira Core
Business project management
software

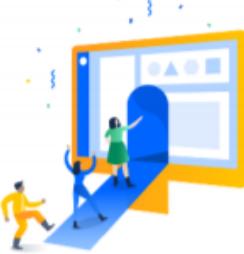


Jira Software
Plan, track, and release
software



Jira Service Desk
Service desk software for IT
teams

Onboarding



Tasks



Create email account



Acquire Laptop



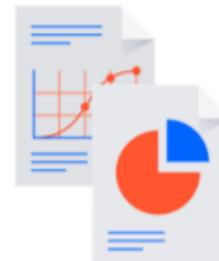
Video chat with new employee



Jira Core

Create & manage

Track progress



Mobile App Development



Manage backlog



Backlog



Manages board



Board



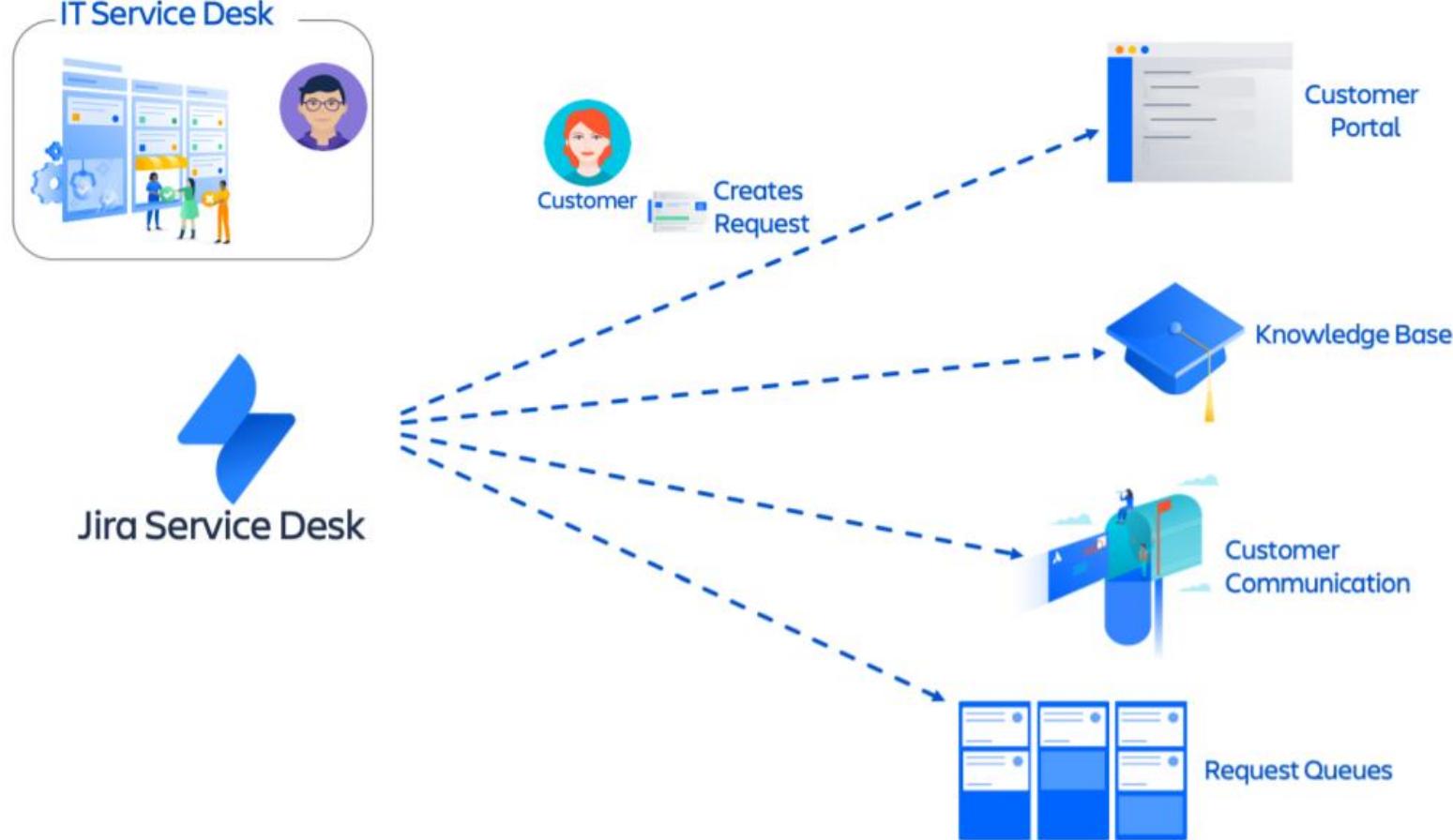
Jira Software



Monitor pull requests



Code Repository



Lab 14- Final project (optional)

- Modify a Jira project to match your team's ever-changing agile processes

Topics

- Course overview
- Jira family
- Wrap up



Congratulations on completing the course!

