

JiraEssentials



COURSE OVER



What will you learn?

- Describe Jira
- Identify the differences between kanban and scrum projects
- Use Jira to organize, find and report on your work
- Configure Jira to match your team's processes

To succeed here, you need to have

- No knowledge of Jira is assumed

Live teach schedule (1 of 2)

1	Course Overview	20 minutes
2	Jira Overview	30 minutes
3	Project Boards	35 minutes
4	Enrich Issues	40 minutes
5	Kanban Projects	20 minutes
6	Scrum Projects	25 minutes

Live teach schedule (2 of 2)

7	Quick Search and Basic Search	25 minutes
8	JQL	20 minutes
9	Filters	15 minutes
10	Epics	20 minutes
11	Dashboards	15 minutes
12	Putting it all Together	15 to 40 minutes*
	Total	~ 5 hours

The lab for module 12 is optional

Jira Cloud vs Jira Server/Data Center

The screenshot shows the Jira Cloud interface. At the top, there's a navigation bar with links for Jira, Your work, Projects, Filters, Dashboards, People, Apps, and Create. A search bar is also at the top. Below the navigation is a 'Kanban board' section for 'projectA'. The board has four columns: BACKLOG, IN PROGRESS, and DONE, with a 'RELEASE' column at the top. Each column contains several cards representing issues. On the left side, there's a sidebar with project settings like PRO board, Kanban board, Reports, Releases, Issues and filters, Pages, On-call schedule, Components, Add item, and Project settings.

Cloud

The screenshot shows the Jira Server/Data Center interface. It has a similar top navigation bar and search bar. The main area is a 'Kanban board' for 'projectA'. The board includes columns for RELEASE, SELECTED FOR DEVELOPMENT, IN PROGRESS, and RELEASED. There are multiple cards on each column. The left sidebar includes options like PROJ board, Backlog, Kanban board, Releases, Reports, Issues, Components, and Project settings.

Server /
Data Center

Jira Cloud-classic vs next-gen projects

The diagram illustrates the difference between Jira Cloud-classic and next-gen project interfaces. It features two main sections connected by dashed arrows.

Left Section (Cloud-classic interface):

- Create project ▾** button at the top.
- Classic project** section:
 - Icon: Blue folder.
 - Description: All the power and functionality you expect. Created and managed by your Jira admin.
- Try a next-gen project** section:
 - Icon: Purple folder with a yellow star.
 - Description: Easy setup and reimagined features. Created and managed by project team members.

Right Section (Next-gen interface):

- Kanban board** view for project "proj0C":
 - Header: Projects / proj0C / proj0 board.
 - Filters: Only My Issues, Recently Updated.
 - Columns: BACKLOG (3 items), IN PROGRESS (2 items), REVIEW (1 item), and DONE (1 item).
 - Items: Add Item 1, Add Item 2, Add Item 3.
- PROJ board** view for project "proj0C":
 - Header: Projects / proj0C / proj0 board.
 - Filters: Label, Type.
 - Columns: ISSUE (1 item), IN PROGRESS (2 items), REVIEW (1 item), and DRAFT (1 item).
 - Items: add item-A, add item-B, add feature-1, add feature-2, add feature-3.
 - Note at bottom: You're in a next-gen project. Edit settings > Learn more.

Lab 1- Course Overview

- Decide if you want to do the Cloud or Server version of the Labs
- Log in to Jira

Jira Overview



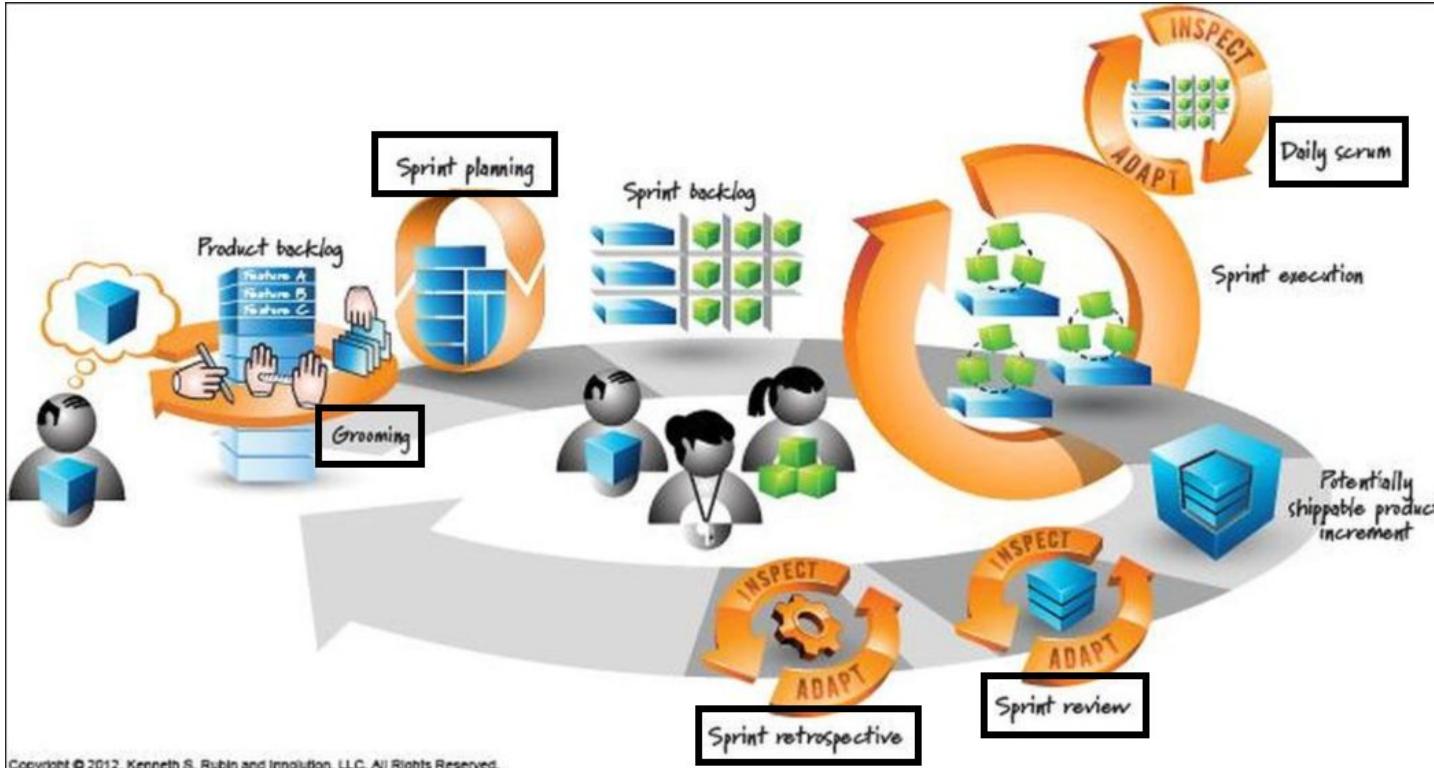
What will you learn?

- Describe Jira
- Create a Jira project
- Create a Jira issue
- Use a project board
- Identify Jira user types

Agility & Ways of Working

- Tony Goulart and Ernesto Lee

Overall Scrum Process



Copyright © 2012, Kenneth S. Rubin and Innovation, LLC. All Rights Reserved.

Backlog Grooming



Backlog Grooming

Purpose and Context

- Prepare User Stories for sprint planning
- Develop shared understanding of Customer needs as represented by user Stories
- Partner with Product Owner to optimize delivery of business value

Level of Facilitation

- Full-scale facilitation by the Product Owner
- Well-designed, closely facilitated task oriented, outcome driven, time-boxed

Outcomes / Results / Deliverables

- Well Groomed User Stories
Include:
 - Business Acceptance Criteria
 - Understand how to demo
- Understand how to Test
- Understand / Extend Definition of Ready and Done for the Stories

Tools & Techniques

- Triangulation
- Powerful Questions
- Listening
- ATDD (Acceptance Test Driven Development)

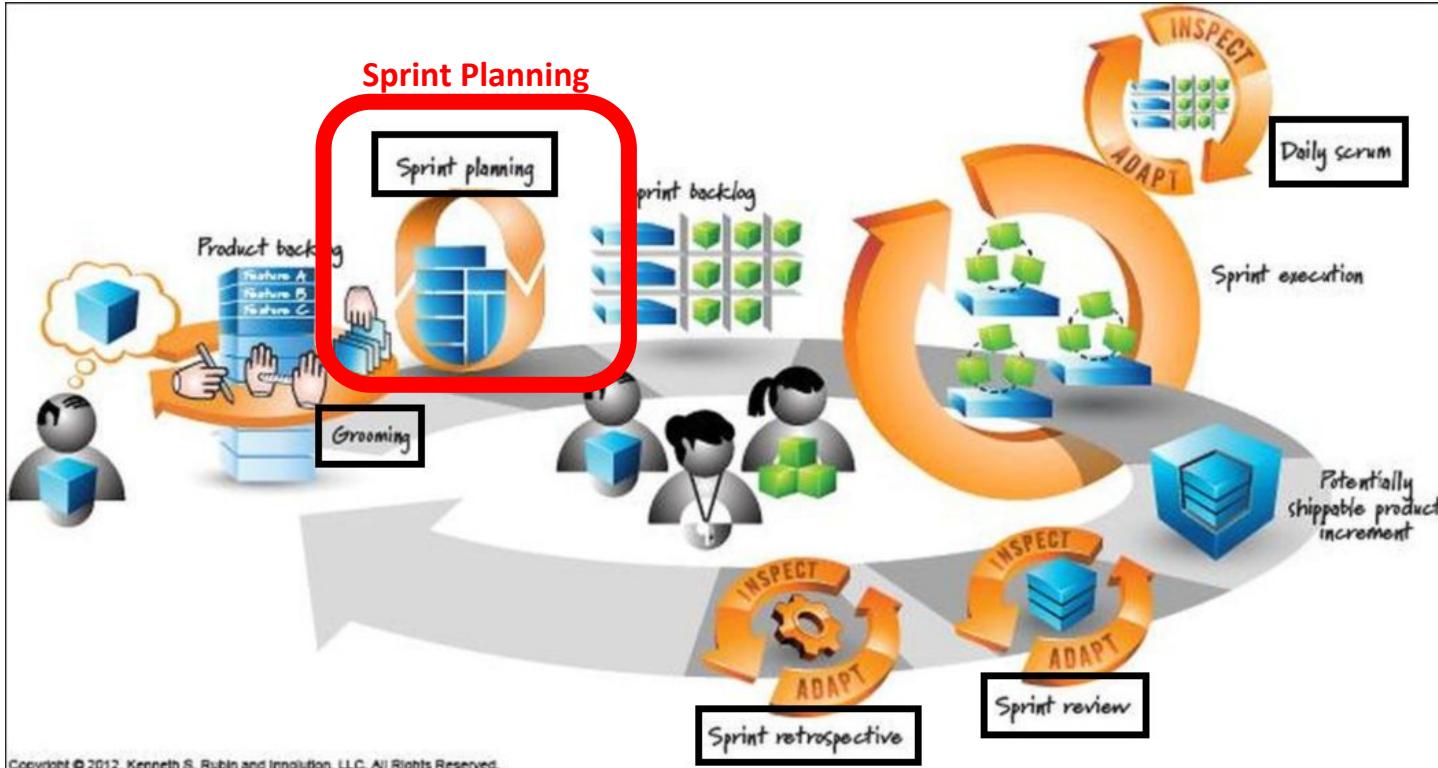
Success Factors

- Engaging & authentic
- Stories moving towards Definition of Ready
- Encourage conversation
- Transparency/Safety

Pitfalls

- Not actually done
- Going thru the motions
- Fear
- Insufficient support from the business stakeholders

Sprint Planning



Copyright © 2012, Kenneth S. Rubin and Innovation, LLC. All Rights Reserved.

Sprint Planning

Purpose and Context

- Aligns team and Product Owner on work plan / commitment
- Management gets transparency into the team's thinking instead of needing a detailed long range plan
- Helps team focus and control their destiny
- Provides empirical measure of teams capability / capacity

Level of Facilitation

- Full-scale facilitation by the Scrum Master with Product Owner
- Well-designed, closely facilitated task oriented, outcome driven, time-boxed

Outcomes / Results / Deliverables

- Defined Iteration Goals
- Iteration Backlog with acceptance criteria
- Team iteration Commitment
- Deeper shared understanding
- User Story Task Breakdown
- User Story Size Estimates

Tools & Techniques

- User Story Grooming
- Silent Work
- Brainstorming
- Story Refactoring
- Task Bucket Estimating
- Time Boxing
- Working Edge Cases First
- Just Enough / Just In Time

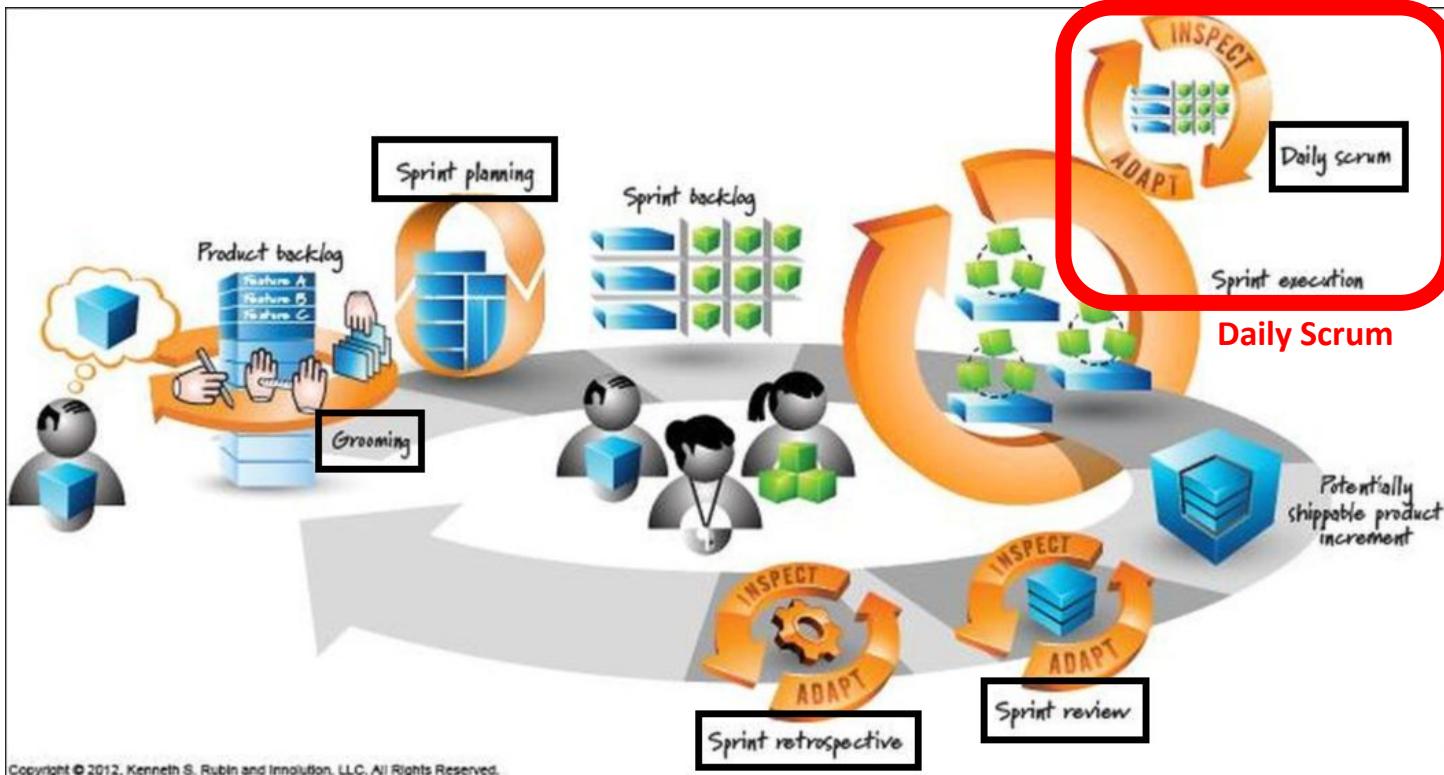
Success Factors

- Highly Engaging
- Team Alignment
- Clear desired outcome

Pitfalls

- Ungroomed user Stories
- Wishful Thinking
- Undue Influence from senior team members

Daily Scrum



Copyright © 2012, Kenneth S. Rubin and Innovation, LLC. All Rights Reserved.

Daily Scrum

Purpose and Context

- Helps Team coordinate their actions in detail
- Allows micro-planning and emergent discoveries
- Daily team commitment meeting (Tasks <1 day)
- Provides visibility into team's process

Level of Facilitation

- Varies based on team maturity (Shu, Ha, Ri), task oriented, outcome driven, time-boxed
- Facilitated by the Scrum Master and be rotated by members on the team

Outcomes / Results / Deliverables

- Team coordination
- Early identification of impediments
- Visibility for non- team members
- Team mutual accountability

Tools & Techniques

- Time Boxing
- Bottom lining
- Just Enough / Just In Time
- “Talking Stick”
- Working the Task Board

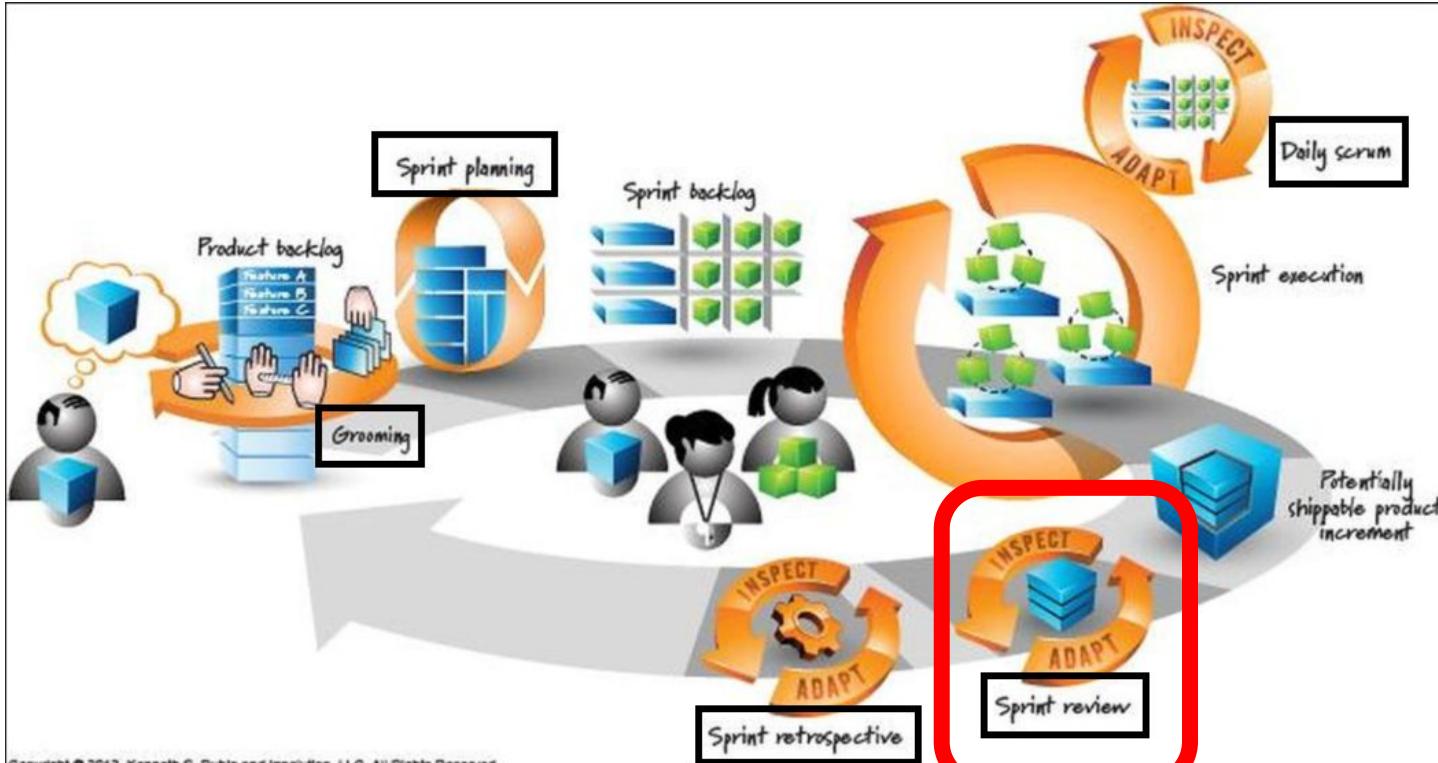
Success Factors

- Start on time
- Rigorous Time-boxing
- Commitment based conversation
- Full permission to interrupt

Pitfalls

- Going through the motions
- Digression into details
- Sitting down
- Entire Team not present

Sprint Review



Sprint Review

Sprint Review

Purpose and Context

- Review the goal of the sprint and compare to what actually got accomplished during sprint
- Demonstrate progress “show off” the product
- Real-time customer feedback from all stakeholders
- Allows management to give feedback & make requests
- Accept the iteration

Level of Facilitation

- Full-scale facilitation by Scrum Master until a team member wants to, and is capable of, facilitating; then facilitation can rotate.

Outcomes / Results / Deliverables

- Feedback on work completed vs goal
- Demonstrate work completed
- Shared understanding of value delivered
- Improved team morale
- Team takes public responsibility for success/failure (bonds them)

Tools & Techniques

- Time Boxing
- Bottom lining
- Information sharing
- Powerful Questions
- Listening

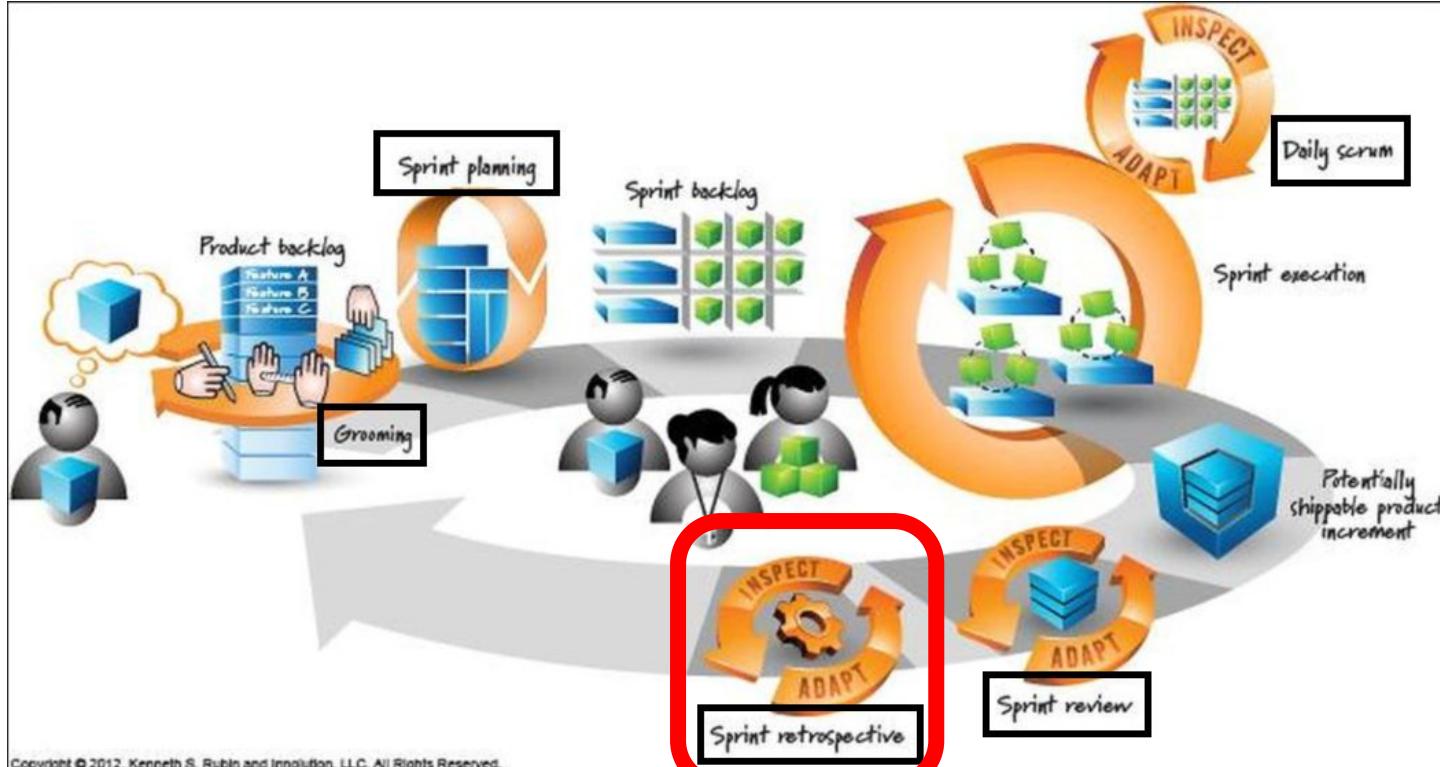
Success Factors

- Engaging & authentic
- “Real” working software
- Encourage conversation
- Transparency/Safety

Pitfalls

- Going through the motions
- PowerPoint
- Fear
- One way information flow
- Lack of interest from business stakeholders

Retrospective



Sprint Retrospective

Retrospective

Purpose and Context

- Primary focal point of the Inspect and Adapt cycle
- The way for the team to take responsibility for their process
- Focuses on three major areas: productivity, process & team

Level of Facilitation

- Full-scale facilitation owned by the Scrum Master until a team member wants to, and is capable of, facilitating; then facilitation can rotate

Outcomes / Results / Deliverables

- Greater team understanding of what actually happened
- Short list (e.g. 1 – 3) of “actions” to improve
- Inspect and Adapt regarding Productivity/process/team
- Team bonding (we’re in this together)

Tools & Techniques

- Steps from *Agile Retrospectives Book*
- Meeting is owned by the team, therefore, they may invite non-team members
- Prod Owner may attend IF they are member of team (rather than an external customer)
- Light and frequent follow up is crucial during sprint

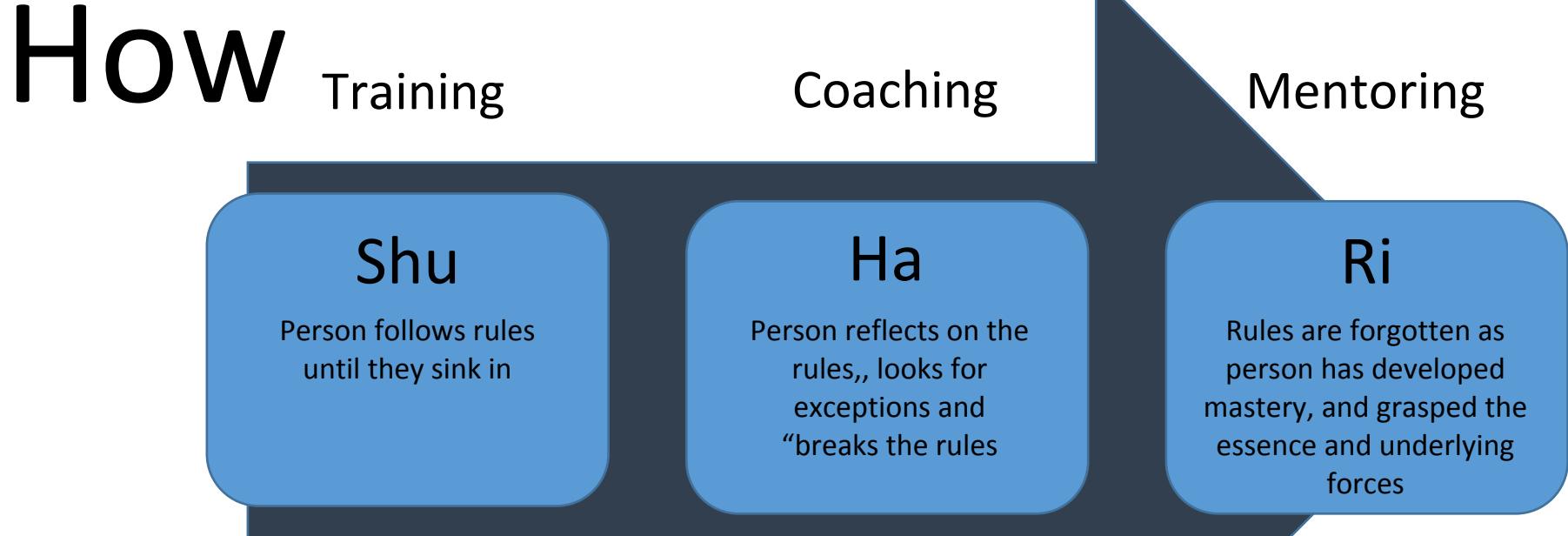
Success Factors

- Planned ahead
- Skillfully facilitated
- Team interacts with the team directly
- All voices are heard

Pitfalls

- Same old – same old
- Everything “is outside the team’s control” (never completely true)

How do we adopt this?



