

# Agile Project Management Using Jira



# COURSE OVERVIEW



# What you will learn

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

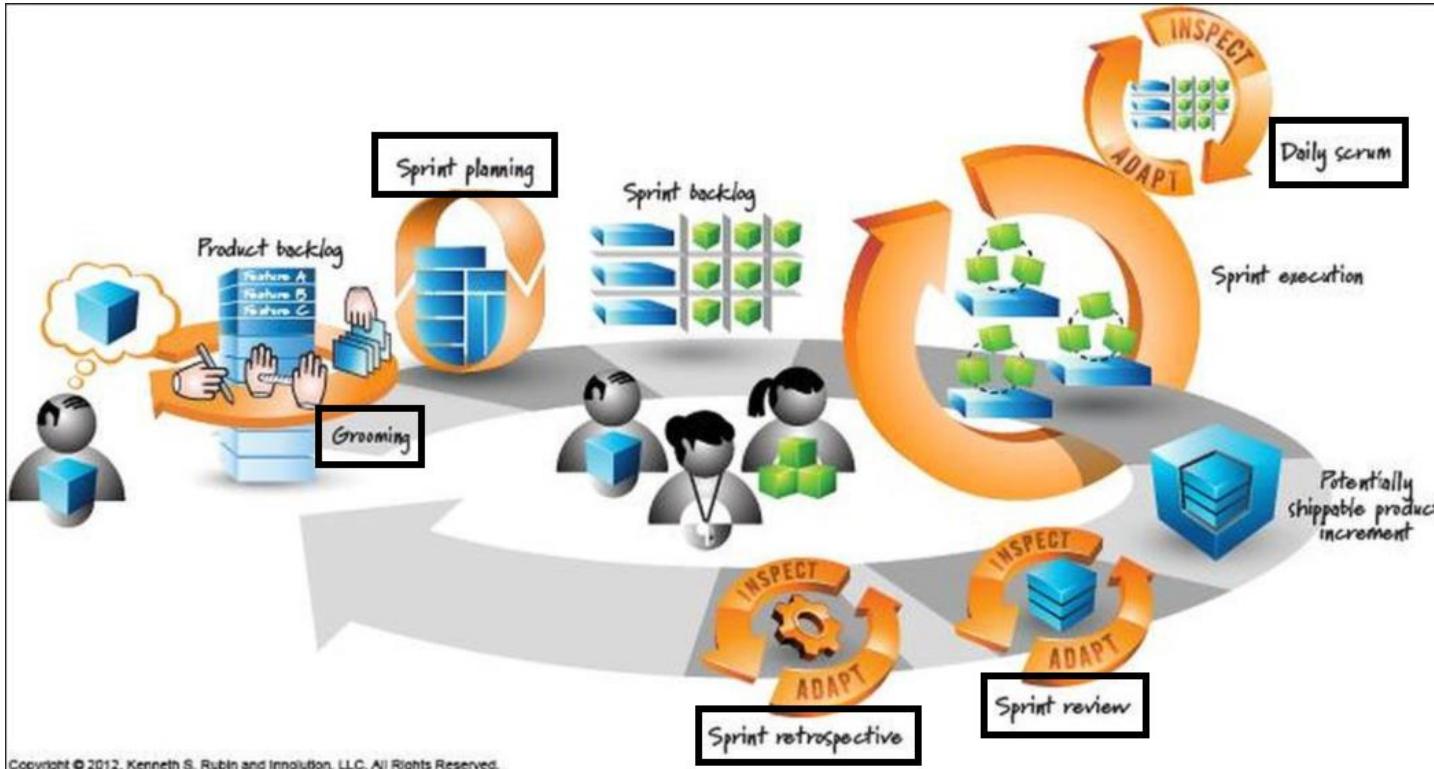
# To succeed here, you need to have

- No knowledge of agile or Jira is assumed



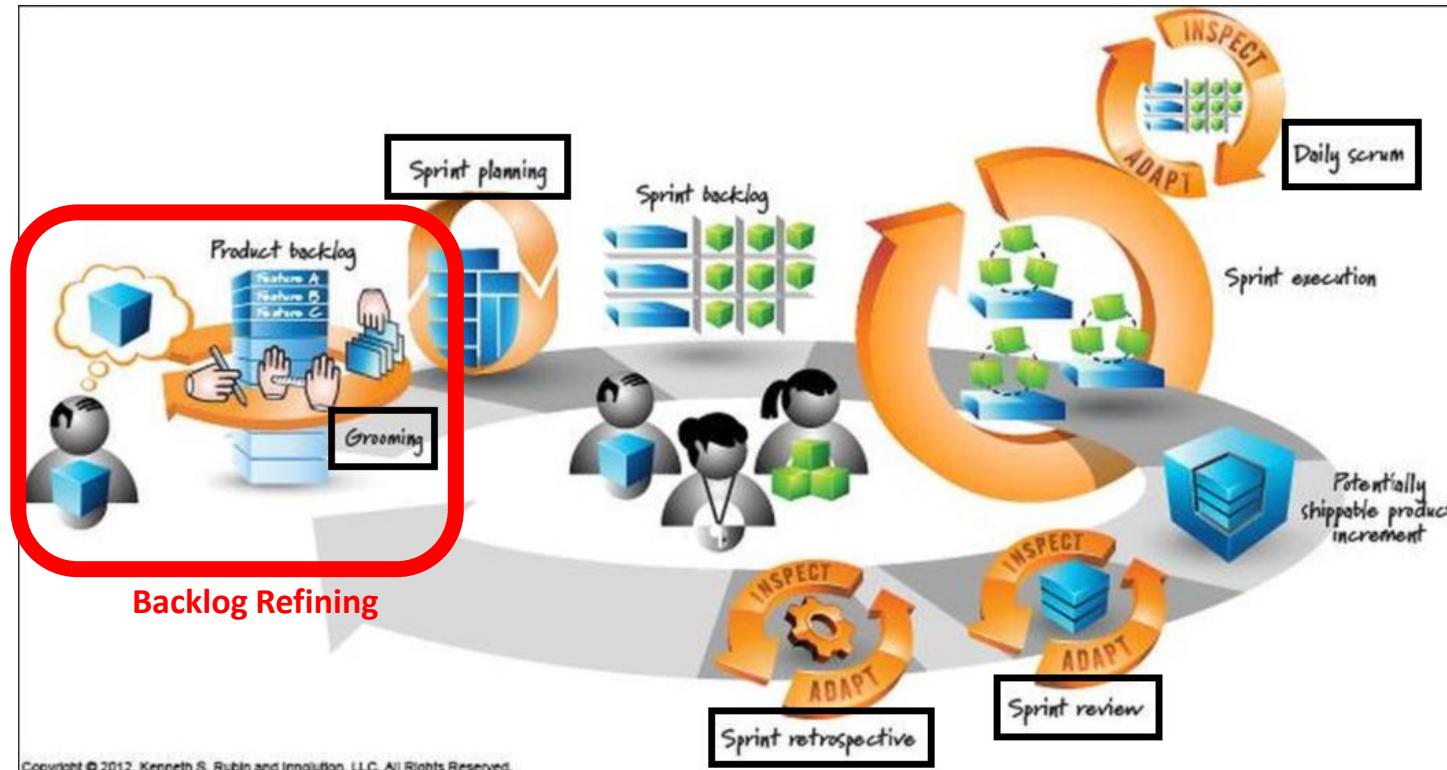