Who Student

Apprentice

Master

Agility – Stages of Learning

- 1. Shu (Following Precisely without Modification) – 2. Ha – 3. Ri



Agility – Stages of Learning

- 1. Shu - 2. Ha (Shifting Between Techniques) - 3. Ri



Agility – Stages of Learning

- 1. Shu - 2. Ha- 3. Ri (New Techniques, even unconsciously)



How do we adopt this?

How

Training

Coaching

Mentoring

Shu

Person follows rules
until they sink in

Ha

Person reflects on the
rules,, looks for
exceptions and
“breaks the rules

Ri

Rules are forgotten as
person has developed
mastery, and grasped the
essence and underlying
forces

Who

Student

Apprentice

Master

Team and Scope

The Team

- Roles:
 - Product Manager (PdM)
 - Scrum Master + Team = Development
- All skills needed to implement
 - Generalists, versatile
 - May include specialists
- Egos are put aside
 - High Trust
 - Is it good enough? Not looking for perfection
- Empowered and Self Organized
 - Members accountable for planning and delivery as a team

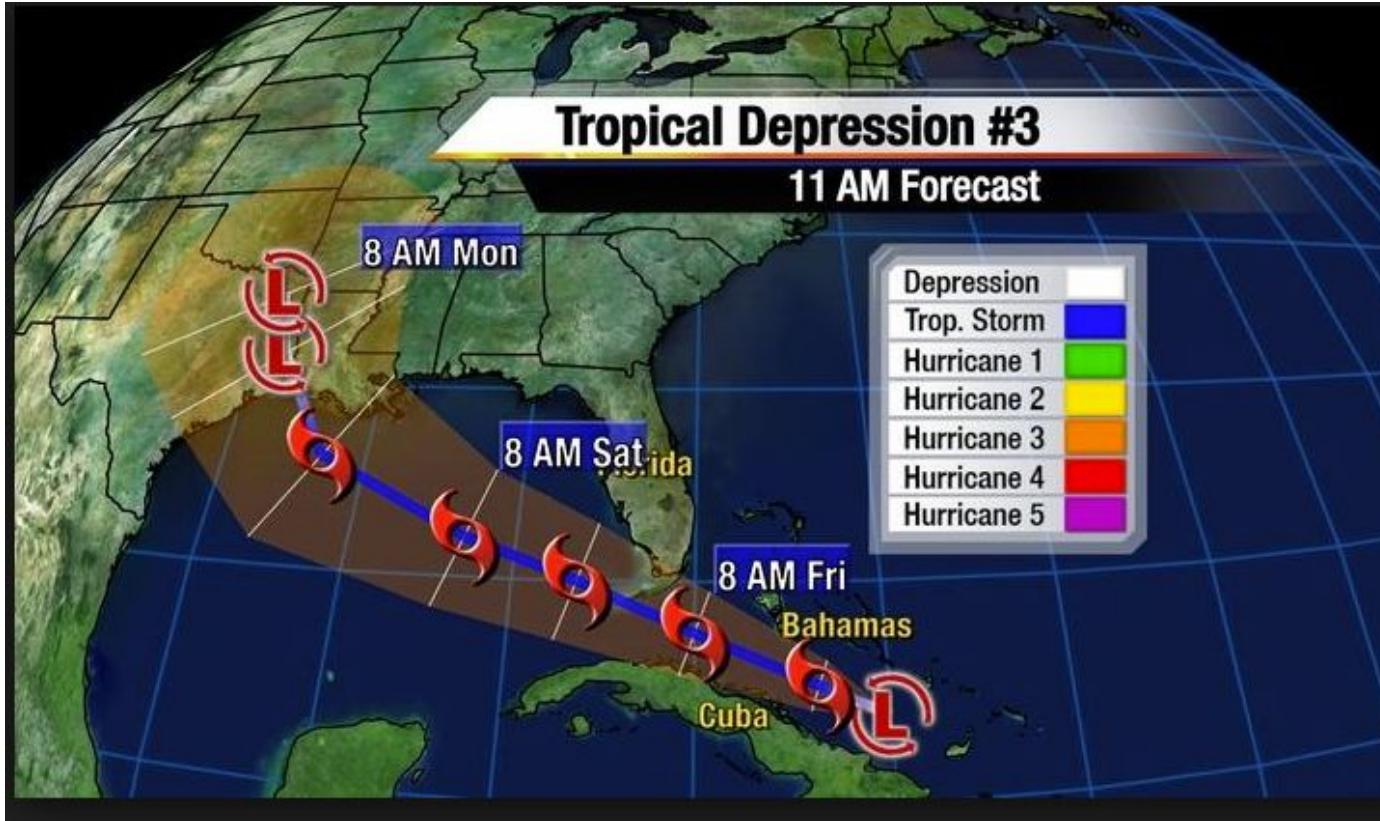


Team and Scope

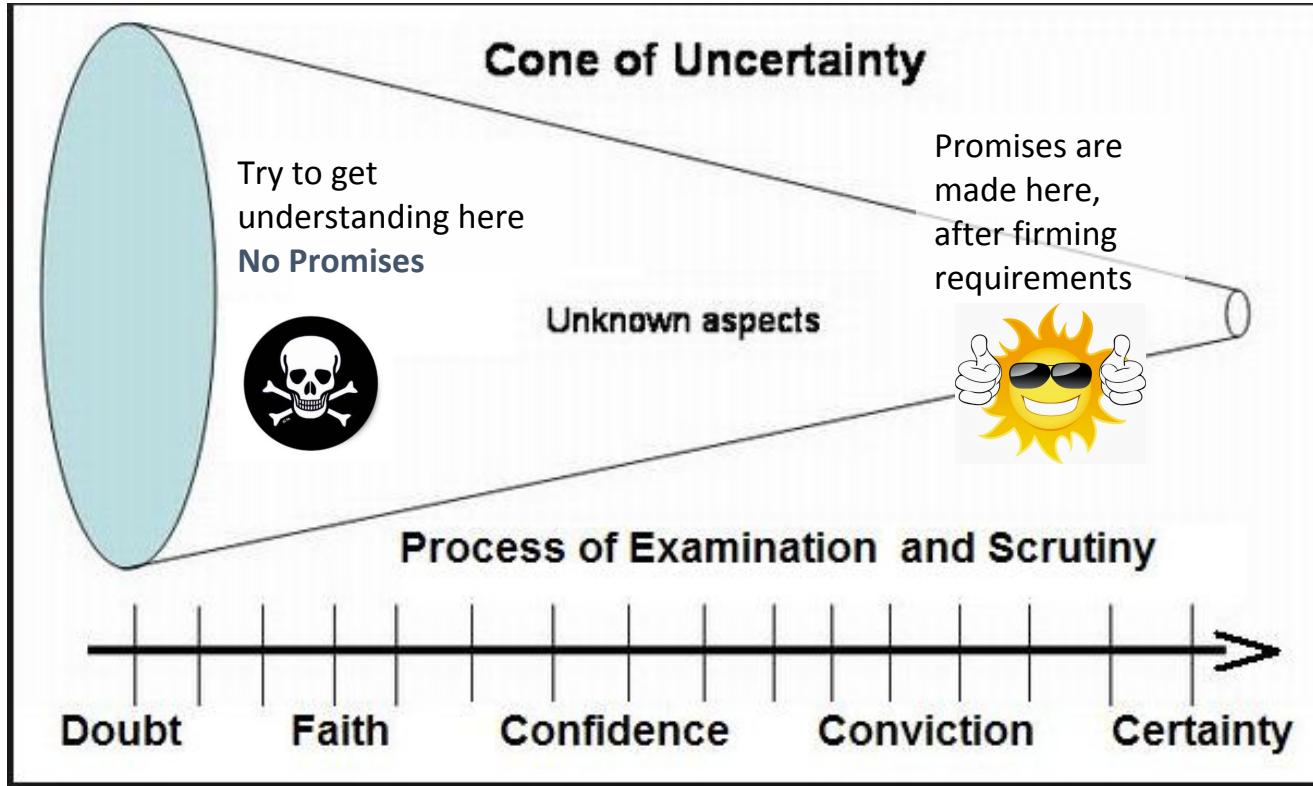
Team Roles

Product Manager	Scrum Master	Development
<ul style="list-style-type: none"> • Owns Product vision • Defines features, decides on release date and content • Responsible for market success • Prioritizes features according to market value • Can change features and priorities every Sprint • Advocate for Business 	<ul style="list-style-type: none"> • Responsible for facilitating process • Focuses Team and protects them from external interruption • Stabilizes Current Sprint • Looks for ways to enhance productivity • Assists Product Owner in leveraging Scrum • Advocate for the Engineer 	<ul style="list-style-type: none"> • Small group containing all necessary people and project skills • Focuses on steady delivery of high quality features • Generates options for delivery • Manages own work within Sprints

Cone of Uncertainty

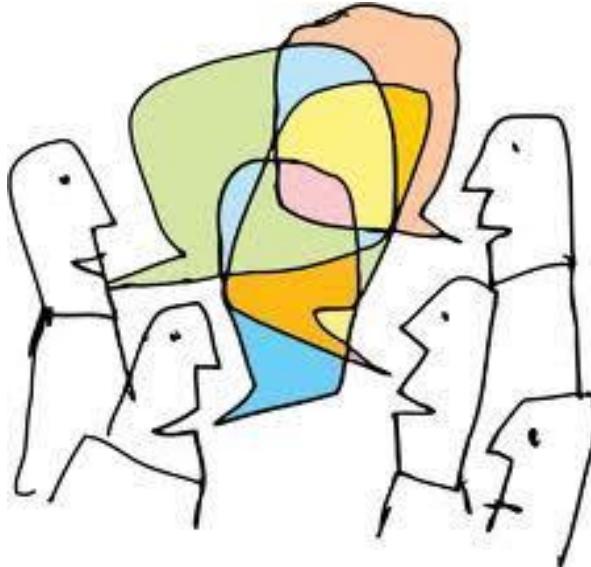
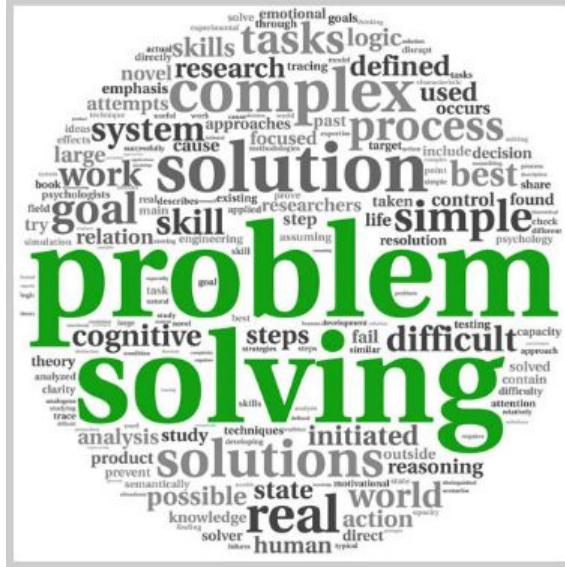


Cone of Uncertainty



User Stories

Helping to solve the Business Problem....



Start with an Epic...



Story Writing

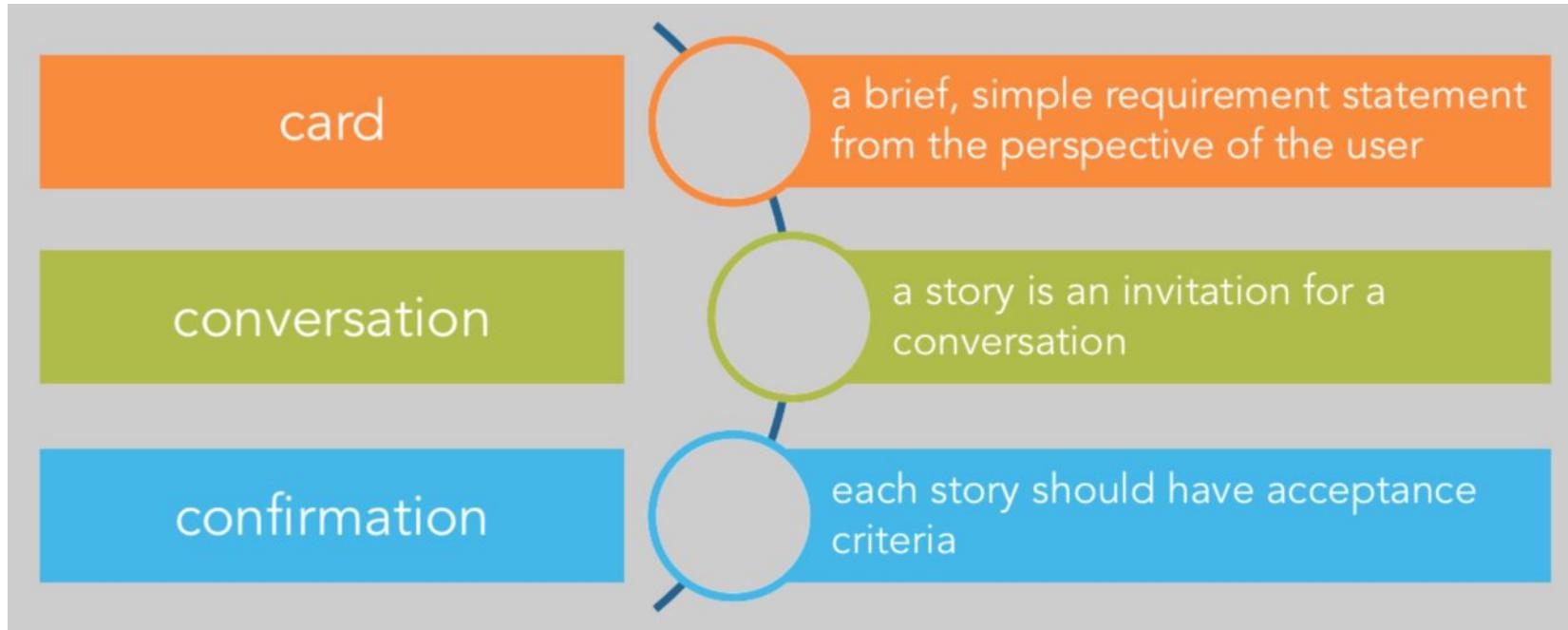
- A user story is not a specification, it is a communication and collaboration tool to help explain the why.
 - *Ron Jefferies - A Card, Conversation and Confirmation.*
- The Product Owner should not hand stories off to the development team:
 - They should rather be part of a conversation together or better yet write them as a team.
 - Share and Leverage each others knowledge and will result in better user stories.



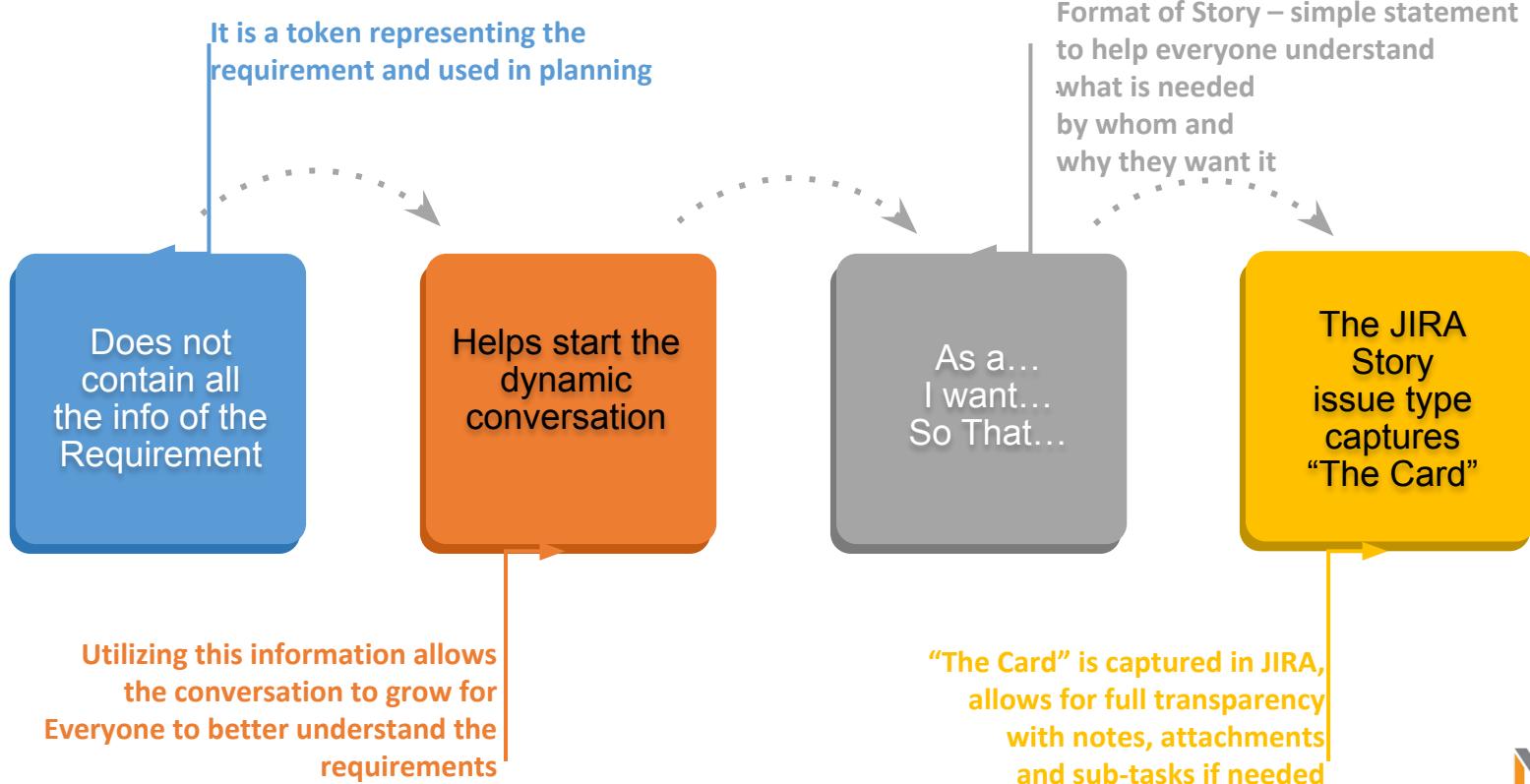
Structure of a Story INVEST

<u>I</u> ndependent	Can deliver value by itself. We want to be able to develop in any sequence.
<u>N</u> egotiable	Avoid too much detail; keep them flexible so the team can adjust how much of the story to implement.
<u>V</u> aluable	Users or customers get some value from the story.
<u>E</u> stimatable	The team must be able to use them for planning
<u>S</u> mall	Large stories are harder to estimate and plan. By the time of sprint planning, the story should be able to be designed, coded and tested within the sprint.
<u>T</u> estable	Document acceptance criteria, or the definition of done for the story which leads to test cases

Structure of a Story



The Card



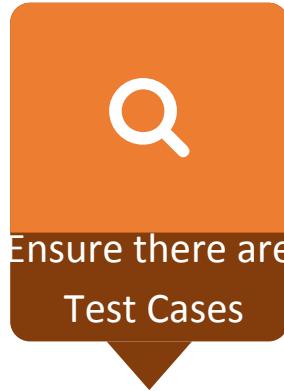
The Conversation



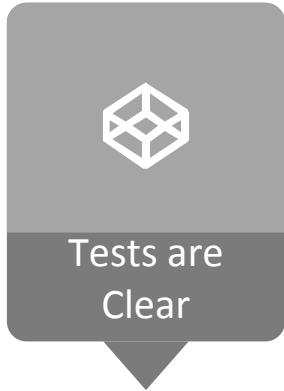
The Confirmation



Story Meets
INVEST



Ensure there are
Test Cases



Tests are
Clear



Estimation is
Completed



Agreement
by All

The story needs to meet the INVEST Criteria that was discussed earlier in the presentation

Having Test Cases will allow the team to understand what needs to be completed and how will everyone understand how they will validate this.

By having clear tests will allow the Developer Test and Customer or User perform UAT

Estimation will allow the team to ensure they understand the size, complexity and effort that is being asked of them. In addition it will allow to see the Burndown and also Velocity of the Team

Agreement by everyone on the team on what will be completed within the sprint will allow everyone to be on the same page.

Story Overview



Story Overview cont.

1st Feedback from the field

“Customers would like the shredder to have a power switch.”



Story Overview cont.

1st Feedback from the field

As a user, I want a power switch so I can turn the shredder off and on.



Story Overview cont.

2nd Feedback from the field

Customers love the power switch but they would like it on the back of the shredder so it cannot be seen from the front.



Story Overview cont.

2nd Feedback from the field

As a user, I want the power switch on the back so it cannot be seen from the front.



Story Overview cont.

3rd Feedback from the field

Customers love the new switch location, but they would like a light on the front that indicates if the shredder is powered on.



Story Overview cont.

3rd Feedback from the field

As a user, I want an indicator light so I know if the shredder is on.



Story Overview cont.

Why does the customer want . . .

- A power switch?
- The power switch on the back of the shredder?
- A power indicator light?



Story Overview cont.



Story Overview cont.

Original user story

As a user, I want a power switch so I can turn the shredder off and on.



Story Overview cont.

Better Story

As a parent, I want a shredder on which my toddler will not accidentally hurt herself so that I do not have to remember to unplug it from the wall when not in use.



Estimating

Agile Estimation (Fibonacci) and Planning

- Fibonacci Numbers
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,
 - We use the following numbers in Agile
 - 0, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?

0	1	2	3	4	5	6	7	8	9	10
0	1	2	3	5	8	13	20	40	100	?

- Let's use Fibonacci and explain how we can use these for sizing stories



Backlogs

Product Backlog is DEEP

Goal - Groom user stories in the backlog for future sprints - “story time” (get them to a “ready” state)

Detailed Appropriately 

 Estimated

Emergent 



Prioritized

Detailed Appropriately

More detail is known about items that will be done sooner and are ordered higher in the backlog

Emergent

Not everything is known upfront. We acknowledge that change happens, so planning is continual

Estimated

Some form of estimate has been given for each product backlog item before a team commits to implement

Prioritized

The backlog is prioritized, so that stories are ordered by value to the business continually as learning progresses

Definition of Ready and Done



Example - Definition of Ready

Definition of Ready (DOR) – checked prior to starting a story in a sprint

Description and Acceptance Criteria are clearly defined

UX Documentation has been reviewed and approved by Team

All functional requirements, non-functional requirements, documents, designs and use cases have been included or linked in ticket

Dependencies are identified and documented or linked in ticket

Ticket must be 100% filled out where required by JIRA

This is suggested DoR.

Each Team needs to define their own DoR.



Definition of Done



Example - Definition of Done

Example – Definition of Done

Product Owner accepts the delivered story and signs off

Example Definition of Done (DoD)

The story meets all listed Acceptance Criteria

The story visually matches UX documentation (if applicable)

The code for the story is reviewed, suggestions incorporated and checked in

Unit tests and test cases are written and executed successfully

The code has been merged to master, and pushed to source control

Product owner has reviewed the story to validate that it meets the Acceptance Criteria



This is suggested DoD.

Each Team needs to define their own DoD.

Scrum Framework - Artifacts

Product Backlog

All work items related to a product/project, ordered by a Product Owner.

Burndown Chart

Visual aid for tracking team progress and forecasting expected completion dates.

Sprint Report

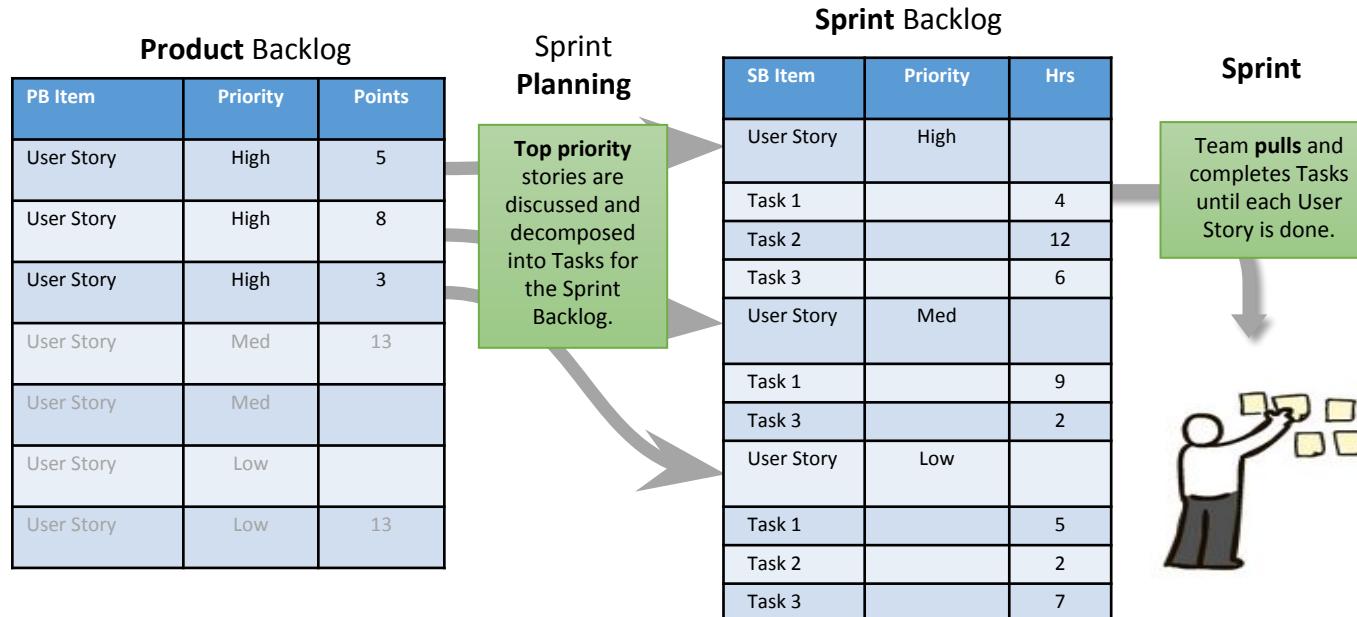
The Sprint Report shows the list of issues in each sprint. It is useful for your Sprint Retrospective meetings, and also for mid-sprint progress checks..

Velocity Chart

Tracks rate of completion for team. Shows the sum of promises per sprint vs. the work that was delivered

Product and Sprint Backlog

- Priority is looked at from the Product Backlog and moved to the Sprint Backlog
- Some teams look at hours at the task level, others only look at Story Points of stories



How to read Burndown Chart

- Visual aid for tracking team progress and forecasting expected completion dates.
- Burndown Charts can provide context to make tough decisions for sprints



Backlog

Story 1
Story 2
Story 3
Story 4
Story 5
Story 6
Story 7
Story 8
Story 9
Story 10

Product Owner Speaking

To date, we have completed Story 1 through Story 4.

Unfortunately, we lost several key members of our team during iteration 6 and **we are unlikely to get all planned Stories done for this release**, unless we execute with perfection.

We will likely **delay Story 9 and 10 until the next release**, unless we make some tradeoffs.

We already started discussions with sales and marketing and we may limit our work on Story 5 and 6 in the next Sprint.

JIRA Sprint Report

The Sprint Report shows the list of issues in each sprint. It is useful for your Sprint Retrospective meetings, and also for mid-sprint progress checks.

ATT_Scrum
Sprint Report -

ATT 7/19 - 7/31 It 62 ▾ [How to read this chart](#)

Active Sprint 27/Aug/18 8:56 PM - 10/Sep/18 8:56 PM [Linked pages](#)

[View ATT 7/19 - 7/31 It 62 in Issue Navigator](#)

1h

0

Aug 27 Sep 10

Status Report

* Issue added to sprint after start time

[View in Issue Navigator](#)

Completed Issues

Key	Summary	Issue Type	Priority	Status	Story Points (3)
ATT-89	Pick a theme for the party	Story	<input type="checkbox"/> None	DONE	3

[View in Issue Navigator](#)

Issues Not Completed

Key	Summary	Issue Type	Priority	Status	Story Points (5)
ATT-88	Delegate Responsibilities	Story	<input type="checkbox"/> None	BACKLOG	3
ATT-183 *	Determine Theme of Party	Story	<input type="checkbox"/> None	BACKLOG	2

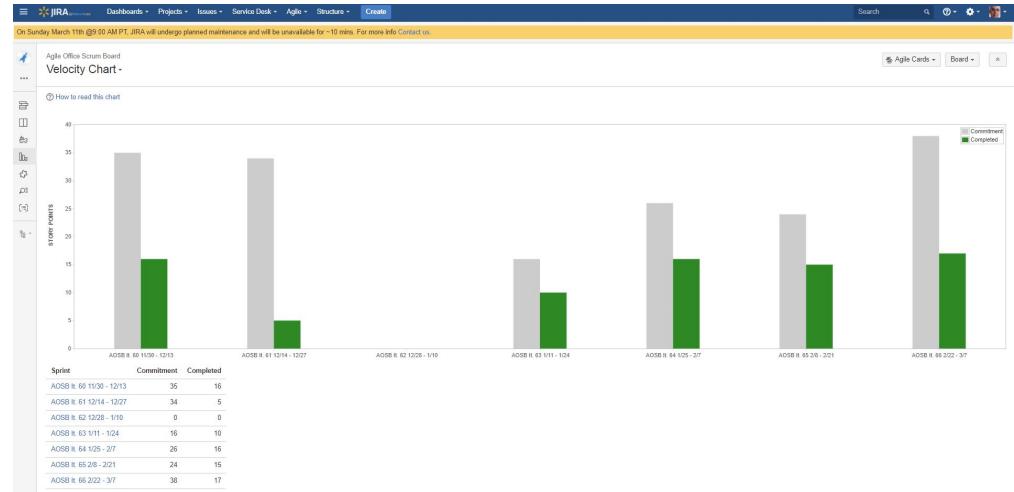
[View in Issue Navigator](#)

Issues Removed From Sprint

Key	Summary	Issue Type	Priority	Status	Story Points (5)
ATT-32	Determine core group of people	Story	<input type="checkbox"/> None	BACKLOG	5

Using Velocity for Planning

- Velocity is a measure of how much work can be delivered in a timebox
- It is based on the historical performance of the team for recent sprints
- We use established velocity for planning sprints and for use in estimating releases



Retrospective

Retrospectives = Focus?



Retrospective – Purpose and When

The PAST is where you learned the lesson
The FUTURE is where you apply the lesson,

Don't GIVE UP
in the middle!



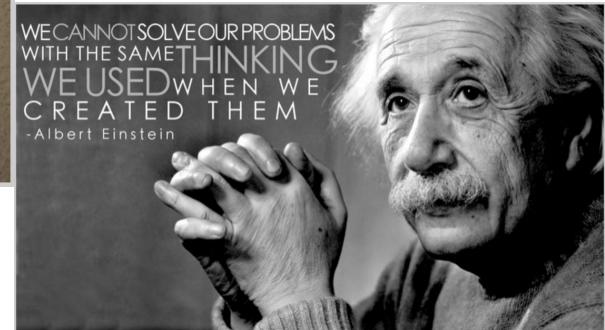
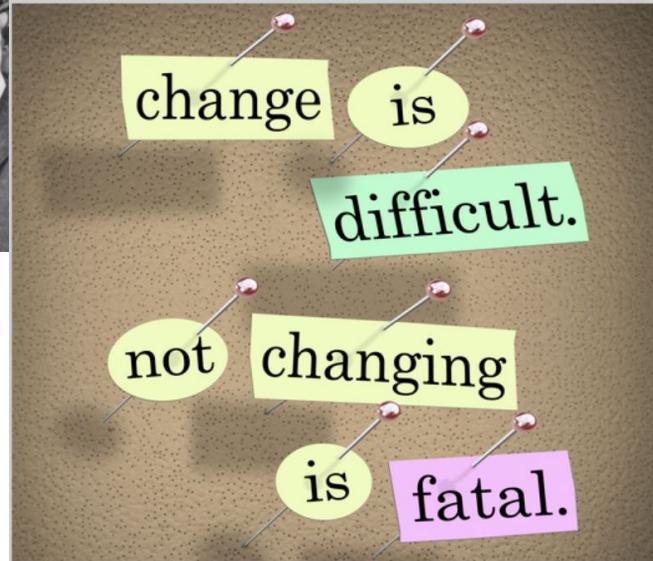
Why do Retrospectives

**"You may never
know what results
come from
your action.
But if you do
nothing,
there will be no
result."**

Mohandas Karamchand Gandhi

happytoinspire

Mindset during Retrospective

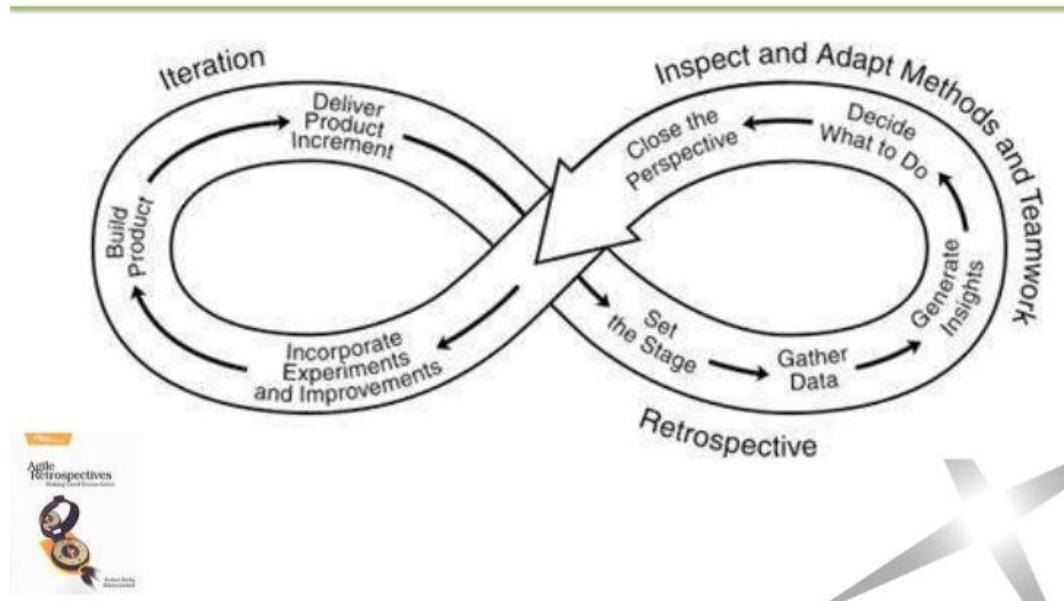


Tip prior to Retrospective



Making good teams Great!!!

Why Retrospect



Credit: Agile Retrospectives – Making good teams Great by Esther Derby & Diana Larsen

<http://pragprog.com/book/dlret/agile-retrospectives>

AgileSparks

Structuring the Retrospective – 6 points

- Open Retrospective
 - Set the Stage
 - Gather Data
 - Generate Discussion
 - Prioritize and Decide What to Do
 - Close Retrospective
-
- Below is timeline of meeting:



Open
Retrospective

Set the
Stage

Gather
Data

Generate
Discussion

Prioritize
and Decide
What to Do

Close
Retrospective

Gathering Data – Learning Matrix



Happy Face – aspects you liked
should be repeated



Sad Face – aspects you disliked –
should be changed

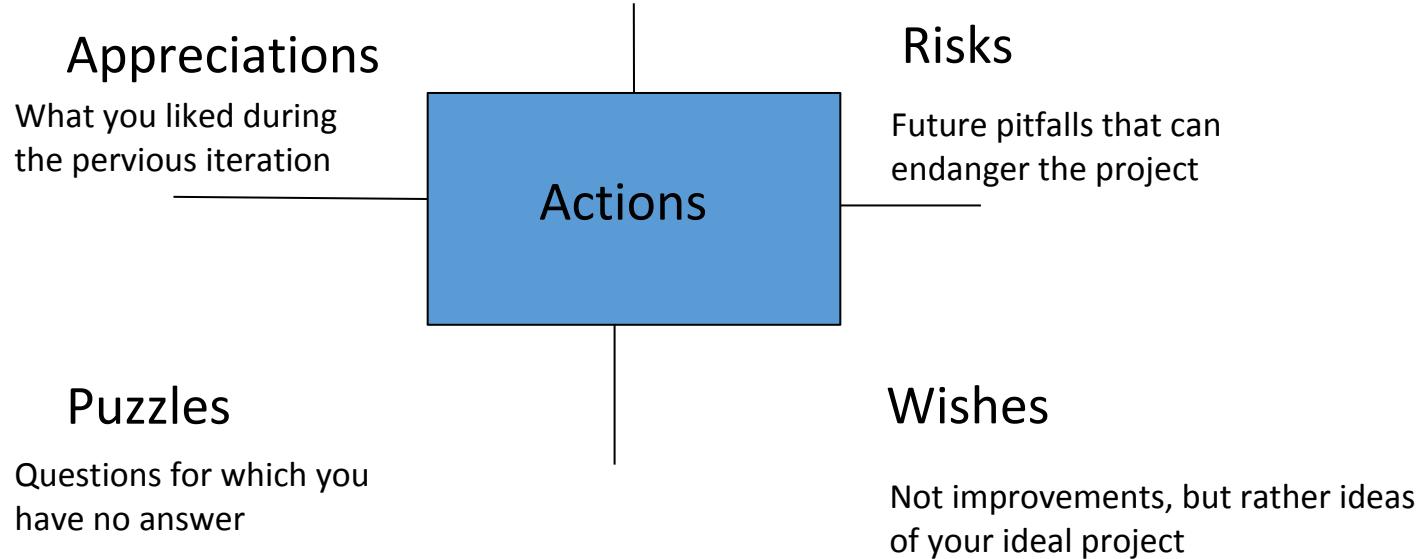


Flower bouquet for people
you wish to appreciate

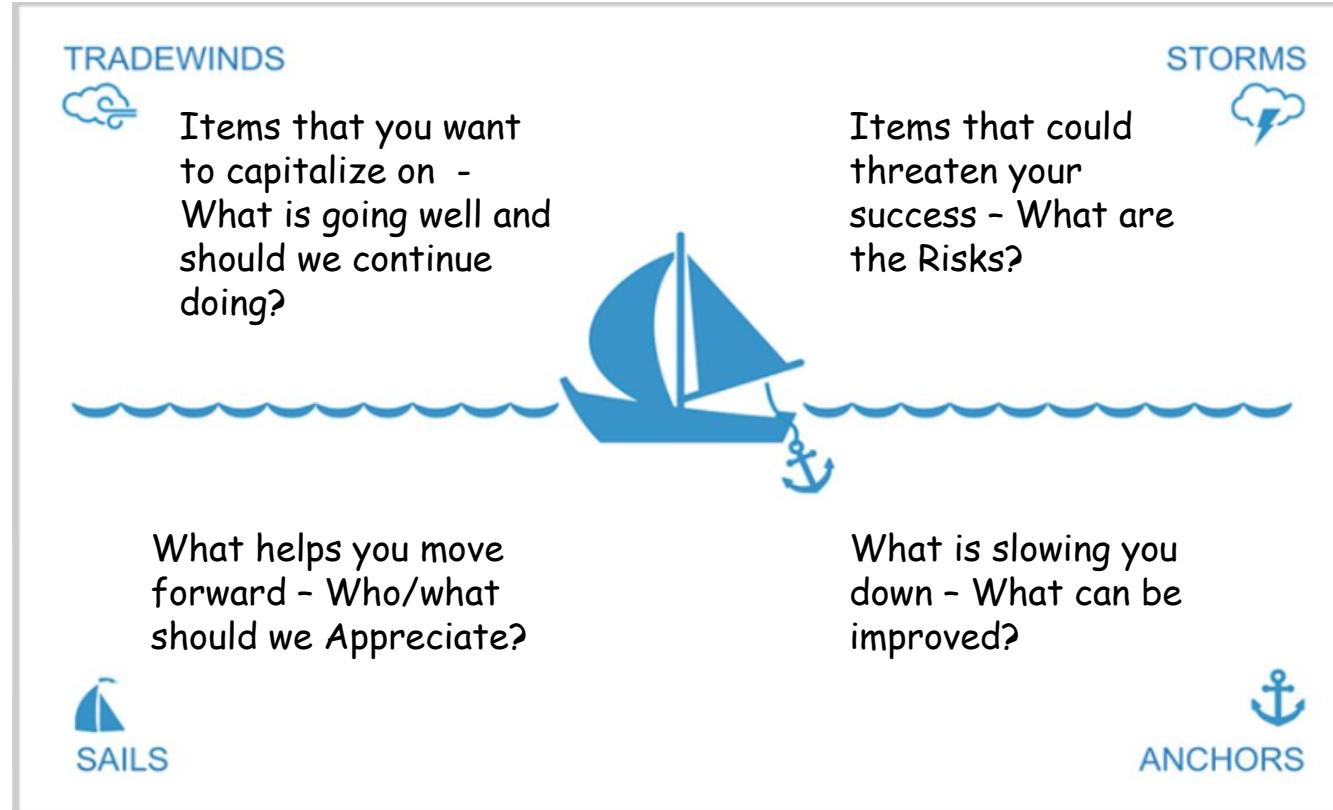


Light bulb for new ideas to try

Gathering Data - Actions



Gathering Data - Sailboat



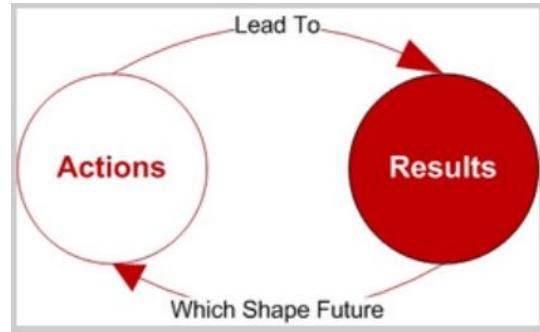
Gathering Data – Start, Stop, Continue



- Simplest retrospective method – very effective
- Reflect on three things
- Facilitator has team focus on each question separately in 3 short “rounds”
- Team votes on specific items to focus on during the coming sprint
- Review these items at the beginning of next retrospective

Prioritize and Close Retrospective

Reiterate actions and follow-up



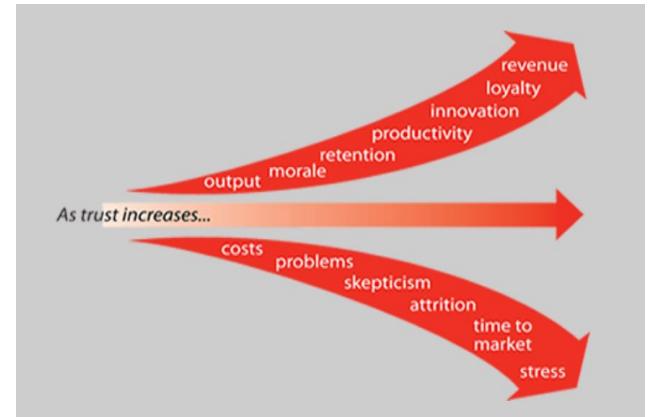
Appreciate contributions



Identify ways to make the next retrospective better

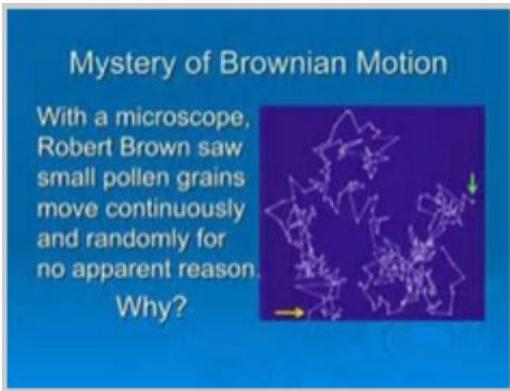


Benefits of doing Retrospective



Leave out...what happens

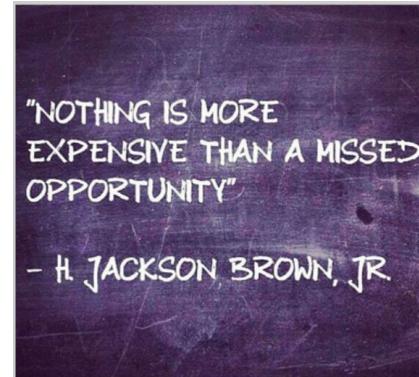
Setting the Stage:



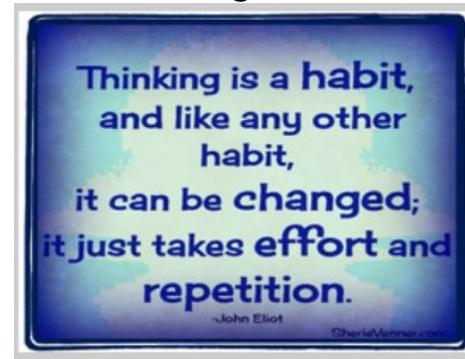
Gathering Data:



Closing the Retrospective:



Generating Discussion:



Deciding What to Do:



How do we adopt this?

How

Training

Coaching

Mentoring

Shu

Person follows rules
until they sink in

Ha

Person reflects on the
rules,, looks for
exceptions and
“breaks the rules

Ri

Rules are forgotten as
person has developed
mastery, and grasped the
essence and underlying
forces

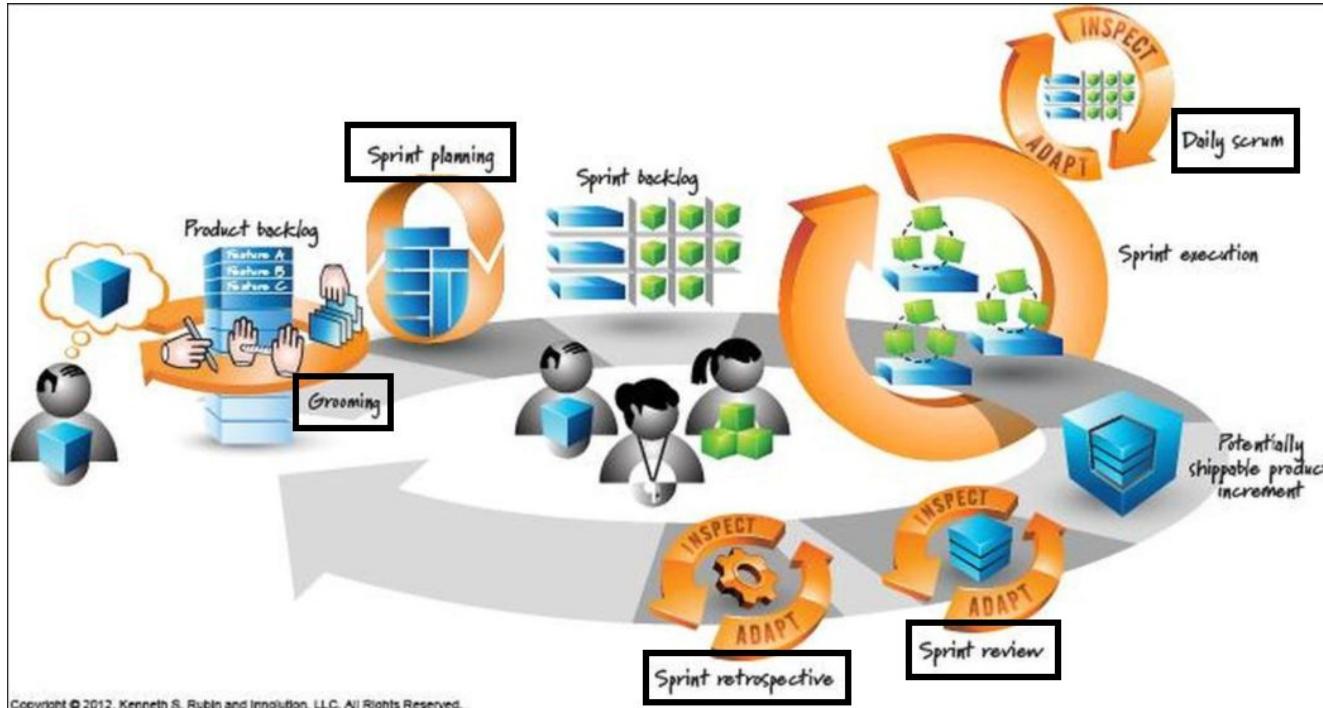
Who

Student

Apprentice

Master

Better Understanding - Scrum Framework



Thanks

Thanks

Topics

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



Topics

Jira Overview

Projects, issues, boards and user types



What is 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*



Issues

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)



iOS Project



HR Onboarding

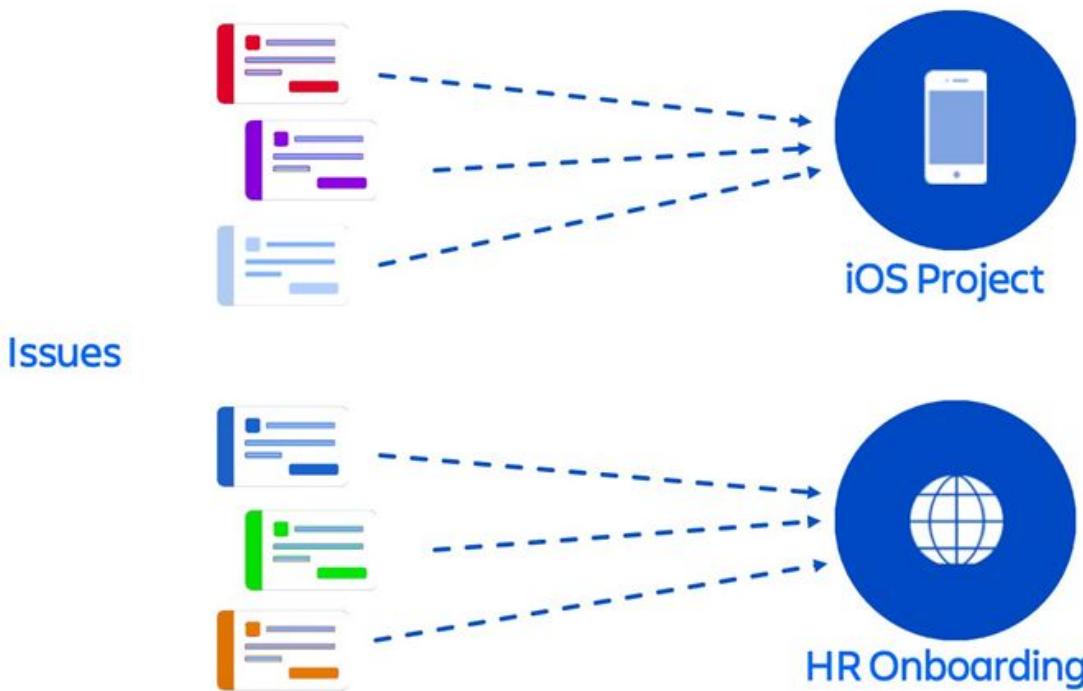
Issue key



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

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

Each issue belong to one project



What is 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

- Jira is a tool teams use to manage and visualize work
- A Jira issue is an 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 - Jira Overview

- Create a project
- Create issues

Visualize Work Using Project Boards



What will you learn?