# Agility & Ways of Working

# Overall Scrum Process



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

# Backlog Refining



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

# Backlog Refining

## 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)

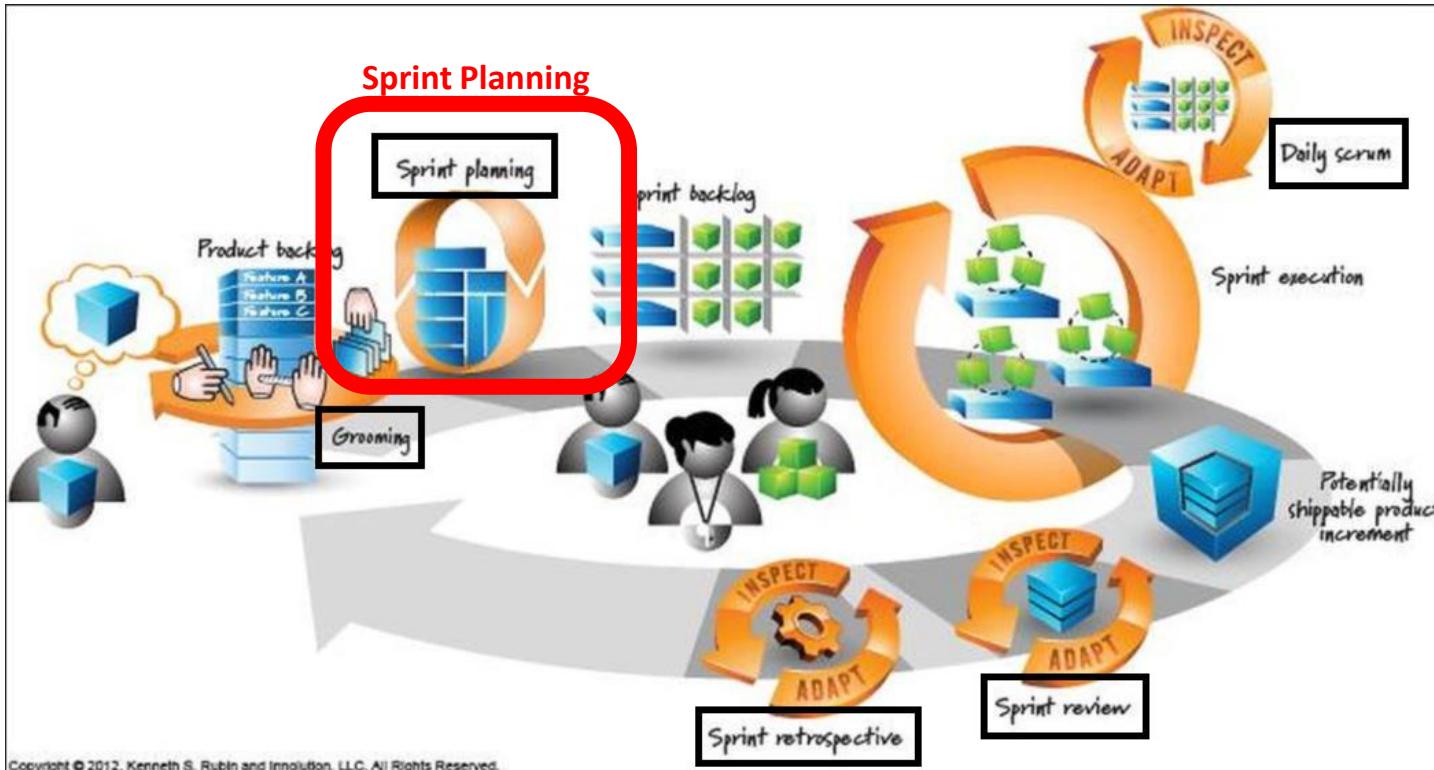