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

Topics

Visualizing work

Workflows

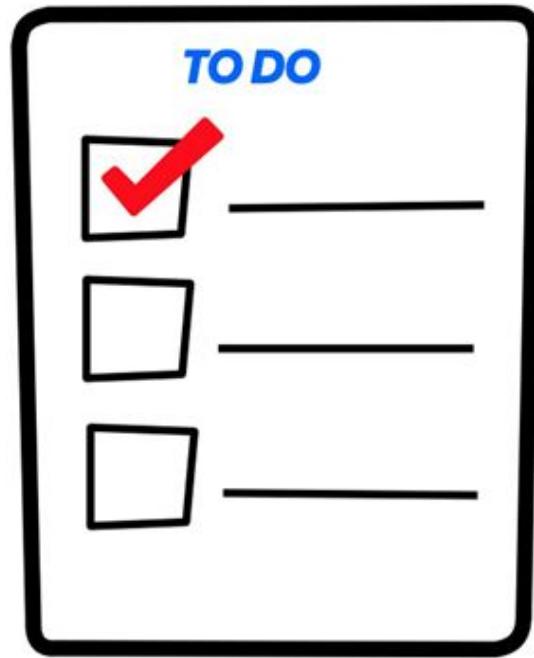
Jira boards and Workflows

Configuring board columns



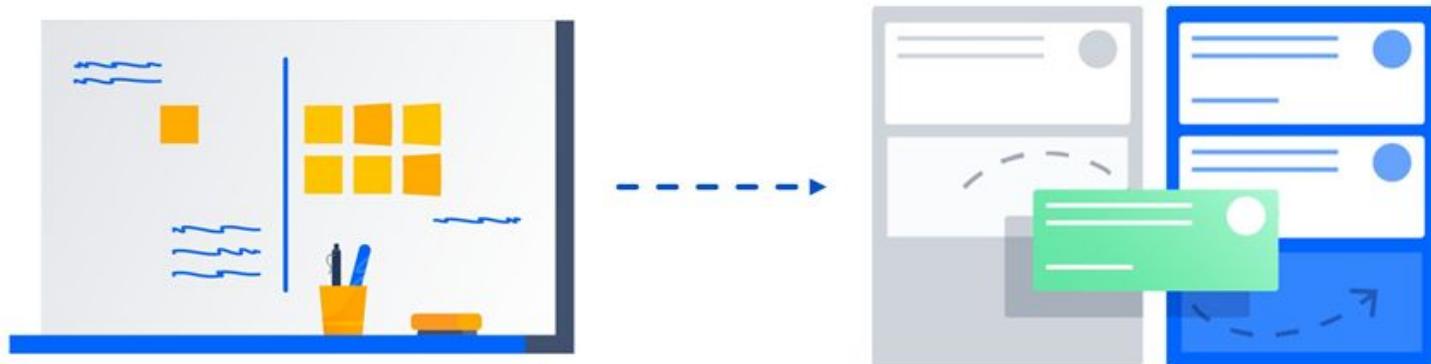
Visualizing work: a “to do” list

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



Visualizing work: a board

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



Visualizing work: reports and dashboards



Reports

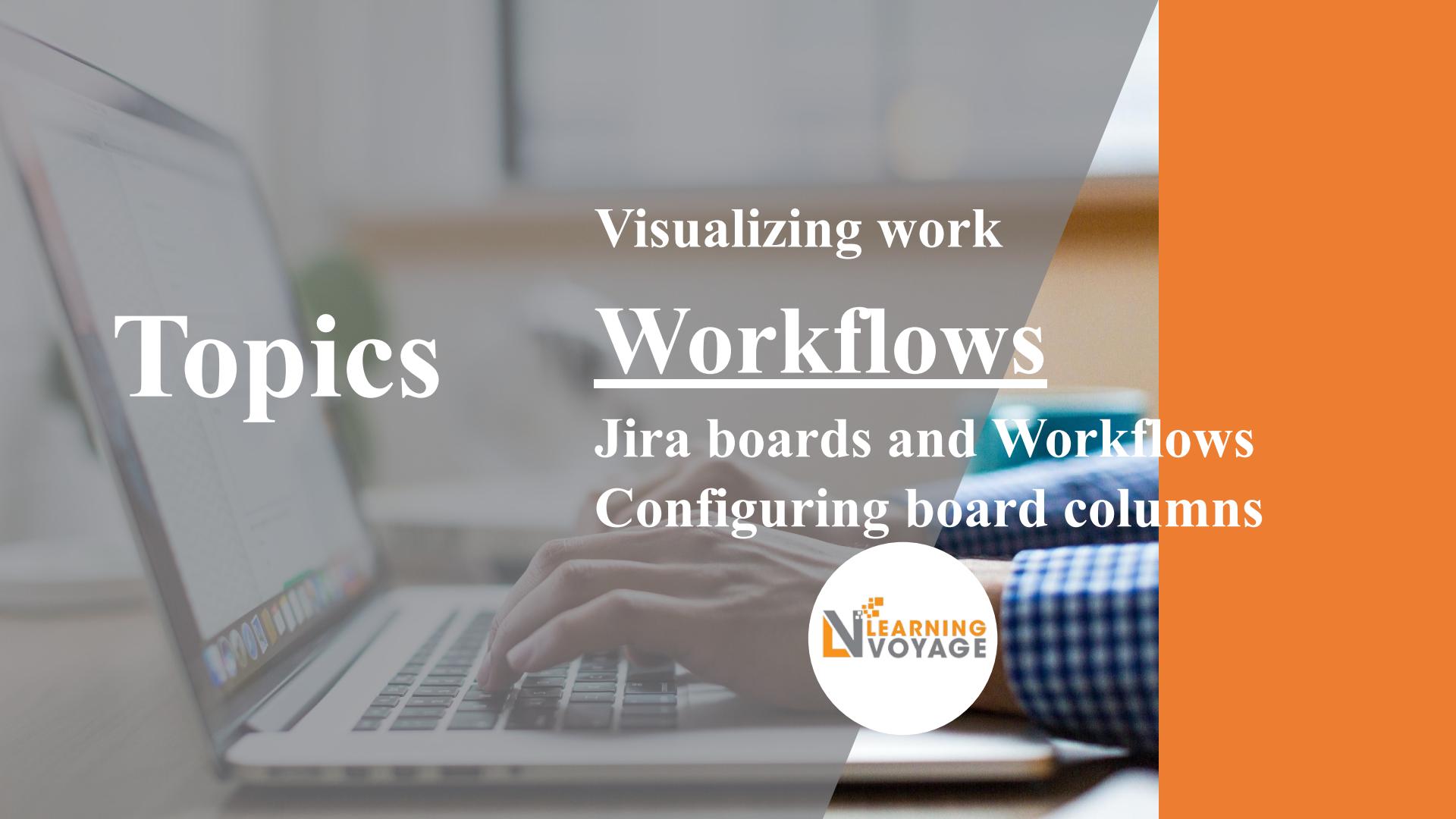


Dashboards

Why visualize work?

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



A blurred background image showing a person's hands typing on a laptop keyboard. The laptop screen is visible on the left, displaying some text. The overall color palette is warm and professional.

Topics

Visualizing work Workflows

Jira boards and Workflows
Configuring board columns

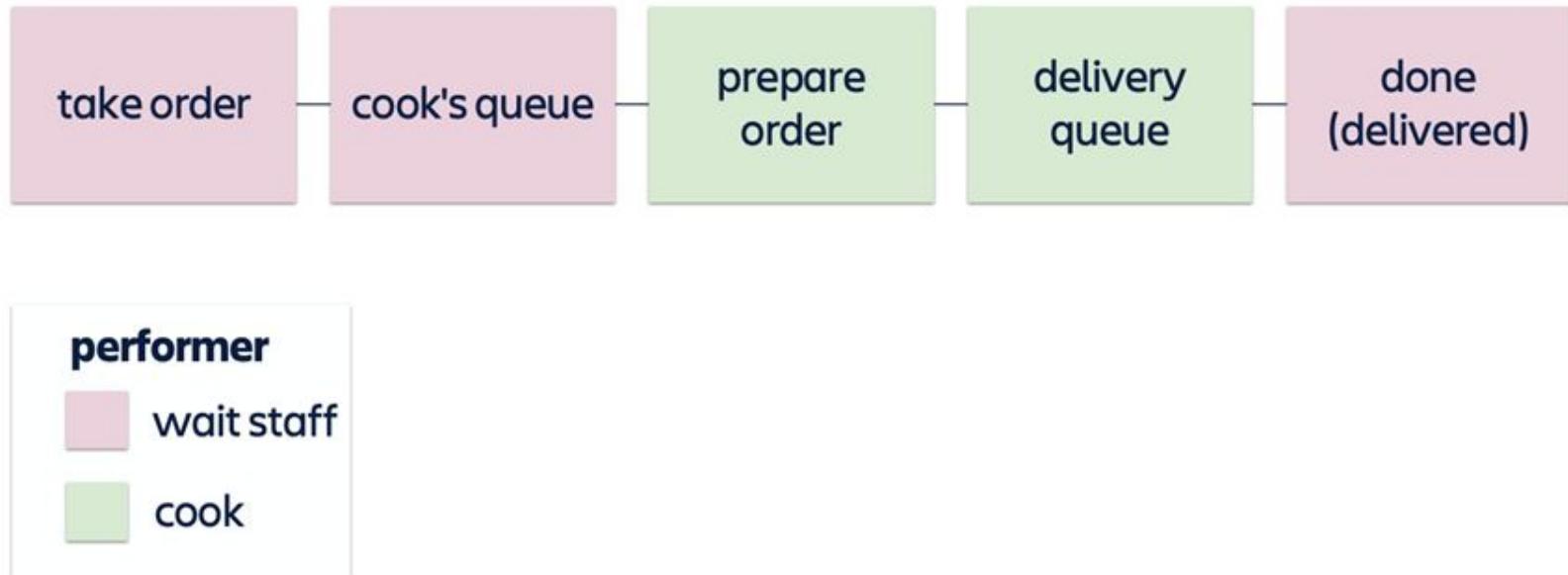


Workflows

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

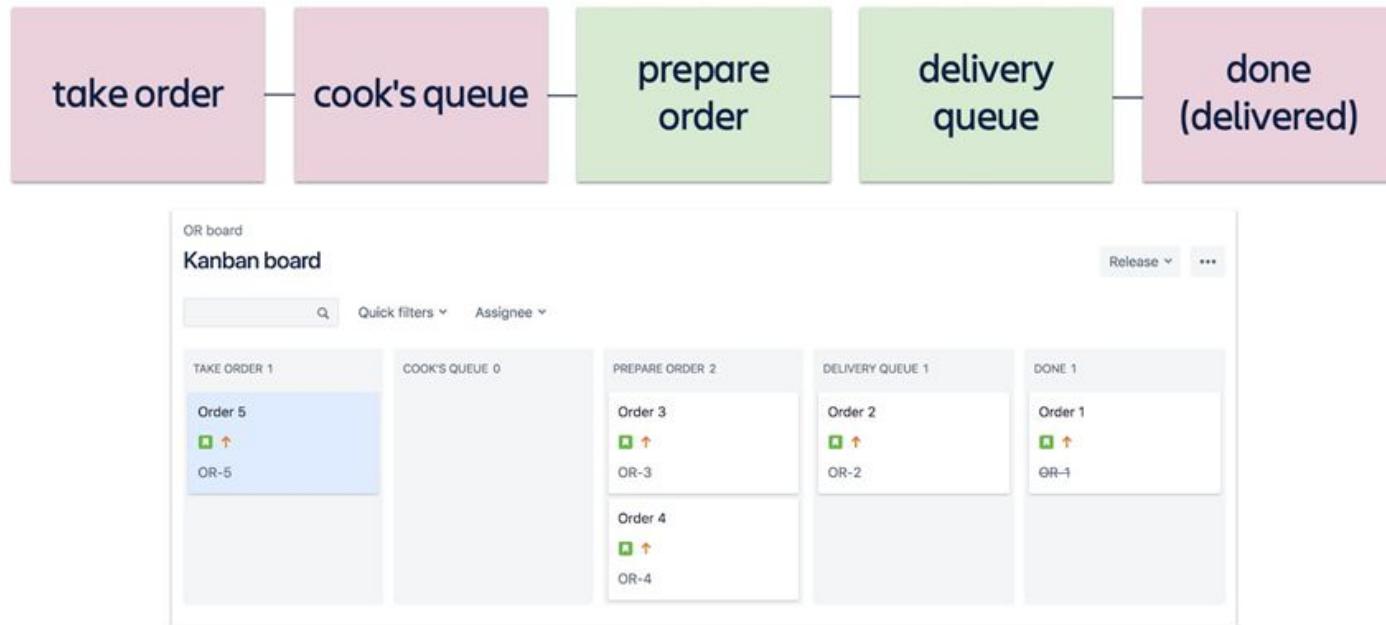
The screenshot shows a Kanban board interface titled "Kanban board". At the top, there are navigation links: "Projects / projectA / PROJ board", a search bar, and user filters "Only My Issues" and "Recently Updated". The board has four columns: "BACKLOG 3", "SELECTED FOR DEVELOP... 0", "IN PROGRESS 0", and "DONE 0". The "BACKLOG" column contains three items: "add feature 1" (PROJ-1), "add feature 2" (PROJ-4), and "add feature 3" (PROJ-5). Each item has a small icon and a red vertical line to its left.

Example Workflow: restaurant order & delivery



board vs Workflow

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



Topics

Visualizing work
Workflows
Jira boards and Workflows
Configuring board columns



How are boards created?

- Automatically
- Create additional boards at any time

The screenshot shows a Jira Kanban board for projectA / PROJ board. The board has three columns: BACKLOG (3 items), SELECTED FOR DEVELOPME... (0 items), and IN PROGRESS (0 items). The first item in the backlog is "add item 1". A context menu is open in the top right corner, with a red arrow pointing to the "Create board" option. The menu also includes "Board settings", "Hide menus", "Show detail view", "Open issues in sidebar", and "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 is a checkbox labeled "PROJ-1". To the right are "Give feedback" and "1" buttons, along with a share icon. Below this is a header with "add item 1" and several action buttons: "Attach", "Create subtask", "Link issue", and an ellipsis button. On the right is a status dropdown menu with three items: "Backlog", "SELECTED FOR DEVELOPMENT", "IN PROGRESS", and "DONE". A large red arrow points from the text "The status field for each issue must be set to a workflow's status" towards the "Backlog" option in the dropdown menu.

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 DEVELOPMEN... 0	IN PROGRESS 0	DONE 0
<p>add item 1</p> <p><input checked="" type="checkbox"/> </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><input checked="" type="checkbox"/> </p> <p>PROJ-2</p>			
<p>add item 3</p> <p><input checked="" type="checkbox"/> </p> <p>PROJ-3</p>			

Topics

Visualizing work

Workflows

Jira boards and Workflows

Configuring board columns



Adding a board columns

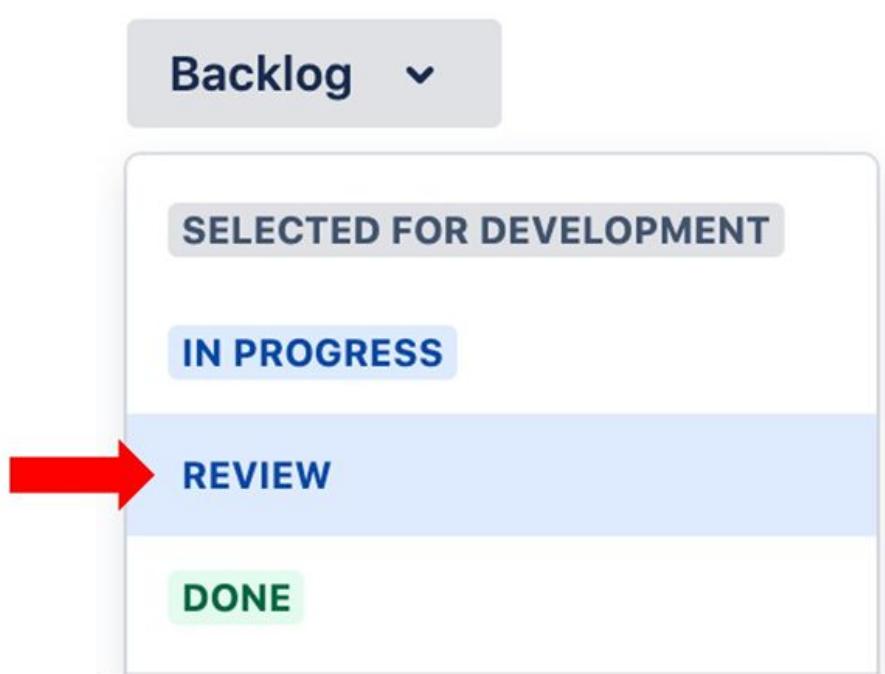
Projects / projectA / PROJ board

Kanban board

The screenshot shows a Jira Kanban board for the 'PROJ board' in the 'projectA' project. The board has five columns: 'BACKLOG 3', 'SELECTED FOR DEVE... 0', 'IN PROGRESS 0', 'REVIEW 0' (which is highlighted with a red border), and 'DONE 0'. The 'BACKLOG' column contains three items: 'add item 1', 'add item 2', and 'add item 3', each associated with a project identifier (PROJ-1, PROJ-2, PROJ-3) and a checkmark icon. The 'REVIEW' column has a note: 'We're only showing recently modified issues.' and a link 'Looking for an older issue?'. The top navigation bar includes links for 'Only My Issues' and 'Recently Updated', along with icons for edit, star, release, share, and more.

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

Viewing the new status



Takeaways

- A board is a two-dimensional way to visualize the work of 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 to the 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

Enriching 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

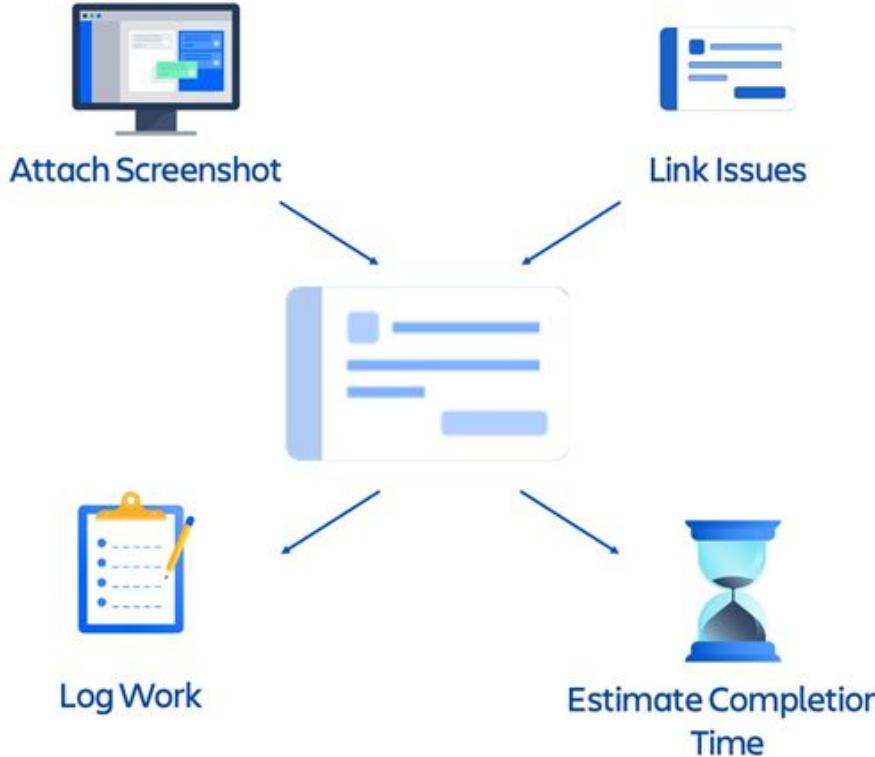
Reporter:



Oliver

Comments: Helena needs the network diagram from IT.

Enriching issues



Mention team members



Topics

Enriching issues

Issue types

Labels

Developer Integration overview



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)

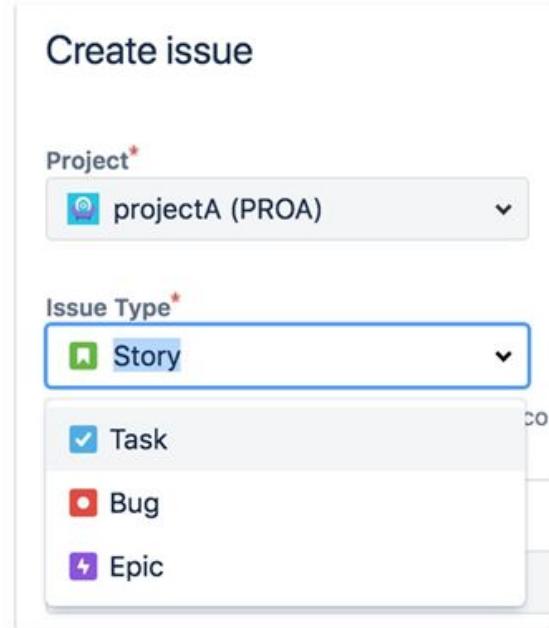
Issue Type*

Story

Task

Bug

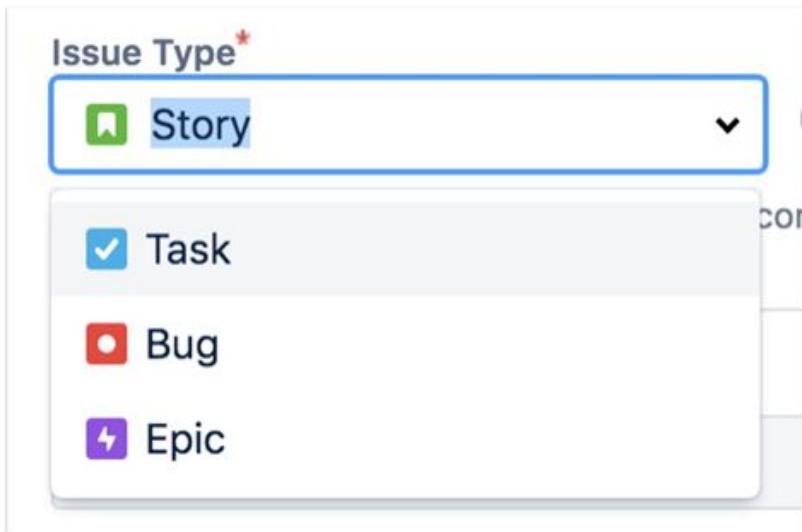
Epic



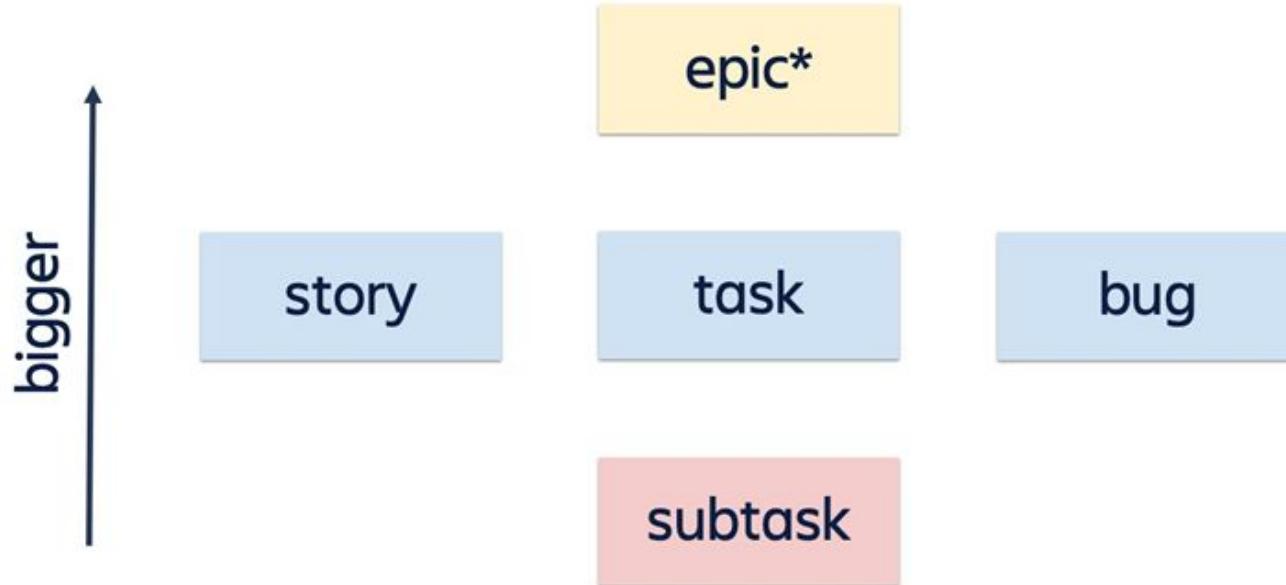
A project can also use custom issue types

Why Issue types?

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



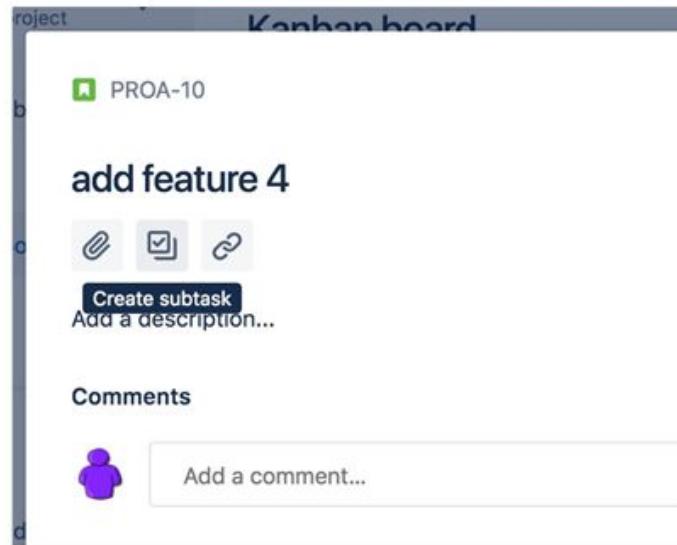
Jira's issue type hierarchy



*Epics are discussed later

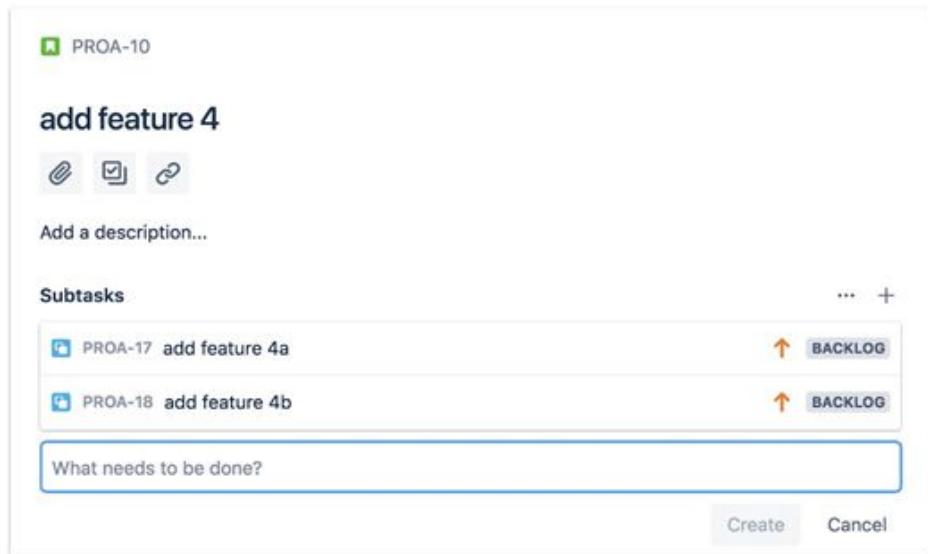
Sub tasks

- 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



Topics

Enriching issues

Issue types

Labels

Developer Integration overview



Labels

A field used to categorize and search for issues

The image shows a Jira issue card for 'PROJ-6' titled 'complete task 1'. The card includes fields for description, subtasks, comments, and status. A callout box highlights the 'Labels' section, which contains 'refactor' and 'database'. Another callout box highlights the 'Status' field, which is set to 'Selected for Development'.

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

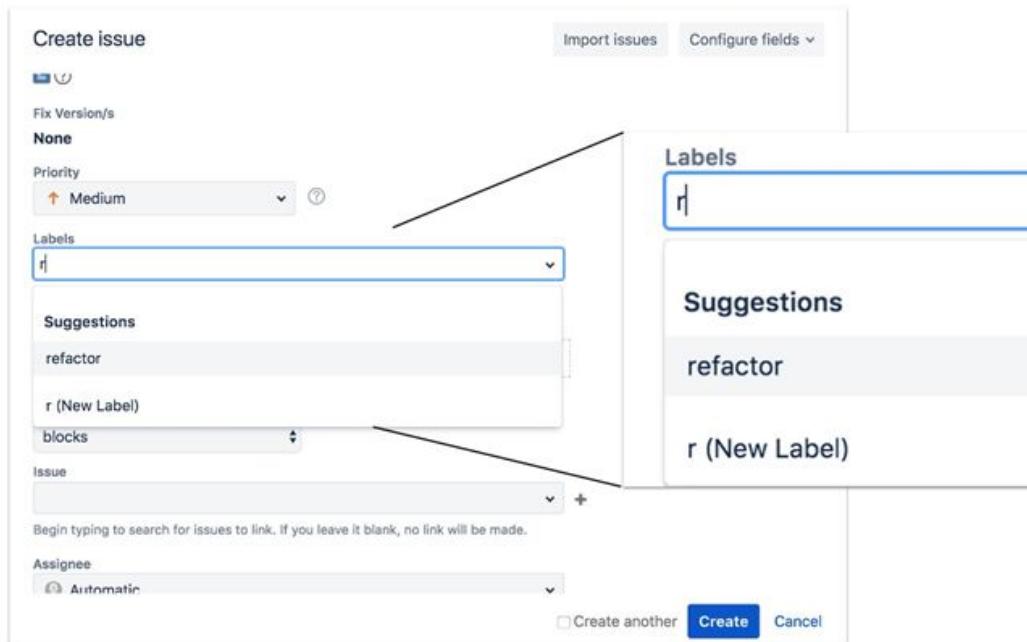
Created 6 days ago

Updated 1 hour ago

Labels

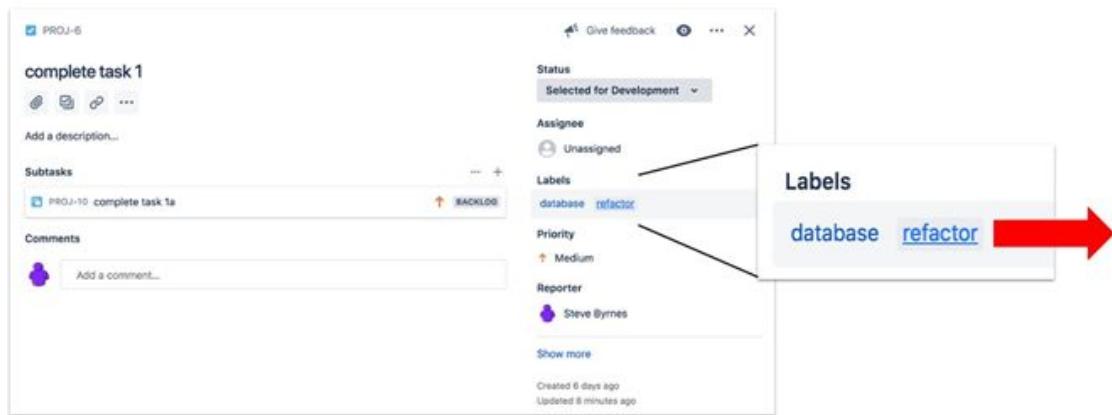
refactor database

Adding or creating a labels



Searching for Labeled issues

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



A screenshot of the Jira search interface. The search query 'labels = "refactor"' is entered in the search bar. The results show two issues: 'PROJ-10 PROJ-6 / complete task 1a' and 'PROJ-6 complete task 1'. Both issues have a checked checkbox next to them. A red arrow points from the 'refactor' label in the previous screenshot to the 'refactor' label in the search results.

Topics

Enriching 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 projects



What will you learn?

- Describe the kanban method
- Describe the Importance of flow
- Identity the purpose of work-In-progress limits
- Identify reasons to separate the backlog from the board

Topics

Kanban Method Overview

Improving flow

Separate Backlog



What is Kanban Method ?

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

[https://en.wikipedia.org/wiki/Kanban_\(development\)](https://en.wikipedia.org/wiki/Kanban_(development))

Topics

Kanban Method Overview
Improving flow
Separate Backlog

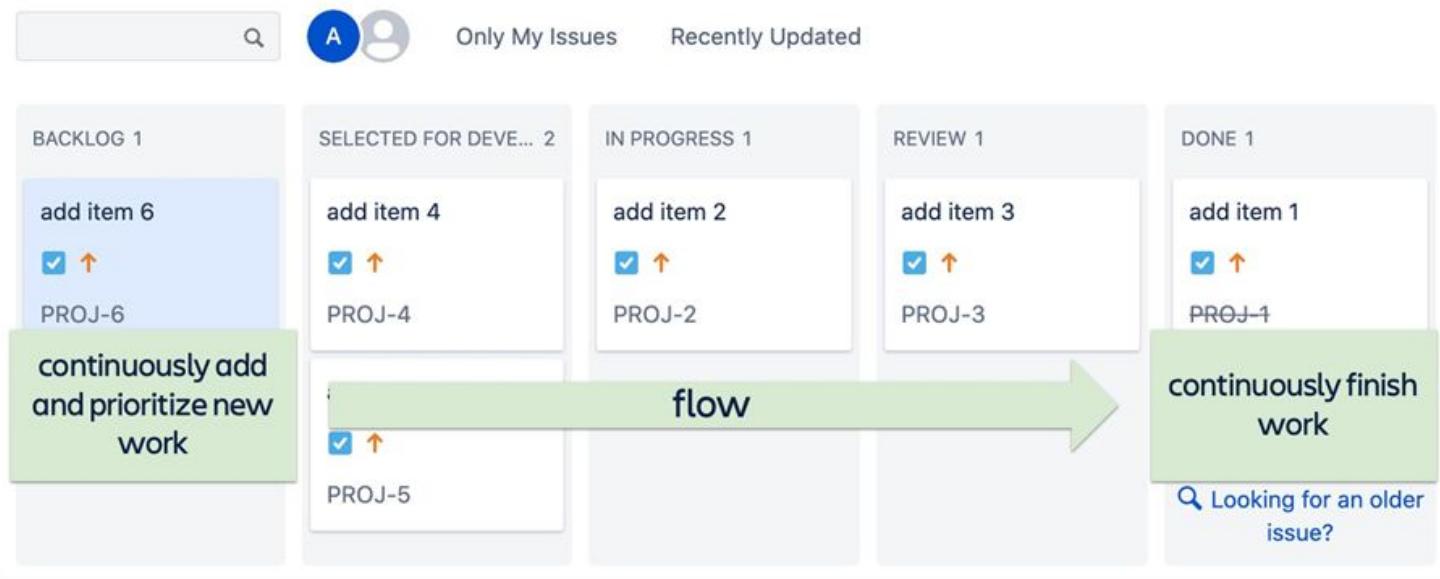


Continuous flow of work items

Projects / projectA / PROJ board

Kanban board

Release    ...



Improving flow-limit work in progression(WIP)

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

<https://www.atlassian.com/agile/kanban/wip-limits>

Column under minimum limit

Projects / projectA / PROJ board

Kanban board

Release

Only My Issues Recently Updated

BACKLOG 2	SELECTED FOR DEVE... 1 Min 2	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
add item 6 PROJ-6				We're only showing recently modified issues. Looking for an older issue?

A red arrow points from the 'IN PROGRESS' column to the 'SELECTED FOR DEVE...' column.

Column over maximum limit

Projects / projectA / PROJ board

Kanban board

The screenshot shows a Jira Kanban board for projectA. The columns are labeled BACKLOG 1, SELECTED FOR DEVEL..., IN PROGRESS 0, REVIEW 3, and DONE 0. A red arrow points down to the REVIEW 3 column, which contains three items: add item 1, add item 2, and add item 3. Above the third item, there is a label "Max 2". The REVIEW 3 column is highlighted with a red border.

BACKLOG 1

SELECTED FOR DEVEL... 2
Min 2

IN PROGRESS 0

REVIEW 3 Max 2

DONE 0

We're only showing recently modified issues.

Looking for an older issue?

Column	Items
BACKLOG 1	add item 6 PROJ-6
SELECTED FOR DEVEL...	add item 4 PROJ-4
IN PROGRESS 0	
REVIEW 3	add item 1 PROJ-1 add item 2 PROJ-2 add item 3 PROJ-3
DONE 0	

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

The Kanban board displays the following data across five columns:

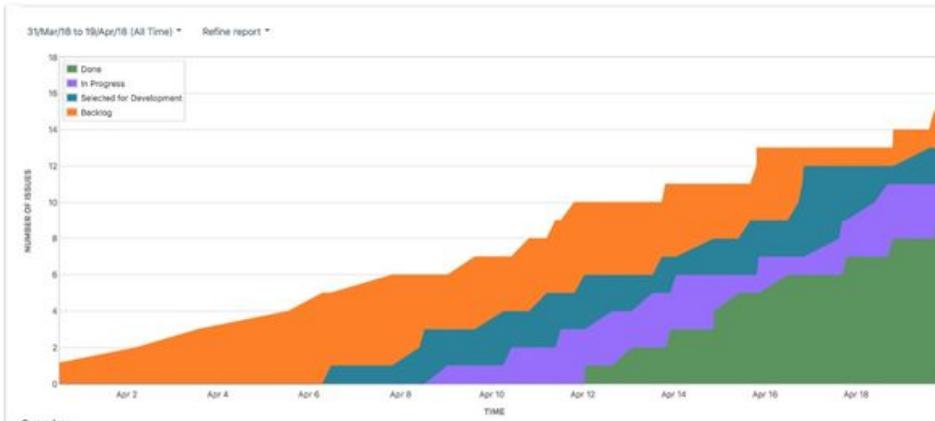
Column	Items	Count	Max
BACKLOG	add item 6 PROJ-6	1	
SELECTED FOR DEVELOPMENT	add item 4 PROJ-4	2	Min 2
IN PROGRESS	add item 3 PROJ-3	1	
REVIEW	add item 1 PROJ-1	1	Max 2
DONE	add item 2 PROJ-2	0	

Filter options at the top include: Only My Issues and Recently Updated.

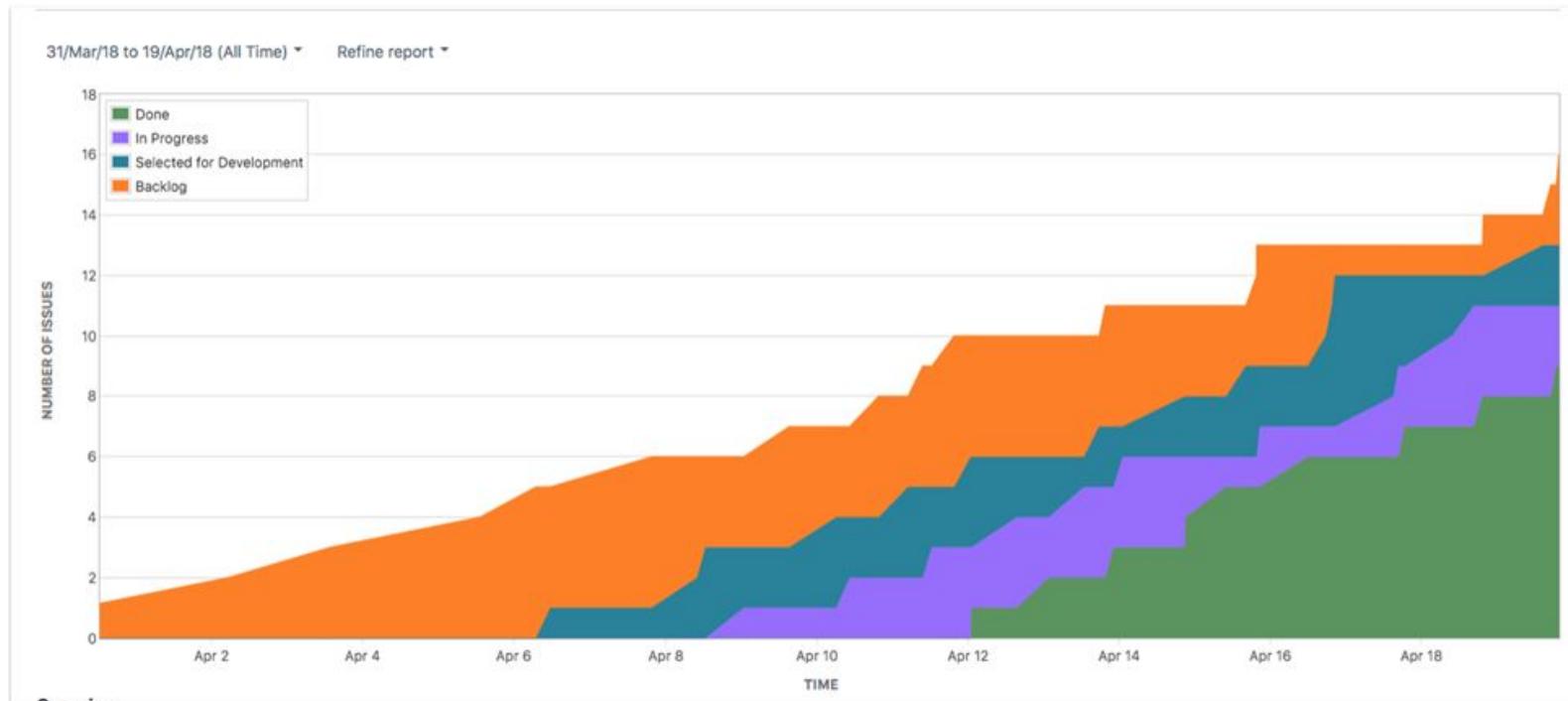
Information on the right side of the board states: "We're only showing recently modified issues." and "Looking for an older issue?"

What agile reports?

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



Cumulative flow diagram



Topics

Kanban Method Overview
Improving flow
Separate Backlog



Separate Backlog

Projects / projectA / PROJ board

Kanban board

Only My Issues Recently Updated

BACKLOG 1

- add item 6
PROJ-6
- add item 3
PROJ-3
- add item 5
PROJ-5

SELECTED FOR DEVELOPMENT 2

- add item 3
PROJ-3
- add item 5
PROJ-5

IN PROGRESS 1

- add item 4
PROJ-4

DONE 2

- add item 1
PROJ-1
- add item 2
PROJ-2

We're only showing recently modified issues.

Looking for an older issue?

can move these issues off of the board

Why a Separate Backlog?

- 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

The screenshot shows a Kanban board interface with four columns: BACKLOG, SELECTED FOR DEVELOPMENT, IN PROGRESS, and DONE. Each column contains several items with small icons next to them. A green callout box points to the first item in the BACKLOG column, which is labeled 'add item 6'. The callout box contains the text: 'can move these issues off of the board'.

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

We're only showing recently modified issues.

Looking for an older issue?

Managing the separate backlog

The screenshot shows a Jira project interface for 'projectA'. On the left, a sidebar menu lists project navigation options: PROJ board (selected), Backlog (highlighted with a red arrow), Kanban board, Reports, Releases, Issues and filters, Pages, Components, Add item, and Project settings.

The main area displays the 'Backlog' for the 'PROJ board'. At the top, there are search, user profile, and filter buttons ('Only My Issues', 'Recently Updated'). A dropdown menu for 'Selected for Development' shows two items: 'add item 3' (PROJ-3) and 'add item 5' (PROJ-5). Below this, another section titled 'Backlog' shows one issue: 'add item 6' (PROJ-6). A '+ Create issue' button is also present.

Takeaways

- 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 the team
- In Jira, the backlog can be separated from the board, simplifying the board and allowing separate backlog work

Lab 5 - Kanban Projects

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

Scrum Projects



What will you learn?

- Identify scrum artifacts
- Create and execute sprints
- View scrum reports

Topics

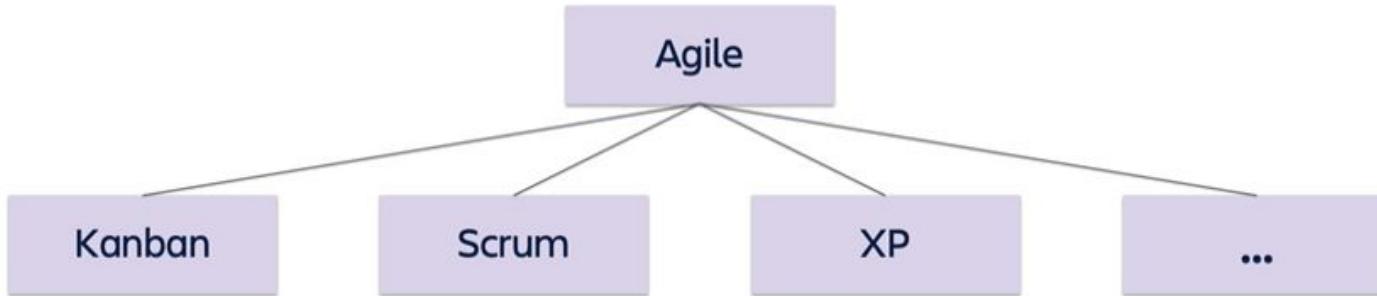
Scrum Overview

Creating and executing sprints
Scrum reports

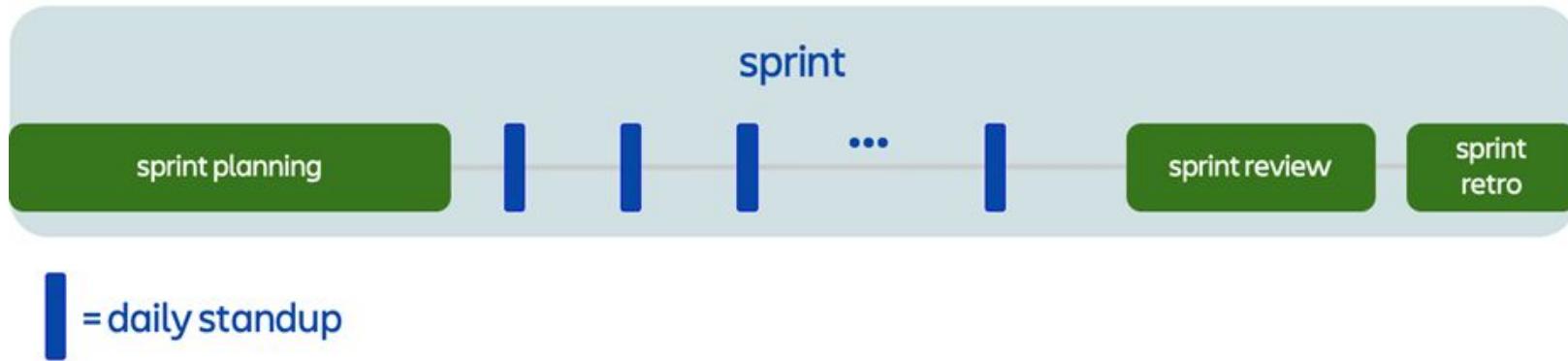


What is Scrum?

- "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



Scrum Events



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

Scrum Overview

Creating and executing sprints

Scrum reports



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

Backlog 3 issues

Create sprint ...

	add item 1	PRJ-1 ↑ -
	add item 2	PRJ-2 ↑ -
	add item 3	PRJ-3 ↑ -

+ Create issue

VERSIONS
EPICS

Creating a sprint with jira

Projects / projectB / PRJ board

Backlog

Share ...

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

- [] add item 1 PRJ-1 ↑ -
- [] add item 2 PRJ-2 ↑ -
- [] add item 3 PRJ-3 ↑ -

+ Create issue

The screenshot shows the Jira Backlog board for the 'PRJ board' under the 'projectB' project. At the top, there are filters for 'Only My Issues' and 'Recently Updated'. Below the header, a section for 'PRJ Sprint 1' is shown with a count of '0 issues'. A 'Start sprint' button is available, along with links for 'Linked pages' and more options. The board is organized into columns labeled 'VERSIONS' and 'EPICS'. A central area is titled 'Plan your sprint' with instructions to agree on work and drag issues to the sprint. Below this, a 'Create issue' button is present. The backlog section contains three items: 'add item 1', 'add item 2', and 'add item 3', each associated with a PRJ-1, PRJ-2, or PRJ-3 epic respectively. A 'Create sprint' button is located at the bottom right of the sprint section.

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

VERSIONS < PRJ Sprint 1 2 issues Start sprint Linked pages 0 ...

EPICs

<input type="checkbox"/> add item 1	PRJ-1 ↑ 1
<input type="checkbox"/> add item 2	PRJ-2 ↑ 2

+ Create issue

2 issues Estimate 3

Backlog 1 issue Create sprint ...

<input type="checkbox"/> add item 3	PRJ-3 ↑ 3
-------------------------------------	-----------

+ Create issue

sprint backlog

Estimation – story points

- 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 ...

PRJ Sprint 1 2 issues

Start sprint

Linked pages 0 ...

PRJ-1 1

add item 1

PRJ-2 2

add item 2

+ Create issue

2 issues Estimate 3

Backlog 1 issue

Create sprint ...

PRJ-3 3

+ Create issue

Description

Add a description...

Story Points

1

The screenshot shows a Jira backlog board for 'projectB'. At the top, there's a header with 'Projects / projectB / PRJ board' and a 'Backlog' section. Below the header, there's a search bar, a user icon, and filter options 'Only My Issues' and 'Recently Updated'. A 'Start sprint' button is visible above the backlog lists. The first list is titled 'PRJ Sprint 1' and contains two items: 'add item 1' (estimated at 1 story point) and 'add item 2' (estimated at 2 story points). Below this is a 'Backlog' section with one item: 'add item 3' (estimated at 3 story points). On the right side of the board, there's a sidebar with a 'Story Points' section containing a large number '1'.

Sprint details

Start sprint

2 issues will be included in this sprint.

Sprint name: *
PRJ Sprint 1

Duration: *
2 weeks

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

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

Sprint goal:

Sprint Goal

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

Start sprint

2 issues will be included in this sprint.

Sprint name: * PRJ Sprint 1

Duration: * 2 weeks

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

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

Sprint goal:

Create the first product increment.



Sprint board

Only contains issues from the sprint backlog

The screenshot shows a Jira project interface for 'projectB' under 'PRJ board'. The title of the board is 'PRJ Sprint 1' with a subtitle 'Create the first product increment.' A progress bar indicates '9 days remaining' until the sprint ends. The board has three columns: 'TO DO', 'IN PROGRESS', and 'DONE'. In the 'TO DO' column, there are two items: 'add item 1' and 'add item 2'. Both items have a green checkmark icon, a small orange upward arrow icon, and a grey circular badge with the number '1' or '2' respectively. To the right of each item is the identifier 'PRJ-1' and 'PRJ-2' respectively. The sidebar on the left lists various project management options: PRJ board (selected), Backlog, Active sprints (selected), Reports, Releases, Issues and filters, Pages, Components, Add item, and Project settings.

Topics

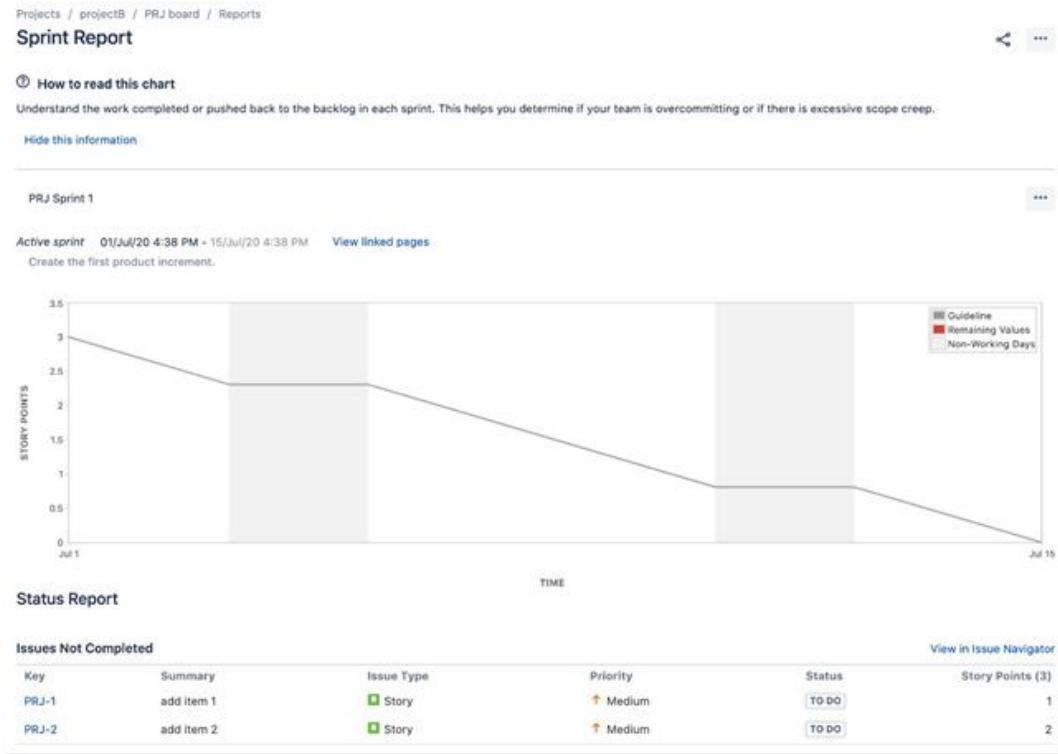
Scrum Overview
Creating and executing sprints
Scrum reports



Scrum reports- burndown chart



Scrum reports - sprint report



Velocity

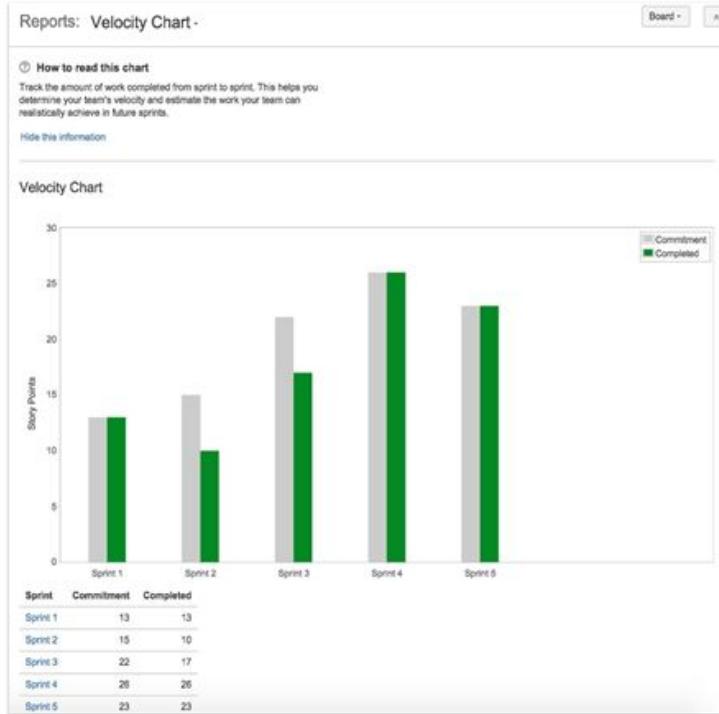
Represents the rate at which the team accomplishes work

- 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 chart

Shows the estimated and actual velocity of the team over time



Takeaways

- Scrum is an agile framework
- Jira induces scrum artifacts such as product backlogs, sprint backlogs and sprint boards
- Common sprint reports include burndown charts and velocity charts

Lab 6 - Scrum Projects

- 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 will you learn?

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

Topics

Searching Overview

Quick Search

Basic Search



Viewing project's progress



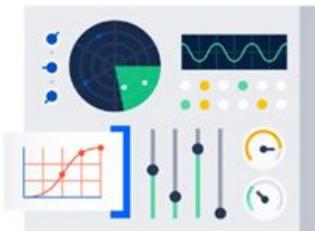
Boards



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 link to "Advanced search for issues". A section titled "RECENTLY VIEWED ISSUES" lists three items: "PRJ-1 add item 1", "PRJ-3 add item 3", and "PRJ-2 add item 2", each with a small green square icon. Below this is a section titled "BOARDS, PROJECTS AND FILTERS" which lists four items: "PROJ board", "PRJ board", "projectA (PROJ)", and "projectB (PRJ)", each with a small colored square icon. At the bottom of the sidebar is a navigation bar with links: "Go to all: Issues", "Boards", "Projects", "Filters", and "People".

Searching

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

All issues Save as

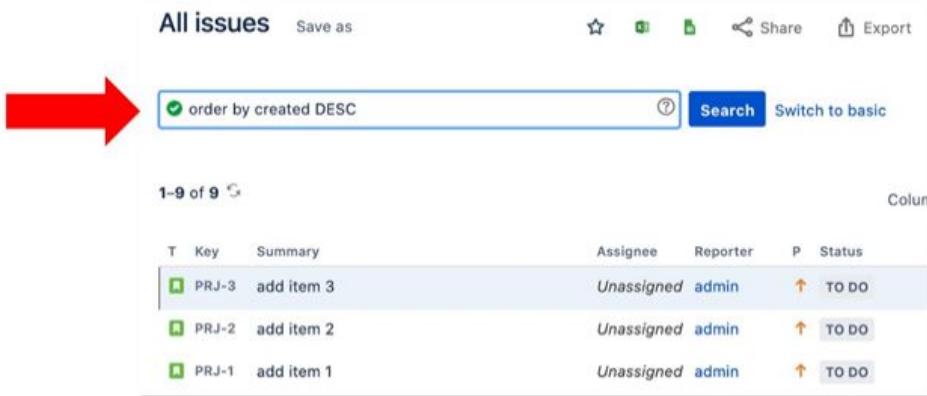
 Project: All ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ + More Contains text **Search** Switch to JQL

1-9 of 9 

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 square, a green checkmark, a share icon, and an export icon. Below these are buttons for "Save as", "Share", and "Export". A red arrow points to a dropdown menu labeled "order by created DESC". To the right of the dropdown are a help icon and a "Search" button. Next to the "Search" button is a link "Switch to basic". The search results table has columns for T, Key, Summary, Assignee, Reporter, P, and Status. It displays three items: PRJ-3 (add item 3), PRJ-2 (add item 2), and PRJ-1 (add item 1). All items are assigned to "Unassigned" and reported by "admin", and are in the "TO DO" status.

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

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

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

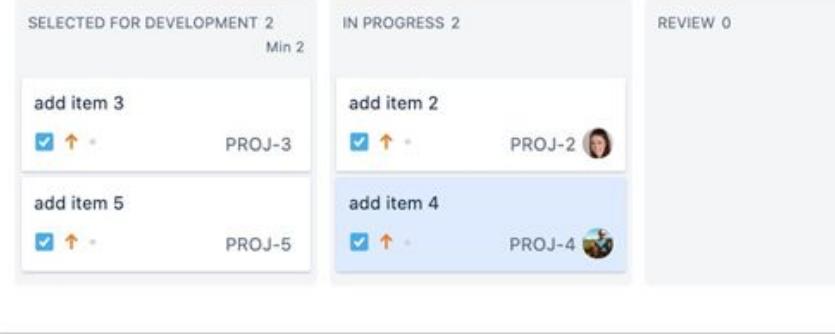
The screenshot shows a Kanban board interface with the following structure:

- Selected for Development:** 2 items (Min 2)
 - add item 3 (PROJ-3)
 - add item 5 (PROJ-5)
- In Progress:** 2 items (Min 2)
 - add item 2 (PROJ-2)
 - add item 4 (PROJ-4)
- Review:** 0 items

The top navigation bar includes a search icon, a user profile (A), and filter options: "Only My Issues" and "Recently Updated". A red box highlights this top navigation area.

Why searching is important?

Adapt your Jira experience to your team's processes

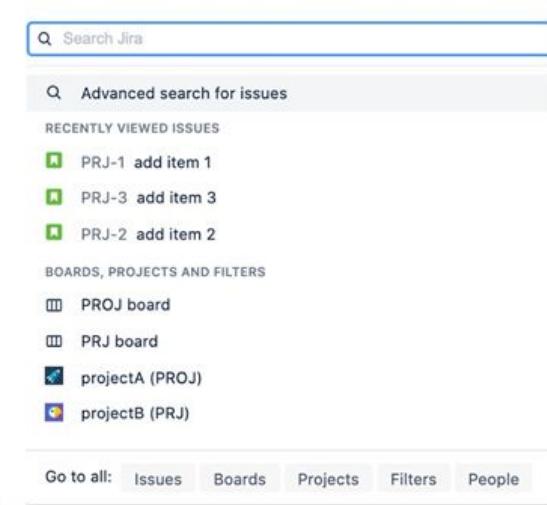


Topics

Searching Overview
Quick Search
Basic Search

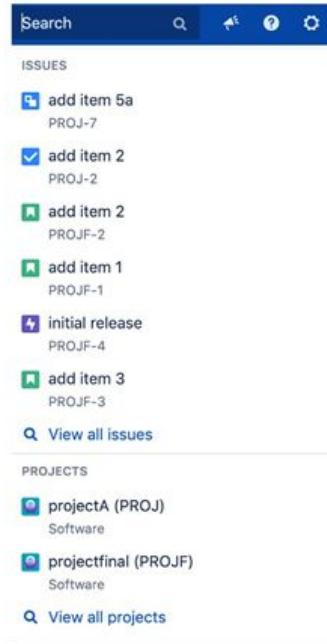


Quick Search



The screenshot shows the Jira Cloud search interface. At the top is a search bar with the placeholder "Search Jira". Below it is a "Advanced search for issues" button. The main area displays "RECENTLY VIEWED ISSUES" with three items: "PRJ-1 add item 1", "PRJ-3 add item 3", and "PRJ-2 add item 2". Under "BOARDS, PROJECTS AND FILTERS", there are two "PROJ board" entries and two project entries: "projectA (PROJ)" and "projectB (PRJ)". At the bottom is a navigation bar with links: "Go to all: Issues", "Boards", "Projects", "Filters", and "People".

Cloud



The screenshot shows the Jira Server search interface. It has a header with "Search" and a magnifying glass icon. Below the header are sections for "ISSUES" and "PROJECTS". The "ISSUES" section lists several items: "add item 5a PROJ-7" (checkbox checked), "add item 2 PROJ-2" (checkbox checked), "add item 2 PROJF-2" (checkbox checked), "add item 1 PROJF-1" (checkbox checked), "initial release PROJF-4" (checkbox checked), and "add item 3 PROJF-3" (checkbox checked). There is also a link to "View all issues". The "PROJECTS" section lists two projects: "projectA (PROJ) Software" and "projectfinal (PROJF) Software". There is also a link to "View all projects".

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

Find Projects... 

All 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	

Columns ▾

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	Resolution
<input checked="" type="checkbox"/>	PROJ-4	add item 4	Ryan Lee	admin	↑	IN PROGRESS	Unresolved
<input checked="" type="checkbox"/>	PROJ-2	add item 2	Alana Grant	admin	↑	IN PROGRESS	Unresolved
<input checked="" type="checkbox"/>	PROJ-3	add item 3	Unassigned	admin	↑	SELECTED FOR DEV...	Unresolved
<input checked="" type="checkbox"/>	PROJ-6	add item 6	Unassigned	admin	↑	BACKLOG	Unresolved
<input checked="" type="checkbox"/>	PROJ-5	add item 5	Unassigned	admin	↑	SELECTED FOR DEV...	Unresolved

1-5 of 5



Searching more fields

Search Save as

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

1-5 of 5

T	Key	Summary	Responsible	Last Activity
<input checked="" type="checkbox"/>	PROJ-4	add item 4	GRESS	Unresolved
<input checked="" type="checkbox"/>	PROJ-2	add item 2	GRESS	Unresolved
<input checked="" type="checkbox"/>	PROJ-3	add item 3	ED FOR DEV...	Unresolved
<input checked="" type="checkbox"/>	PROJ-6	add item 6	DG	Unresolved
<input checked="" type="checkbox"/>	PROJ-5	add item 5	ED FOR DEV...	Unresolved

All Criteria

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

...excluding 9 hidden

1-5 of 5

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
Ryan Lee	admin	↑	IN PROGRESS	Unresolved
Alana Grant	admin	↑	IN PROGRESS	Unresolved
SELECTED FOR DEVELOPMENT			SELECTED FOR DEV...	Unresolved
Unassigned	admin	↑	BACKLOG	Unresolved
Unassigned	admin	↑	SELECTED FOR DEV...	Unresolved

1-5 of 5

The screenshot shows a search interface for 'projectA'. The search bar includes filters for Type, Status, Assignee, and Priority, with 'Priority: All' currently selected. A dropdown menu lists priority levels: Highest (red up arrow), High (red up arrow), Medium (orange up arrow), Low (green down arrow), and Lowest (green down arrow). Below the dropdown is a table listing five items. Each item has columns for Assignee, Reporter, Priority (indicated by an orange up arrow), Status, and Resolution. The first two items are in 'IN PROGRESS' status, while the others are 'BACKLOG'. The third item is highlighted with a dark gray background and labeled 'SELECTED FOR DEVELOPMENT'. The fifth item is labeled 'SELECTED FOR DEV...'. At the bottom, a page number '1-190' is visible.

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

Within the last 3 days

More than minutes ago

Between 11-Jan-2011 and 30-Jan-2011

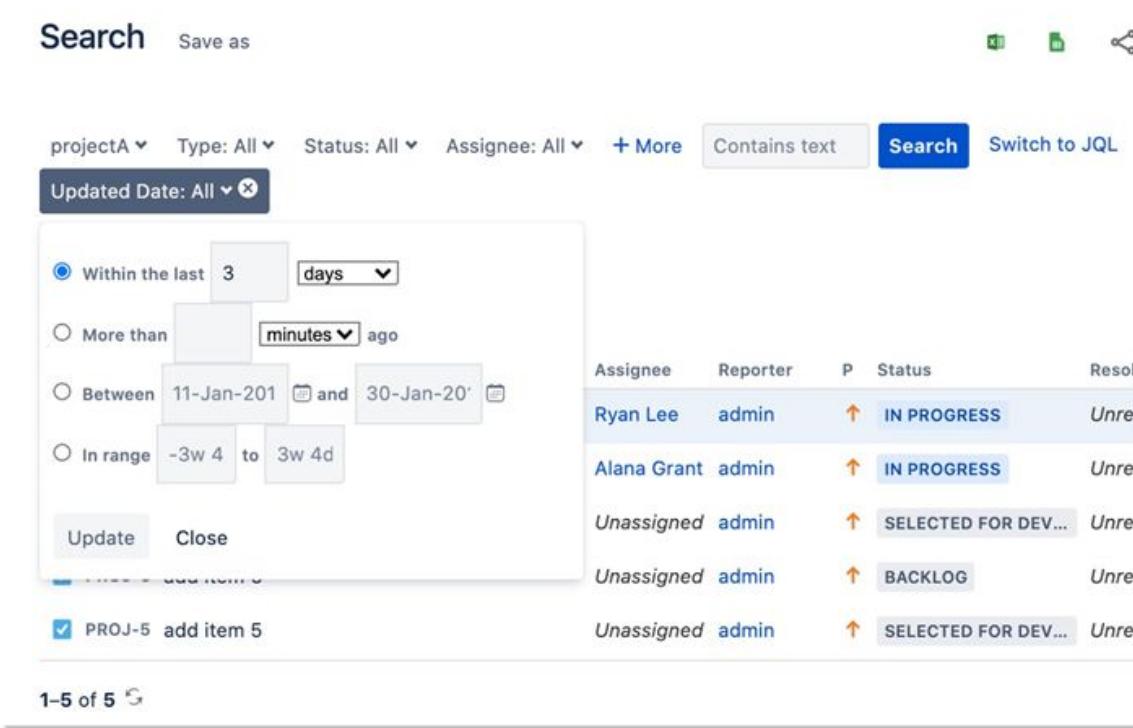
In range -3w 4d to 3w 4d

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

PROJ-5 add item 5

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

1-5 of 5 [G](#)



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 7 - 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

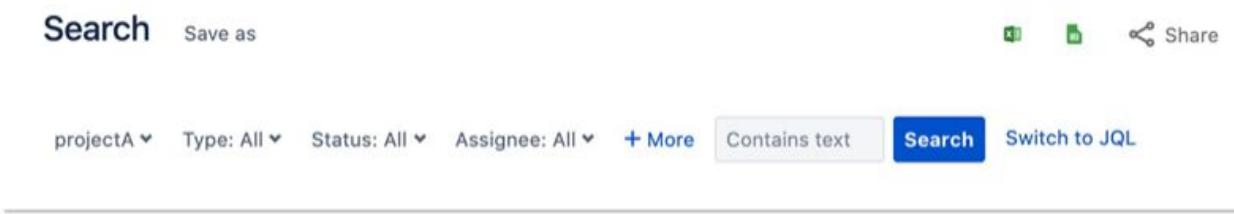


Basic and advanced/JQL search

basic

Search Save as

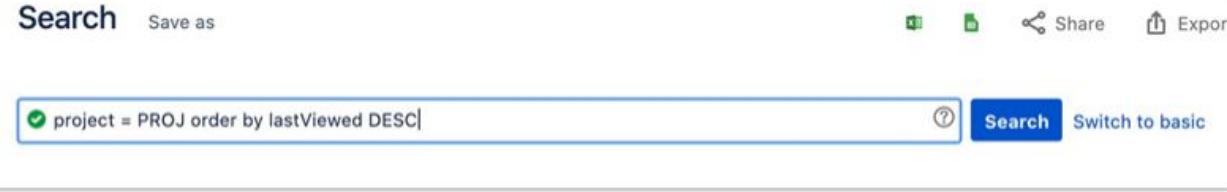
projectA ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ + More Contains text **Search** Switch to JQL



advanced/JQL

Search Save as

project = PROJ order by lastViewed DESC [?](#) **Search** Switch to basic



JQL

All issues

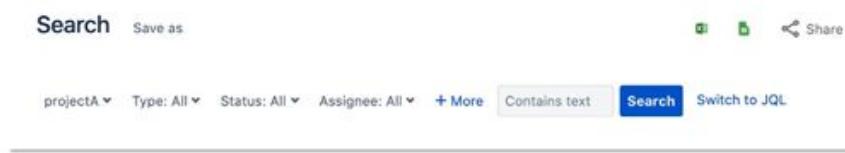
Save as



order by created DESC

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

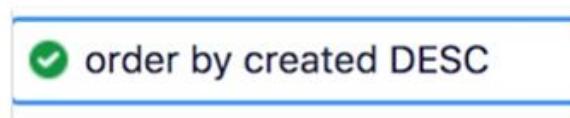
Basic vs Advanced/JQL search



A screenshot of a search interface. At the top, there's a "Search" button and a "Save as" link. On the right, there are icons for copy, paste, and share, followed by a "Share" button. Below the top bar, there are dropdown menus for "projectA", "Type: All", "Status: All", "Assignee: All", and a "+ More" button. There are also buttons for "Contains text" and "Search". To the right of the search buttons is a "Switch to JQL" link.

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

The screenshot shows a Jira search interface. At the top, there are dropdowns for 'projectA', 'Type: All', 'Status: All', 'Assignee: All', and a 'More' button. To the right of these are input fields for 'Contains text' and a 'Search' button. Next to the search buttons is a 'Switch to JQL' link. A large red arrow points from the 'Switch to JQL' link to a tooltip box. The tooltip box has a blue header bar with a green checkmark icon and the text 'project = PROJ ORDER BY created DESC'. Below the header, it says '1-6 of 6'.

The two main parts of a JQL query



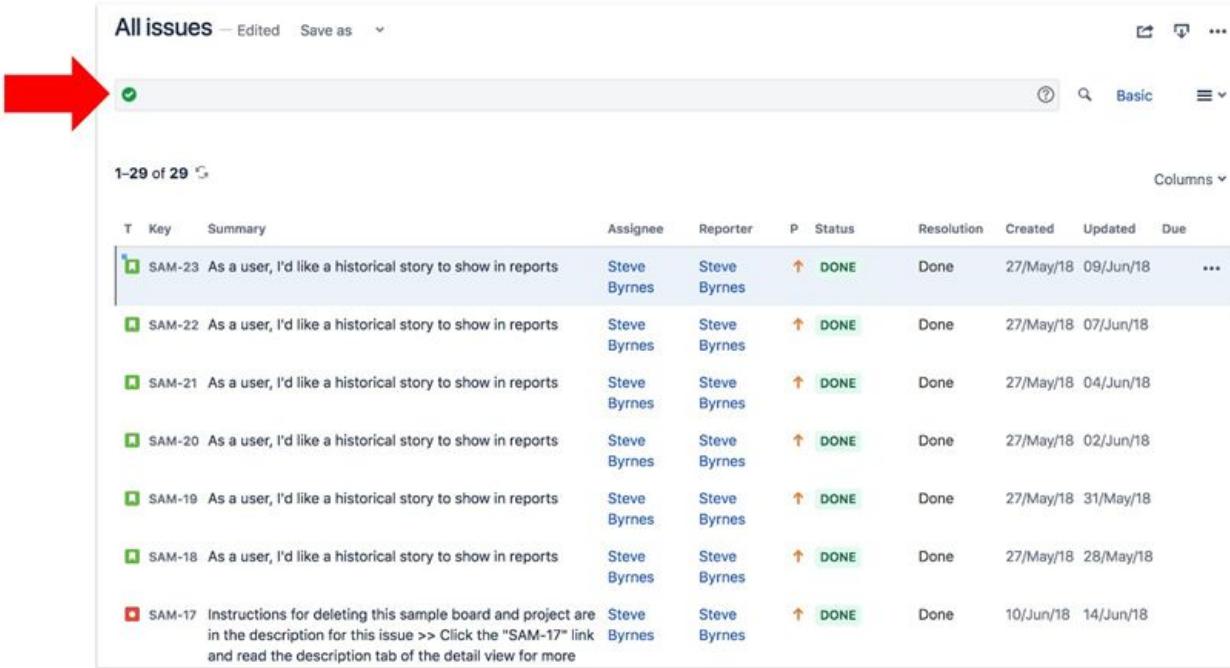
project = PROJ order by created DESC



search clause - selects a
subset of issues

order by clause - orders the
results

The simplest JQL query



A screenshot of a Jira software interface showing a list of issues titled "All issues". A red arrow points to the search bar at the top left. The search bar contains a green checkmark icon. The interface includes standard Jira navigation elements like "Edited" and "Save as", and a toolbar with "Basic" and "More" buttons.

1-29 of 29

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
1	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	...
2	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	
3	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	
4	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	
5	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	
6	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	
7	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

All issues Discard changes

The screenshot shows a search interface for JQL queries. At the top, it says "All issues" and "Discard changes". Below this is a search bar containing the prefix "pr|". An autocomplete dropdown menu is open, listing suggestions: "priority", "progress", "project", "project.property", and "Syntax Help". The "priority" suggestion is the first item in the list.

- priority
- progress
- project
- project.property
- Syntax Help

Advanced searching field 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', and a search icon. Below the header, a breadcrumb trail shows 'Atlassian Support / Jira Software / Documentation / ... / ... / Advanced searching'. A dropdown menu indicates 'Cloud' is selected. The main content title is 'Advanced searching - fields reference'. A paragraph explains what JQL fields are and how clauses work. Below this is a section titled 'Affected version' with a note about searching by version name or ID. A table provides details for the 'affectedVersion' field. To the right is a sidebar titled 'On this page' listing various JQL fields.

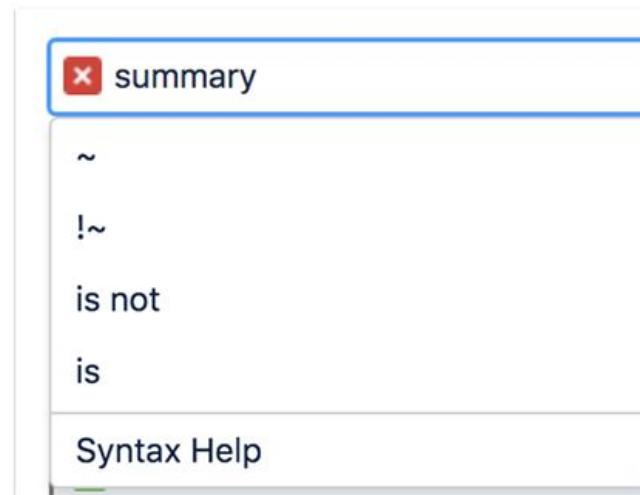
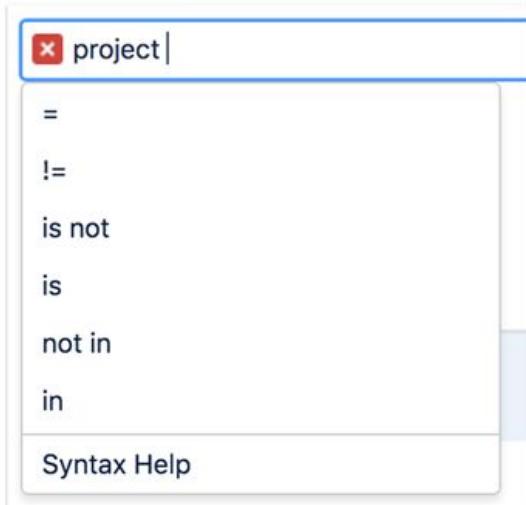
Syntax	affectedVersion
Field Type	VERSION
Auto-complete	Yes
Supported operators	=, !=, >, >=, <, <= IS, IS NOT, IN, NOT IN Note that the comparison operators (e.g. ">") use the version order that has been set up by your project

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 displaying the Jira Core Support documentation. The page title is "Advanced searching - operators reference". The left sidebar contains navigation links for Atlassian Support, Jira Core, Documentation, and a search bar. The main content area includes a brief introduction to JQL operators, a section on the EQUALS operator with examples, and a "On this page" sidebar listing various operators.

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

project = projectA

1-6 of 6

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/20
<input checked="" type="checkbox"/>	PROJ-5	add item 5	Unassigned	admin		SELECTED FOR DEV...	Unresolved	01/Jul/20	01/Jul/20
<input checked="" type="checkbox"/>	PROJ-4	add item 4	Ryan Lee	admin		IN PROGRESS	Unresolved	01/Jul/20	02/Jul/20

Share Export

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> <b>field value</b>  
project = projectA
```

```
<field name> <operator> <b>function</b>  
assignee = currentUser()
```

Advanced Searching-functions Reference

The screenshot shows a web browser displaying the Jira Core Support documentation. The URL is https://confluence.atlassian.com/jiracorecloud/advanced-searching-functions-reference-765593719.html. The page title is "Advanced searching - functions reference". The left sidebar includes links for Atlassian Support, Jira Core, Documentation, Cloud, and Server. The main content area starts with a brief introduction about JQL functions and a note about empty fields. Below this is a section for the "approved()" function, which is described as being applicable if Jira Service Desk is installed and licensed, and is used for searching requests requiring approval. A sidebar on the right lists other available functions: approved(), approver(), breached(), cascadeOption(), closedSprints(), completed(), componentsLeadByUser(), currentLogin(), currentUser(), earliestUnreleasedVersion(), elapsed(), endOfDay(), and endOfMonth().

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.

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 function

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

Issues created since the start of today

created > startOfDay()

Time 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

- A JQL query is behind all basic and advanced searches
- Leverage basic queries and autocomplete 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

Lab 8 - JQL

- Creates 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

Filter
Board filters
Quick Filters



Filter

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

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

Save a search to create filter

The screenshot shows the Jira interface with a sidebar on the left containing various filters and a main search results page on the right.

Left Sidebar (Filters):

- Filters
- Search issues
- OTHER
- My open issues (highlighted)
- Reported by me
- All issues
- Open issues
- Done issues
- Viewed recently
- Created recently
- Resolved recently
- Updated recently
- [View all filters](#)

Main Search Results:

Title: My open issues

Toolbar: Edited, Save as (highlighted with a red box), Share, Export

Search Bar: assignee = currentUser() AND statusCategory = "In Progress" order by updated DESC

Buttons: Search, Switch to basic

Results: 1-1 of 1

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 filter

The screenshot shows the Jira interface for managing filters. On the left, a sidebar lists various filter categories: Filters, Search issues, OTHER, My open issues (which is selected and highlighted in grey), 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" and shows a single result: "PROJ-2 a". A modal window titled "Save Filter" is open in the center, prompting the user to enter a "Filter Name". The input field contains "My in progress". Below the input field is a placeholder text: "Enter a name for this Filter". At the bottom of the modal are two buttons: "Submit" and "Cancel".

The newly created filter

The screenshot shows a Jira search interface. On the left, a sidebar lists various filters, with 'My in progress' highlighted by a red arrow. The main area displays a search result titled 'My in progress' with the following details:

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

Results: 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	***

Filter Details:

- Key: PROJ-2
- Summary: add item 2
- Assignee: admin
- Reporter: admin
- Status: IN PROGRESS
- Resolution: Unresolved
- Created: 01/Jul/20
- Updated: 02/Jul/20

Edit Filter Details

The screenshot shows the Jira interface for managing filter details and subscriptions.

Left Panel (Filters):

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

Middle Panel (My in progress Filter Details):

Header: My in progress [Save as](#) [Details](#)

Owner: A Owned by admin

Permissions: This filter is only visible to you. [Edit permissions](#)

Subscriptions: This filter has no subscriptions. [New subscription](#)

Right Panel (Subscription Management):

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

The screenshot shows the Jira Filters interface. On the left, a sidebar lists various filter categories: Filters, Search issues, STARRED (with 'My in progress' selected), OTHER (with 'My open issues', 'Reported by me', 'All issues', 'Open issues', 'Done issues', 'Viewed recently', 'Created recently', 'Resolved recently', and 'Updated recently'), and a bottom link 'View all filters'. A large red arrow points from the bottom of the sidebar towards this link. The main area displays a table of filters with columns for Name, Owner, Project, Group, Access, and Starred by. Three filters are listed: 'Filter for PRJ board' (Owner: admin, Access: projectB, All roles, Starred by: 0 people), 'Filter for PROJ board' (Owner: admin, Access: projectA, All roles, Starred by: 0 people), and 'My in progress' (Owner: admin, Access: Private, Starred by: 1 person). A context menu is open over the third filter, showing options: 'Manage subscriptions', 'Copy filter', 'Edit', and 'Delete'.

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

Edit filter 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

Columns ▾

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

Filter
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 filters

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

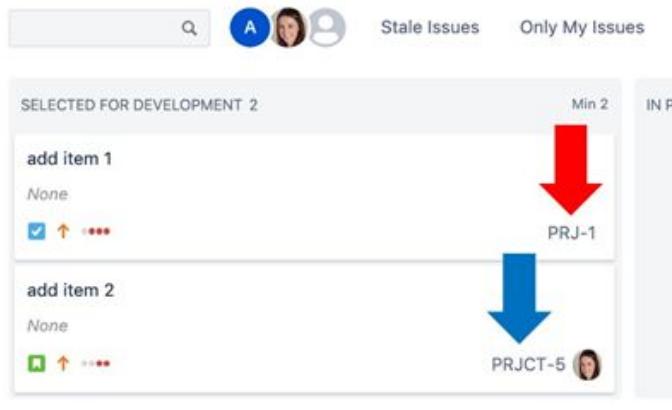
No shares

Edit Filter Shares

Filter Query

project in (PRJCT, PRJ) ORDER BY Rank ASC

Kanban board



Topics

Filter
Board filters
Quick Filters



Filtering a board: quick filters

Further filters issues displayed on a board

Projects / projectA / PROJ board

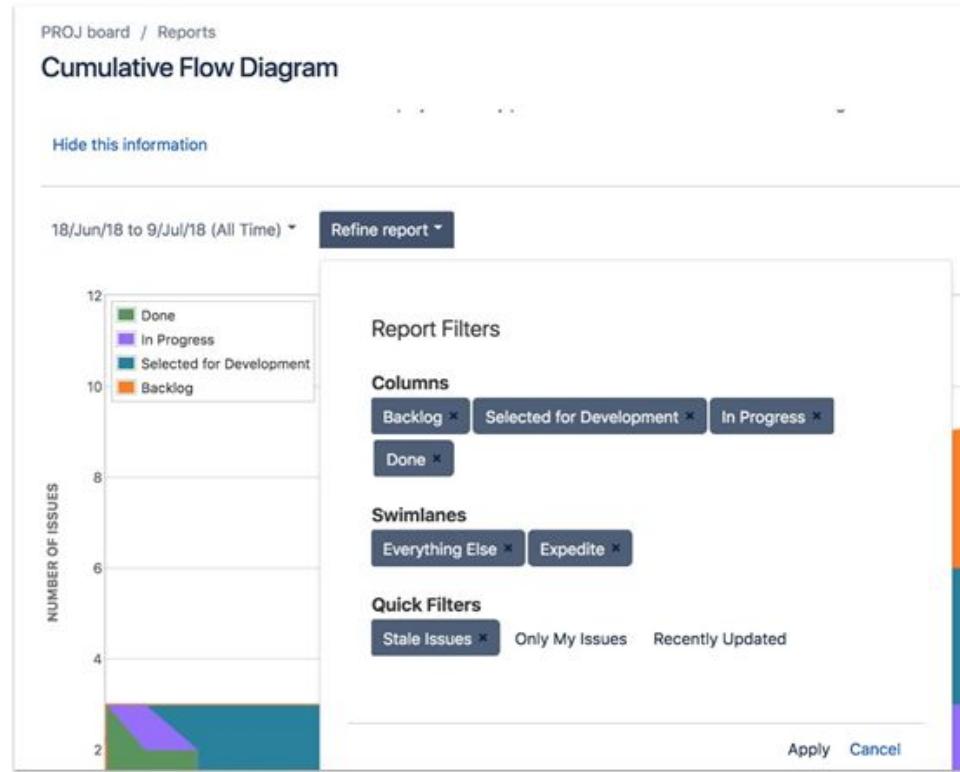
Kanban board

The screenshot shows a Kanban board interface with three columns: "SELECTED FOR DEVELOPMENT", "IN PROGRESS", and "REVIEW". Each column contains two items, each with a title, a checkmark icon, and a small profile picture. The "IN PROGRESS" column has one item with a checkmark and another with a blue circle icon. The "REVIEW" column is currently empty.

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

Only My Issues Recently Updated

Quick filters and reports



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 9 - Filters

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

Epics



What will you Learn?

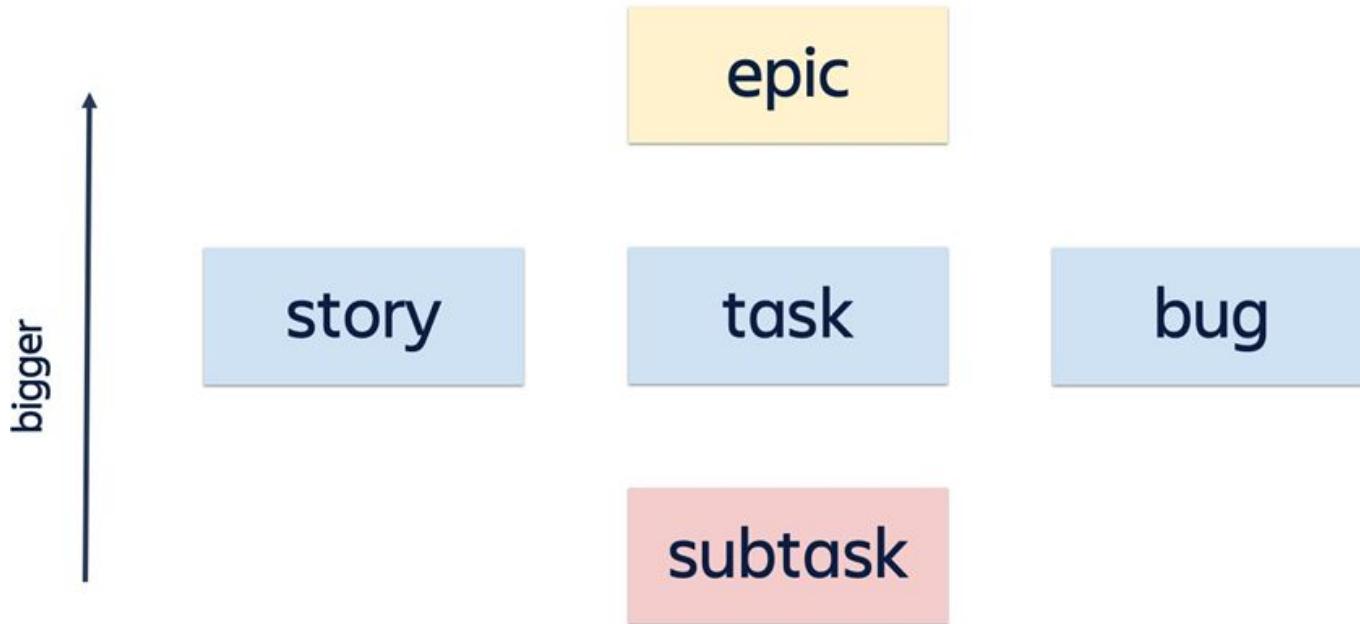
- Describe epics
- Work with epics
- Manage epics in the backlog

Topics

Epic Overview
Working with Epics
Epics in backlog



Jira's issue type hierarchy



Epic

- 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.

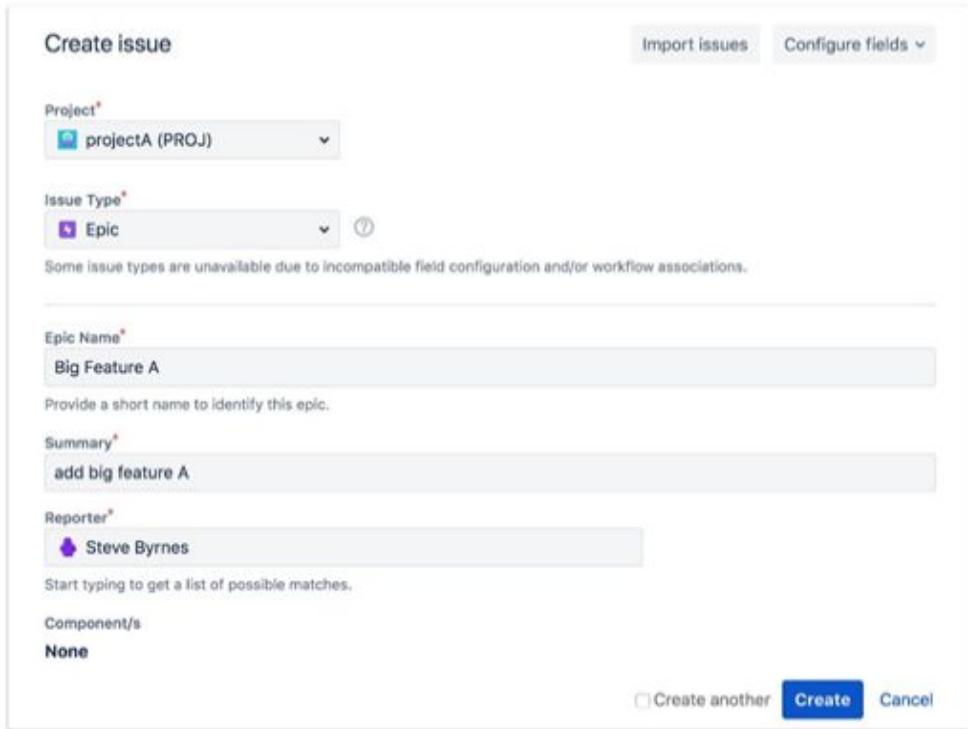
Summary* add big feature A

Reporter* Steve Byrnes

Start typing to get a list of possible matches.

Component/s None

Create another Cancel



Why Epic?

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

Topics

Epic Overview Working with Epics Epics in backlog



Creating an epics

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

Components
None

Attachment
Drop files to attach, or browse.

Description

Style B I U A ^A _A ^{2A} _{2A} ^{3A} _{3A} ^{4A} _{4A} ^{5A} _{5A} ^{6A} _{6A} ^{7A} _{7A} ^{8A} _{8A} ^{9A} _{9A} ^{10A} _{10A} ^{11A} _{11A} ^{12A} _{12A} ^{13A} _{13A} ^{14A} _{14A} ^{15A} _{15A} ^{16A} _{16A} ^{17A} _{17A} ^{18A} _{18A} ^{19A} _{19A} ^{20A} _{20A} ^{21A} _{21A} ^{22A} _{22A} ^{23A} _{23A} ^{24A} _{24A} ^{25A} _{25A} ^{26A} _{26A} ^{27A} _{27A} ^{28A} _{28A} ^{29A} _{29A} ^{30A} _{30A} ^{31A} _{31A} ^{32A} _{32A} ^{33A} _{33A} ^{34A} _{34A} ^{35A} _{35A} ^{36A} _{36A} ^{37A} _{37A} ^{38A} _{38A} ^{39A} _{39A} ^{40A} _{40A} ^{41A} _{41A} ^{42A} _{42A} ^{43A} _{43A} ^{44A} _{44A} ^{45A} _{45A} ^{46A} _{46A} ^{47A} _{47A} ^{48A} _{48A} ^{49A} _{49A} ^{50A} _{50A} ^{51A} _{51A} ^{52A} _{52A} ^{53A} _{53A} ^{54A} _{54A} ^{55A} _{55A} ^{56A} _{56A} ^{57A} _{57A} ^{58A} _{58A} ^{59A} _{59A} ^{60A} _{60A} ^{61A} _{61A} ^{62A} _{62A} ^{63A} _{63A} ^{64A} _{64A} ^{65A} _{65A} ^{66A} _{66A} ^{67A} _{67A} ^{68A} _{68A} ^{69A} _{69A} ^{70A} _{70A} ^{71A} _{71A} ^{72A} _{72A} ^{73A} _{73A} ^{74A} _{74A} ^{75A} _{75A} ^{76A} _{76A} ^{77A} _{77A} ^{78A} _{78A} ^{79A} _{79A} ^{80A} _{80A} ^{81A} _{81A} ^{82A} _{82A} ^{83A} _{83A} ^{84A} _{84A} ^{85A} _{85A} ^{86A} _{86A} ^{87A} _{87A} ^{88A} _{88A} ^{89A} _{89A} ^{90A} _{90A} ^{91A} _{91A} ^{92A} _{92A} ^{93A} _{93A} ^{94A} _{94A} ^{95A} _{95A} ^{96A} _{96A} ^{97A} _{97A} ^{98A} _{98A} ^{99A} _{99A} ^{100A} _{100A} ^{101A} _{101A} ^{102A} _{102A} ^{103A} _{103A} ^{104A} _{104A} ^{105A} _{105A} ^{106A} _{106A} ^{107A} _{107A} ^{108A} _{108A} ^{109A} _{109A} ^{110A} _{110A} ^{111A} _{111A} ^{112A} _{112A} ^{113A} _{113A} ^{114A} _{114A} ^{115A} _{115A} ^{116A} _{116A} ^{117A} _{117A} ^{118A} _{118A} ^{119A} _{119A} ^{120A} _{120A} ^{121A} _{121A} ^{122A} _{122A} ^{123A} _{123A} ^{124A} _{124A} ^{125A} _{125A} ^{126A} _{126A} ^{127A} _{127A} ^{128A} _{128A} ^{129A} _{129A} ^{130A} _{130A} ^{131A} _{131A} ^{132A} _{132A} ^{133A} _{133A} ^{134A} _{134A} ^{135A} _{135A} ^{136A} _{136A} ^{137A} _{137A} ^{138A} _{138A} ^{139A} _{139A} ^{140A} _{140A} ^{141A} _{141A} ^{142A} _{142A} ^{143A} _{143A} ^{144A} _{144A} ^{145A} _{145A} ^{146A} _{146A} ^{147A} _{147A} ^{148A} _{148A} ^{149A} _{149A} ^{150A} _{150A} ^{151A} _{151A} ^{152A} _{152A} ^{153A} _{153A} ^{154A} _{154A} ^{155A} _{155A} ^{156A} _{156A} ^{157A} _{157A} ^{158A} _{158A} ^{159A} _{159A} ^{160A} _{160A} ^{161A} _{161A} ^{162A} _{162A} ^{163A} _{163A} ^{164A} _{164A} ^{165A} _{165A} ^{166A} _{166A} ^{167A} _{167A} ^{168A} _{168A} ^{169A} _{169A} ^{170A} _{170A} ^{171A} _{171A} ^{172A} _{172A} ^{173A} _{173A} ^{174A} _{174A} ^{175A} _{175A} ^{176A} _{176A} ^{177A} _{177A} ^{178A} _{178A} ^{179A} _{179A} ^{180A} _{180A} ^{181A} _{181A} ^{182A} _{182A} ^{183A} _{183A} ^{184A} _{184A} ^{185A} _{185A} ^{186A} _{186A} ^{187A} _{187A} ^{188A} _{188A} ^{189A} _{189A} ^{190A} _{190A} ^{191A} _{191A} ^{192A} _{192A} ^{193A} _{193A} ^{194A} _{194A} ^{195A} _{195A} ^{196A} _{196A} ^{197A} _{197A} ^{198A} _{198A} ^{199A} _{199A} ^{200A} _{200A} ^{201A} _{201A} ^{202A} _{202A} ^{203A} _{203A} ^{204A} _{204A} ^{205A} _{205A} ^{206A} _{206A} ^{207A} _{207A} ^{208A} _{208A} ^{209A} _{209A} ^{210A} _{210A} ^{211A} _{211A} ^{212A} _{212A} ^{213A} _{213A} ^{214A} _{214A} ^{215A} _{215A} ^{216A} _{216A} ^{217A} _{217A} ^{218A} _{218A} ^{219A} _{219A} ^{220A} _{220A} ^{221A} _{221A} ^{222A} _{222A} ^{223A} _{223A} ^{224A} _{224A} ^{225A} _{225A} ^{226A} _{226A} ^{227A} _{227A} ^{228A} _{228A} ^{229A} _{229A} ^{230A} _{230A} ^{231A} _{231A} ^{232A} _{232A} ^{233A} _{233A} ^{234A} _{234A} ^{235A} _{235A} ^{236A} _{236A} ^{237A} _{237A} ^{238A} _{238A} ^{239A} _{239A} ^{240A} _{240A} ^{241A} _{241A} ^{242A} _{242A} ^{243A} _{243A} ^{244A} _{244A} ^{245A} _{245A} ^{246A} _{246A} ^{247A} _{247A} ^{248A} _{248A} ^{249A} _{249A} ^{250A} _{250A} ^{251A} _{251A} ^{252A} _{252A} ^{253A} _{253A} ^{254A} _{254A} ^{255A} _{255A} ^{256A} _{256A} ^{257A} _{257A} ^{258A} _{258A} ^{259A} _{259A} ^{260A} _{260A} ^{261A} _{261A} ^{262A} _{262A} ^{263A} _{263A} ^{264A} _{264A} ^{265A} _{265A} ^{266A} _{266A} ^{267A} _{267A} ^{268A} _{268A} ^{269A} _{269A} ^{270A} _{270A} ^{271A} _{271A} ^{272A} _{272A} ^{273A} _{273A} ^{274A} _{274A} ^{275A} _{275A} ^{276A} _{276A} ^{277A} _{277A} ^{278A} _{278A} ^{279A} _{279A} ^{280A} _{280A} ^{281A} _{281A} ^{282A} _{282A} ^{283A} _{283A} ^{284A} _{284A} ^{285A} _{285A} ^{286A} _{286A} ^{287A} _{287A} ^{288A} _{288A} ^{289A} _{289A} ^{290A} _{290A} ^{291A} _{291A} ^{292A} _{292A} ^{293A} _{293A} ^{294A} _{294A} ^{295A} _{295A} ^{296A} _{296A} ^{297A} _{297A} ^{298A} _{298A} ^{299A} _{299A} ^{300A} _{300A} ^{301A} _{301A} ^{302A} _{302A} ^{303A} _{303A} ^{304A} _{304A} ^{305A} _{305A} ^{306A} _{306A} ^{307A} _{307A} ^{308A} _{308A} ^{309A} _{309A} ^{310A} _{310A} ^{311A} _{311A} ^{312A} _{312A} ^{313A} _{313A} ^{314A} _{314A} ^{315A} _{315A} ^{316A} _{316A} ^{317A} _{317A} ^{318A} _{318A} ^{319A} _{319A} ^{320A} _{320A} ^{321A} _{321A} ^{322A} _{322A} ^{323A} _{323A} ^{324A} _{324A} ^{325A} _{325A} ^{326A} _{326A} ^{327A} _{327A} ^{328A} _{328A} ^{329A} _{329A} ^{330A} _{330A} ^{331A} _{331A} ^{332A} _{332A} ^{333A} _{333A} ^{334A} _{334A} ^{335A} _{335A} ^{336A} _{336A} ^{337A} _{337A} ^{338A} _{338A} ^{339A} _{339A} ^{340A} _{340A} ^{341A} _{341A} ^{342A} _{342A} ^{343A} _{343A} ^{344A} _{344A} ^{345A} _{345A} ^{346A} _{346A} ^{347A} _{347A} ^{348A} _{348A} ^{349A} _{349A} ^{350A} _{350A} ^{351A} _{351A} ^{352A} _{352A} ^{353A} _{353A} ^{354A} _{354A} ^{355A} _{355A} ^{356A} _{356A} ^{357A} _{357A} ^{358A} _{358A} ^{359A} _{359A} ^{360A} _{360A} ^{361A} _{361A} ^{362A} _{362A} ^{363A} _{363A} ^{364A} _{364A} ^{365A} _{365A} ^{366A} _{366A} ^{367A} _{367A} ^{368A} _{368A} ^{369A} _{369A} ^{370A} _{370A} ^{371A} _{371A} ^{372A} _{372A} ^{373A} _{373A} ^{374A} _{374A} ^{375A} _{375A} ^{376A} _{376A} ^{377A} _{377A} ^{378A} _{378A} ^{379A} _{379A} ^{380A} _{380A} ^{381A} _{381A} ^{382A} _{382A} ^{383A} _{383A} ^{384A} _{384A} ^{385A} _{385A} ^{386A} _{386A} ^{387A} _{387A} ^{388A} _{388A} ^{389A} _{389A} ^{390A} _{390A} ^{391A} _{391A} ^{392A} _{392A} ^{393A} _{393A} ^{394A} _{394A} ^{395A} _{395A} ^{396A} _{396A} ^{397A} _{397A} ^{398A} _{398A} ^{399A} _{399A} ^{400A} _{400A} ^{401A} _{401A} ^{402A} _{402A} ^{403A} _{403A} ^{404A} _{404A} ^{405A} _{405A} ^{406A} _{406A} ^{407A} _{407A} ^{408A} _{408A} ^{409A} _{409A} ^{410A} _{410A} ^{411A} _{411A} ^{412A} _{412A} ^{413A} _{413A} ^{414A} _{414A} ^{415A} _{415A} ^{416A} _{416A} ^{417A} _{417A} ^{418A} _{418A} ^{419A} _{419A} ^{420A} _{420A} ^{421A} _{421A} ^{422A} _{422A} ^{423A} _{423A} ^{424A} _{424A} ^{425A} _{425A} ^{426A} _{426A} ^{427A} _{427A} ^{428A} _{428A} ^{429A} _{429A} ^{430A} _{430A} ^{431A} _{431A} ^{432A} _{432A} ^{433A} _{433A} ^{434A} _{434A} ^{435A} _{435A} ^{436A} _{436A} ^{437A} _{437A} ^{438A} _{438A} ^{439A} _{439A} ^{440A} _{440A} ^{441A} _{441A} ^{442A} _{442A} ^{443A} _{443A} ^{444A} _{444A} ^{445A} _{445A} ^{446A} _{446A} ^{447A} _{447A} ^{448A} _{448A} ^{449A} _{449A} ^{450A} _{450A} ^{451A} _{451A} ^{452A} _{452A} ^{453A} _{453A} ^{454A} _{454A} ^{455A} _{455A} ^{456A} _{456A} ^{457A} _{457A} ^{458A} _{458A} ^{459A} _{459A} ^{460A} _{460A} ^{461A} _{461A} ^{462A} _{462A} ^{463A} _{463A} ^{464A} _{464A} ^{465A} _{465A} ^{466A} _{466A} ^{467A} _{467A} ^{468A} _{468A} ^{469A} _{469A} ^{470A} _{470A} ^{471A} _{471A} ^{472A} _{472A} ^{473A} _{473A} ^{474A} _{474A} ^{475A} _{475A} ^{476A} _{476A} ^{477A} _{477A} ^{478A} _{478A} ^{479A} _{479A} ^{480A} _{480A} ^{481A} _{481A} ^{482A} _{482A} ^{483A} _{483A} ^{484A} _{484A} ^{485A} _{485A} ^{486A} _{486A} ^{487A} _{487A} ^{488A} _{488A} ^{489A} _{489A} ^{490A} _{490A} ^{491A} _{491A} ^{492A} _{492A} ^{493A} _{493A} ^{494A} _{494A} ^{495A} _{495A} ^{496A} _{496A} ^{497A} _{497A} ^{498A} _{498A} ^{499A} _{499A} ^{500A} _{500A} ^{501A} _{501A} ^{502A} _{502A} ^{503A} _{503A} ^{504A} _{504A} ^{505A} _{505A} ^{506A} _{506A} ^{507A} _{507A} ^{508A} _{508A} ^{509A} _{509A} ^{510A} _{510A} ^{511A} _{511A} ^{512A} _{512A} ^{513A} _{513A} ^{514A} _{514A} ^{515A} _{515A} ^{516A} _{516A} ^{517A} _{517A} ^{518A} _{518A} ^{519A} _{519A} ^{520A} _{520A} ^{521A} _{521A} ^{522A} _{522A} ^{523A} _{523A} ^{524A} _{524A} ^{525A} _{525A} ^{526A} _{526A} ^{527A} _{527A} ^{528A} _{528A} ^{529A} _{529A} ^{530A} _{530A} ^{531A} _{531A} ^{532A} _{532A} ^{533A} _{533A} ^{534A} _{534A} ^{535A} _{535A} ^{536A} _{536A} ^{537A} _{537A} ^{538A} _{538A} ^{539A} _{539A} ^{540A} _{540A} ^{541A} _{541A} ^{542A} _{542A} ^{543A} _{543A} ^{544A} _{544A} ^{545A} _{545A} ^{546A} _{546A} ^{547A} _{547A} ^{548A} _{548A} ^{549A} _{549A} ^{550A} _{550A} ^{551A} _{551A} ^{552A} _{552A} ^{553A} _{553A} ^{554A} _{554A} ^{555A} _{555A} ^{556A} _{556A} ^{557A} _{557A} ^{558A} _{558A} ^{559A} _{559A} ^{560A} _{560A} ^{561A} _{561A} ^{562A} _{562A} ^{563A} _{563A} ^{564A} _{564A} ^{565A} _{565A} ^{566A} _{566A} ^{567A} _{567A} ^{568A} _{568A} ^{569A} _{569A} ^{570A} _{570A} ^{571A} _{571A} ^{572A} _{572A} ^{573A} _{573A} ^{574A} _{574A} ^{575A} _{575A} ^{576A} _{576A} ^{577A} _{577A} ^{578A} _{578A} ^{579A} _{579A} ^{580A} _{580A} ^{581A} _{581A} ^{582A} _{582A} ^{583A} _{583A} ^{584A} _{584A} ^{585A} _{585A} ^{586A} _{586A} ^{587A} _{587A} ^{588A} _{588A} ^{589A} _{589A} ^{590A} _{590A} ^{591A} _{591A} ^{592A} _{592A} ^{593A} _{593A} ^{594A} _{594A} ^{595A} _{595A} ^{596A} _{596A} ^{597A} _{597A} ^{598A} _{598A} ^{599A} _{599A} ^{600A} _{600A} ^{601A} _{601A} ^{602A} _{602A} ^{603A} _{603A} ^{604A} _{604A} ^{605A} _{605A} ^{606A} _{606A} ^{607A} _{607A} ^{608A} _{608A} ^{609A} _{609A} ^{610A} _{610A} ^{611A} _{611A} ^{612A} _{612A} ^{613A} _{613A} ^{614A} _{614A} ^{615A} _{615A} ^{616A} _{616A} ^{617A} _{617A} ^{618A} _{618A} ^{619A} _{619A} ^{620A} _{620A} <

Managing the issue of an epics

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

Header: PROJ-7, Give feedback, 1, ... X

Buttons: Attach, Create issue in epic, Link issue, ...

Assignee: Unassigned

Reporter: admin

Issues in this epic:

- PROJ-8 add feature A task 1 (checkbox checked, BACKLOG)

Comments: Add a comment... (with a tip: press M to comment)

Feature A:

- Story Points: None
- Original Estimate: 0m

Epic link

A field in the child issue pointing to the parent epic

The screenshot shows a Jira issue page for 'add feature A task 1' in 'PROJ-7 / PROJ-8'. The issue has a status of 'Backlog', is unassigned, and is reported by 'admin'. It is associated with the epic 'Feature A'. The epic card for 'Feature A' is shown on the right, with the title 'Epic Link' and the sub-task 'Feature A' listed.

PROJ-7 / PROJ-8

add feature A task 1

Backlog

Assignee: Unassigned

Reporter: admin

Labels: None

Epic Link

Feature A

Created 6 minutes ago | Updated 6 minutes ago

Adding an existing issue to an epics

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, there is a checkbox labeled 'PROJ-2'. On the right, there are standard Jira navigation icons: 'Give feedback', a comment count (1), a share icon, three dots, and a close button.

The main area contains fields for 'Description' (with placeholder 'Add a description...') and 'Activity' (with tabs for 'Comments' (selected), 'History', and 'Work log'). Below the activity section is a comment input field with a placeholder 'Add a comment...' and a note 'Pro tip: press M to comment'. A large red arrow points from the bottom left towards the 'Epic Link' dropdown menu.

The 'Epic Link' menu is open, showing a list of existing epics. The first item in the list is 'Select Epic', which is highlighted with a blue border. Below it is another entry: '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 

Epic labels on a board

Projects / projectA / PROJ board

Kanban board

The screenshot shows a Jira Kanban board with two columns: "BACKLOG 3" and "SELECTED FOR DEVELOPMENT... 2".

BACKLOG 3:

- Card: add feature A
Label: Feature A
- Card: add feature A task 1
Label: Feature A

SELECTED FOR DEVELOPMENT... 2:

- Card: add item 3
Label: PROJ-3
- Card: add item 5
Label: PROJ-5

Epic-based swimlanes

Projects / projectA / PROJ board

Kanban board

BACKLOG 3

SELECTED FOR DEVELOPM... 2
Min 2

IN PROGRESS 2

Feature A 1 issue

add feature A task 1

Feature A

PROJ-8

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

Epic Overview
Working with Epics
Epics in backlog



Creating an epics 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. On the left, there is a sidebar with 'VERSIONS' and tabs for 'EPICS' (which is selected), 'Create epic', and 'X'. A red arrow points upwards from the 'All issues' section towards the 'Create epic' tab. The main area shows a collapsed section for 'PRJ Sprint 1' which contains 2 issues, with a note to 'Create the first product increment.' and dates '01/Jul/20 4:38 PM • 15/Jul/20 4:38 PM'. Below this, there is a list of items under 'add item 1' and 'add item 2'. Further down, there is a section for 'Backlog' which contains 1 issue, with an option to 'add item 3' and a '+ Create issue' button.

EPICS Create epic X

All issues

VERSIONS

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

Only My Issues Recently Updated

EPICS Create epic X

All issues

VERSIONS

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 • 16/Jul/20 4:38 PM

A ...

add item 1

feature B PRJ-5 ↑ 1

add item 2

feature B PRJ-2 ↑ 2

Backlog 1 issue Create sprint ...

add item 3

feature B PRJ-3 ↑ 3

+ Create issue

Issues without epics

Mark an epics 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' under 'projectB'. At the top, there's a search bar, a user icon (A), and filter buttons for 'Only My Issues' and 'Recently'. A context menu is open over an epic titled 'feature B'. The menu has sections for 'EPICS', 'VERSIONS', and 'Color'. It includes links to 'Create issue in epic', 'View linked pages', and 'Issues without epics'. The 'Mark as Done' option is highlighted with a red box. Other visible items in the menu include 'Edit name' and 'View epic details'.

- EPICS
- All issues
- feature B
- PRJ-4 add feature B
- Issues 3
- Completed 3
- Unestimated 0
- Estimate 6
- Create issue in epic
- View linked pages
- Issues without epics

Color

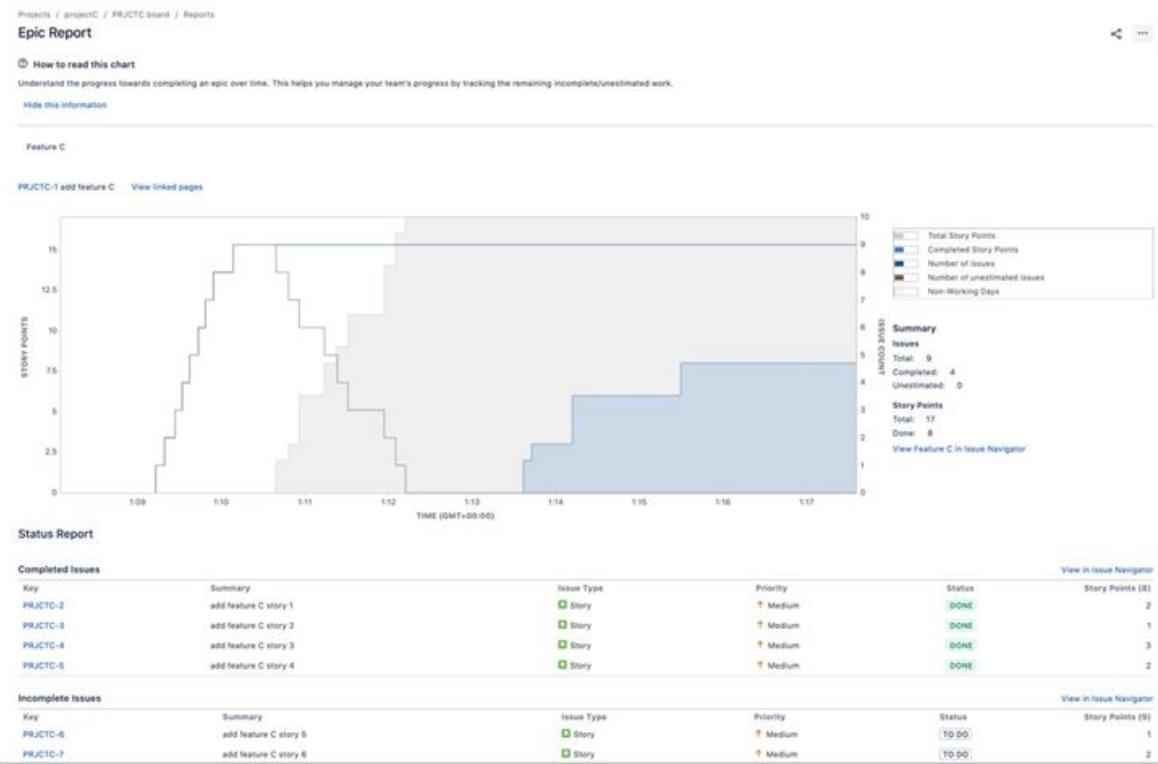
- e e e e e
- e e e e e
- e e

Edit name

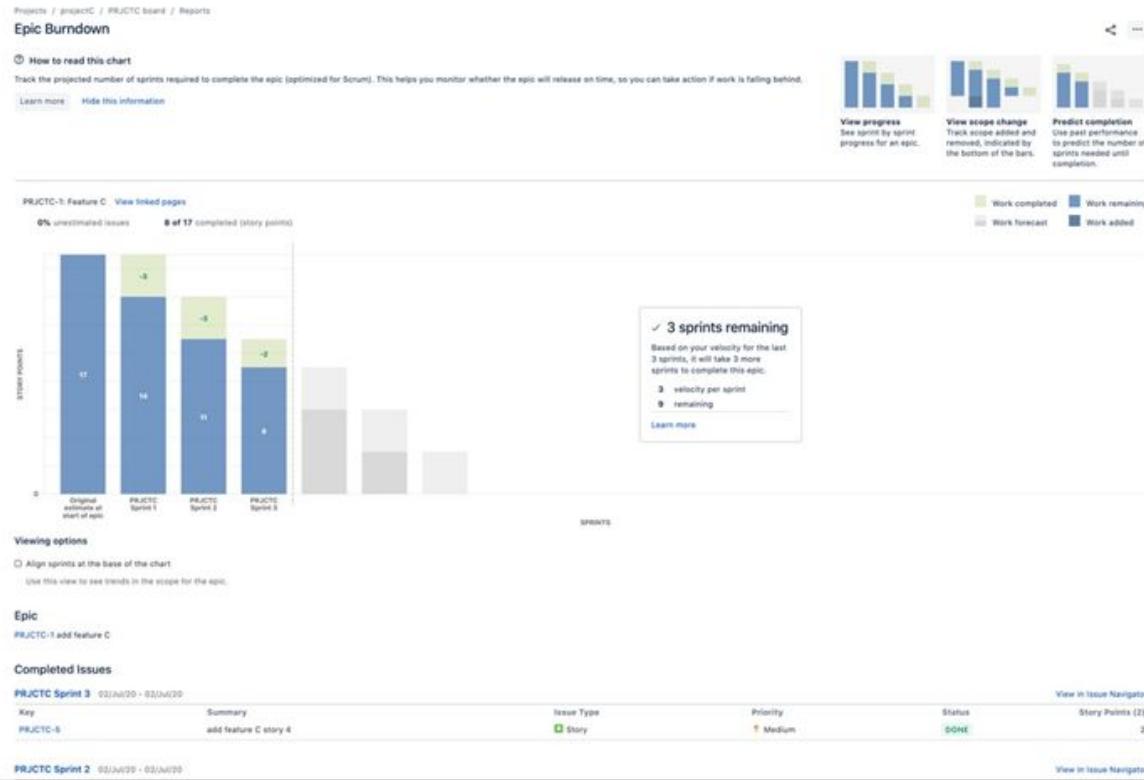
View epic details

Mark as Done

Epic 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 associates issues with on epic
- Epics can be shown on boards or in backlogs

Lab 10 - Epics

- Create an issue of type “epic”.
- 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 Dashboard
- Displays dashboard as a wallboard

Visualizing work



Boards



Search



Reports



Dashboards

Dashboards

The dashboard displays the following information:

- Sprint Health Gadget:** PROJB Sprint 3 - PROJB board. Overall sprint progress (Story Points) is shown as a bar chart with 0% completed. Time elapsed, Work complete, Scope change, Blocked, and Flagged metrics are also listed.
- Sprint Burndown Gadget:** A burndown chart for Nov 3 to Nov 9. The Y-axis is Story Points (0 to 6). The X-axis is TIME. The chart shows a single burndown line starting at 6 points and ending at 0 points by Nov 9. A legend indicates: Guideline (blue line), Remaining Values (red line), and Non-Working Days (grey area).
- Activity Stream:** Your Company JIRA. Shows a list of recent activity items from Alana Grant, such as updating sprints, creating items, and updating story points.

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

Types of gadgets

Add a gadget

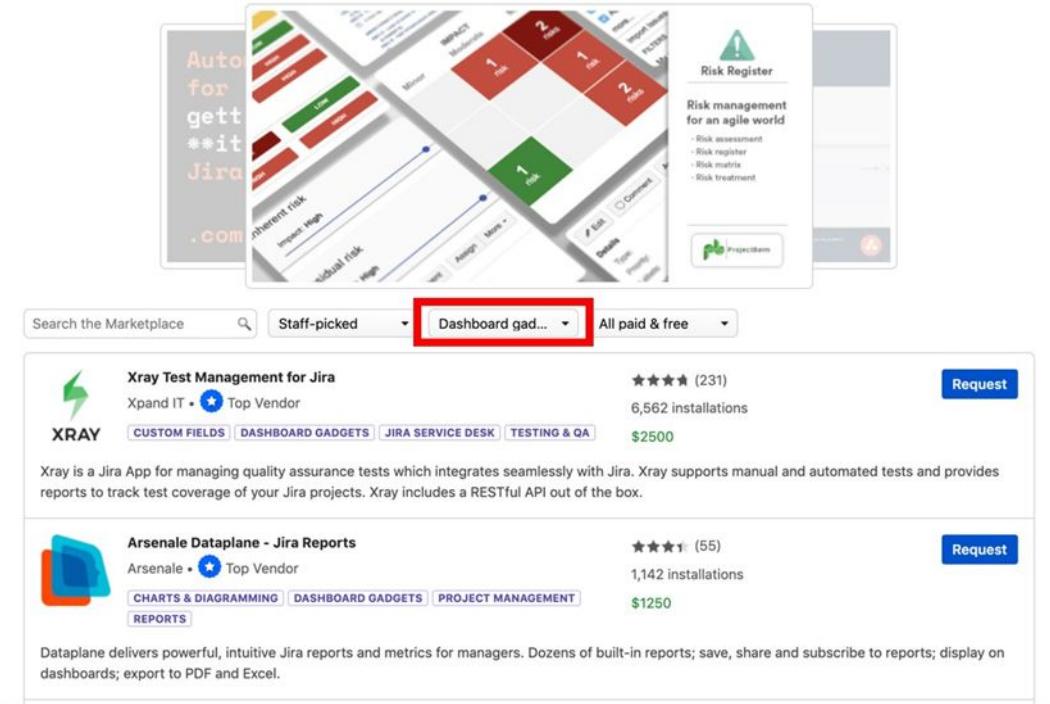
Search

CATEGORIES

All	31
Charts	8
Jira	27
Other	3
Wallboard	7

Marketplace

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, a staff-picked dropdown, and a dropdown menu where 'Dashboard gad...' is selected (highlighted with a red box). Below this, 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 includes tags for CUSTOM FIELDS, DASHBOARD GADGETS, JIRA SERVICE DESK, TESTING & QA. A 'Request' button is present. 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 4-star rating (55 reviews), 1,142 installations, and a price of \$1250. It includes tags for CHARTS & DIAGRAMMING, DASHBOARD GADGETS, PROJECT MANAGEMENT, and REPORTS. A 'Request' button is present. 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

The screenshot shows a user interface for adding gadgets to a dashboard. At the top right are three buttons: "Add gadget", "Edit layout", and "...". Below this is a search bar and a sidebar with categories: All (31), Charts (8), Jira (27), Other (3), and Wallboard (7). The main area displays five gadget cards:

- Activity Stream** By Atlassian • Local. Lists recent activity in a single project, or in all projects. Includes a "Show XML link" button and an "Add gadget" button.
- Agile Wallboard Gadget** By Atlassian • Local. Displays a board as a Wallboard gadget. Includes a "Show XML link" button and an "Add gadget" button.
- Assigned to Me** By Atlassian • Local. Displays all unresolved issues assigned to me. Includes a "Show XML link" button and an "Add gadget" button.
- Average Age Chart** By Atlassian • Local. Displays the average number of days issues have been unresolved. Includes a "Show XML link" button and an "Add gadget" button.

Configuring a gadget

The image shows a configuration dialog for a 'Sprint Health Gadget'. At the top right of the dialog, there is a context menu with the following options: 'Minimize' (with a small icon), 'Delete' (with a trash icon), and 'Edit' (highlighted with a red arrow). Below the menu, the dialog has the following settings:

- Board:** PROJB board
- Sprint:** Next Sprint Due (auto)
- Auto refresh:** Update every 15 minutes
- Show options:** Three checkboxes are checked:
 - Show board name
 - Show sprint name
 - Show assignees

At the bottom left of the dialog are 'Save' and 'Cancel' buttons.

Sharing dashboards

The screenshot shows a user interface for managing a dashboard. At the top right, there is a context menu 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:

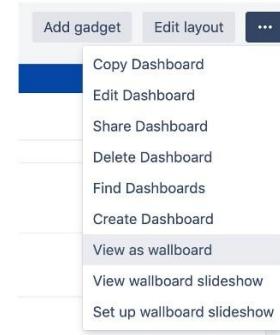
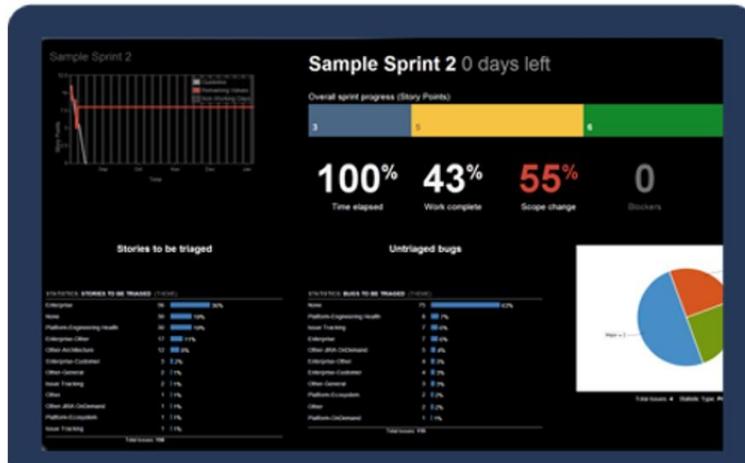
- Name: Team Sprint Dashboard
- Description: (empty)
- Favorite: (star icon)
- Shares: Not shared
- Add Shares: A dropdown menu is open, showing "Project" and "Teams in Space" as current share levels. A sub-menu is displayed with "All" selected (indicated by a blue background and a checked checkbox). Other options in the sub-menu include Project Roles, Administrators, Developers, and Users. A "+ Add" button is also visible.

Below the shares section, a message states: "Shared with everyone with permission to browse the 'Teams in Space' space".

At the bottom are two buttons: Update and Cancel.

Wallboards

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



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 11-Dashboards

- Create a dashboard
- Display a dashboard as a wallboard

Putting it all together



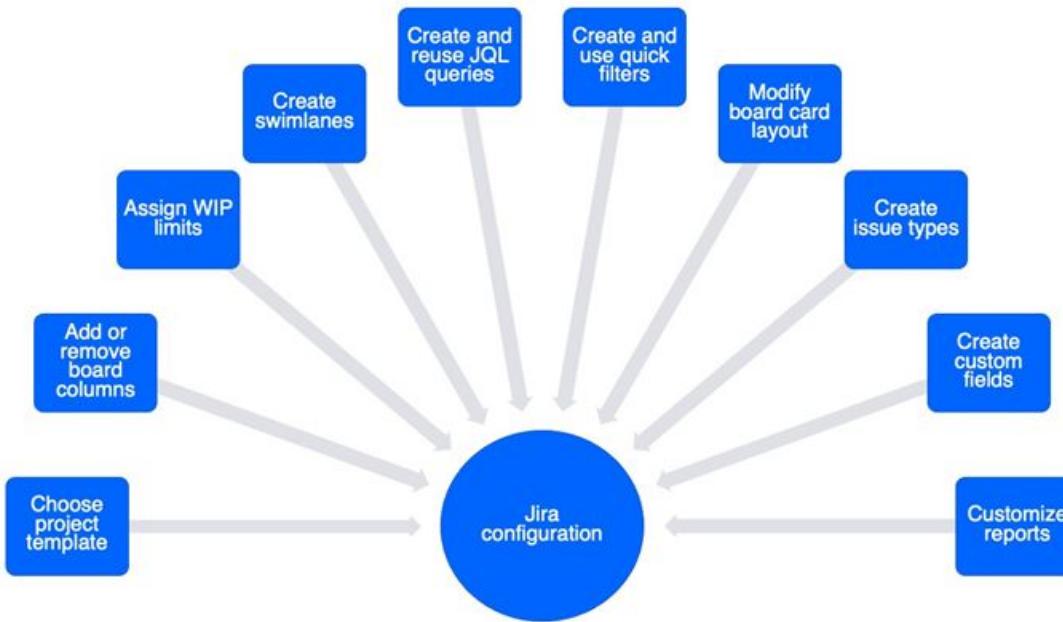
Topics

Quick Course Overview

Jira Family
Wrap up



Ways to configure Jira to match your team's process



Topics

Quick Course Overview
Jira Family
Wrap up



Jira Family of products



Jira Core

Business project management software



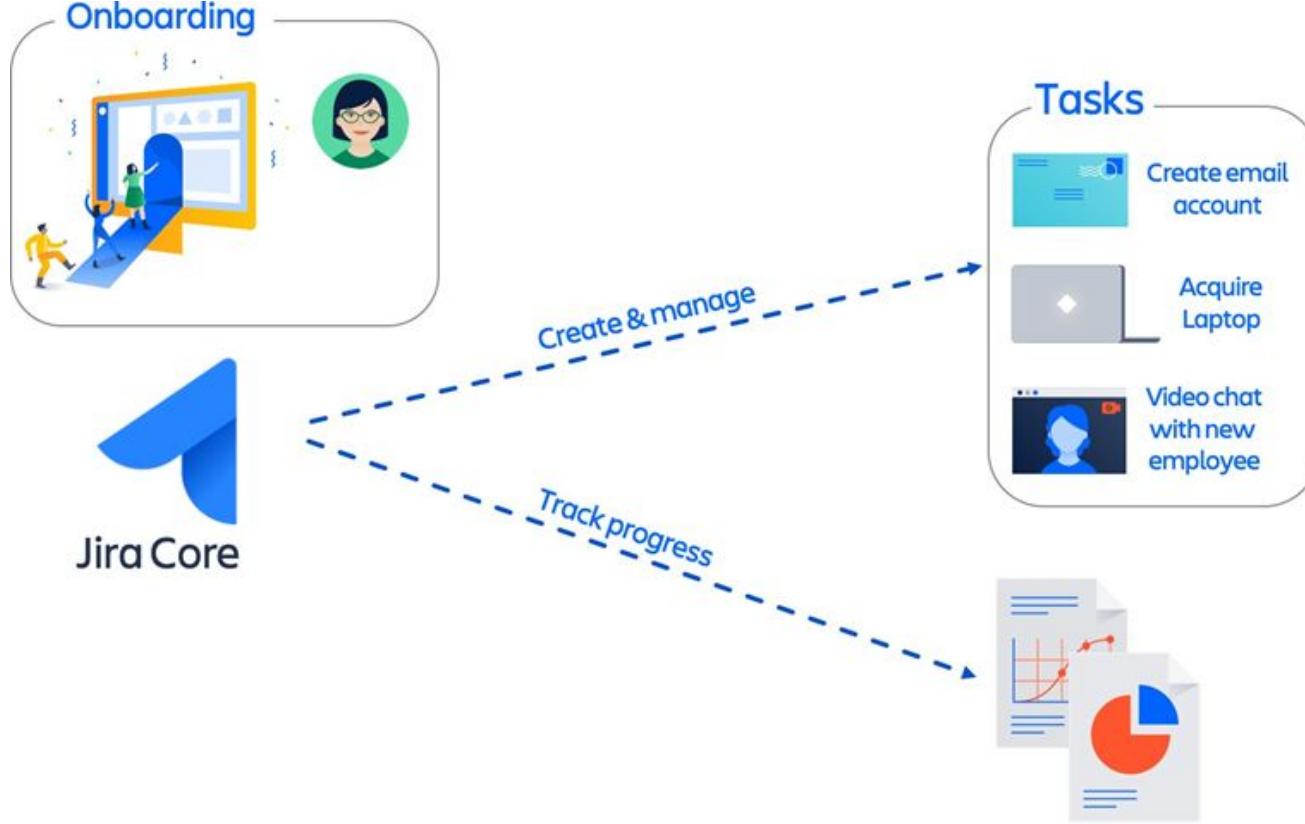
Jira Software

Plan, track, and release software

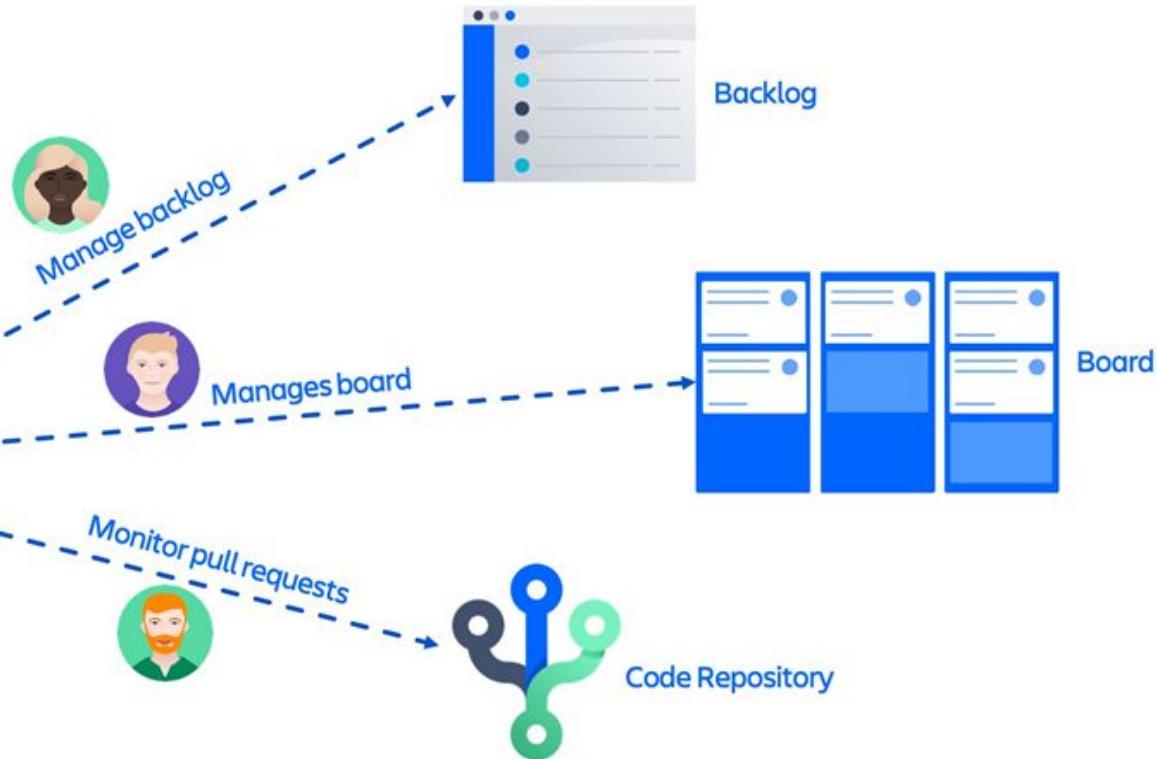
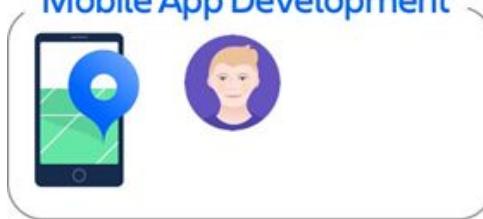


Jira Service Desk

Service desk software for IT teams

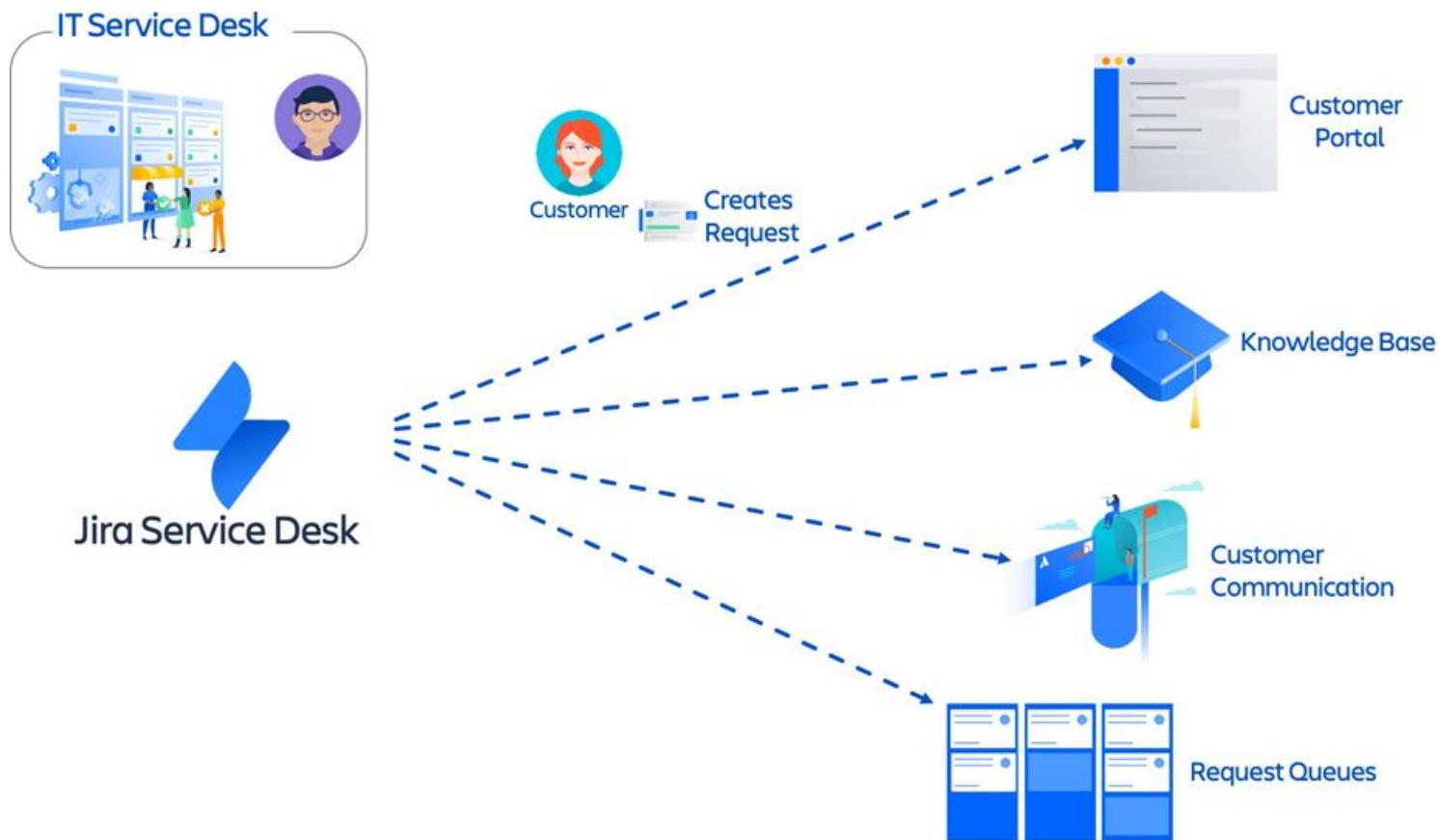


Mobile App Development

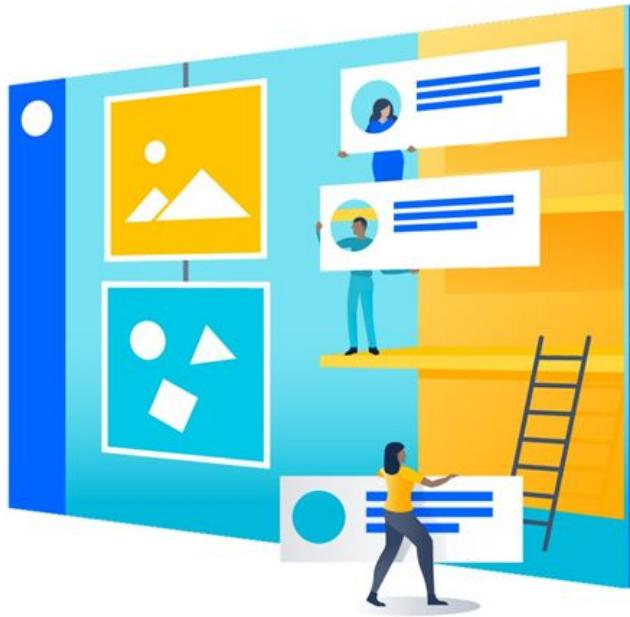


Jira Software

Code Repository



Marketplace



Lab 12 – Final Project (Optional)

- Modify a Jira project to meet your team's process

Topics

Quick Course Overview
Jira Family
Wrap up



Earn Learning Voyage Certifications!!



Propel your career



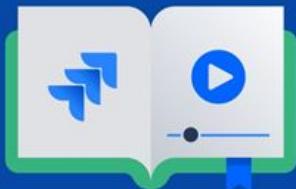
Boost your skills



Help your teams do
their best work

Training for Jira

Marketplace App



- Interactive tutorials
- Gets teams using Jira quickly
- Covers popular topics and tasks
- Introduces essentials concepts
- Showcases in-product demos
- Non-graded quizzes to track learning
- Available for Cloud, Server, Data Center

Congratulations on completing the course!