## Success Factors

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

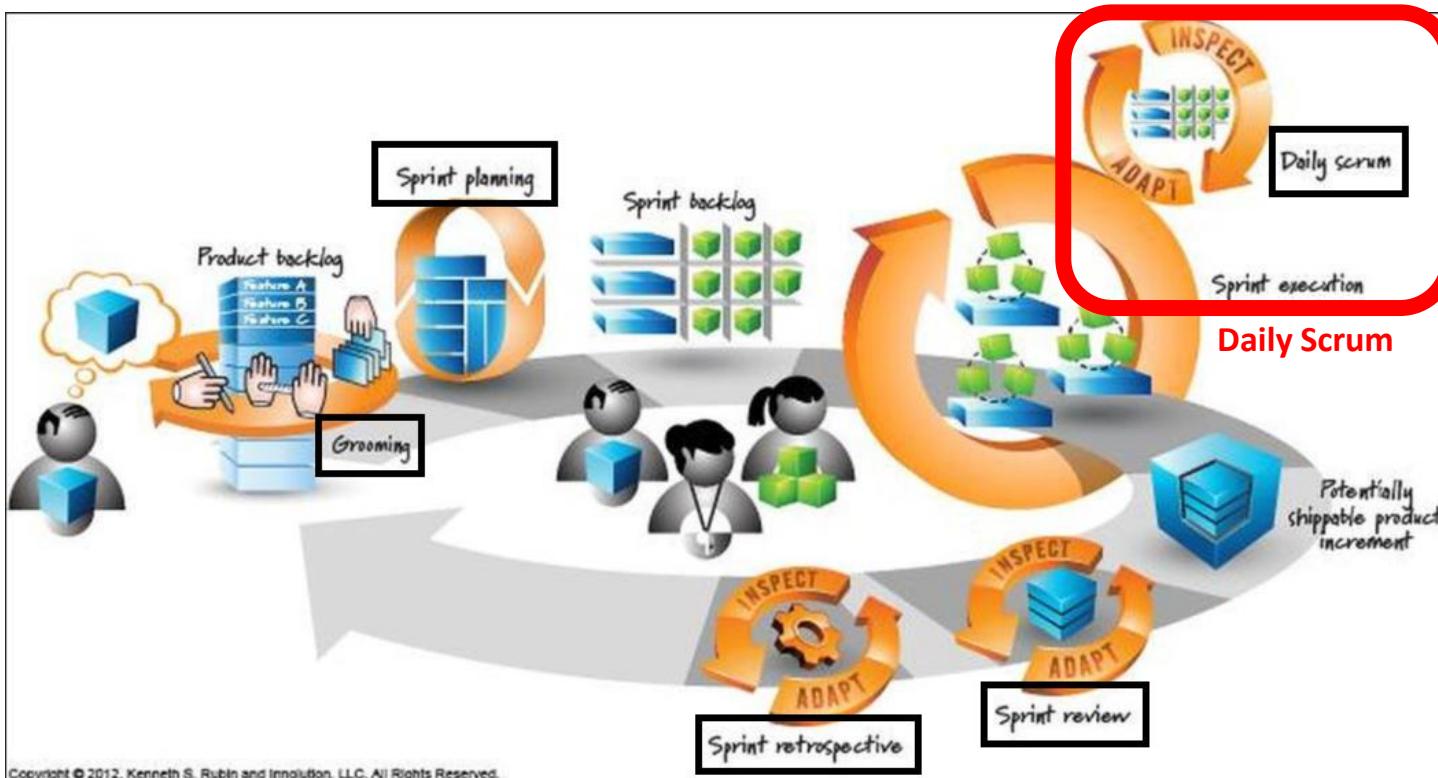
## Success Factors

- 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

## 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

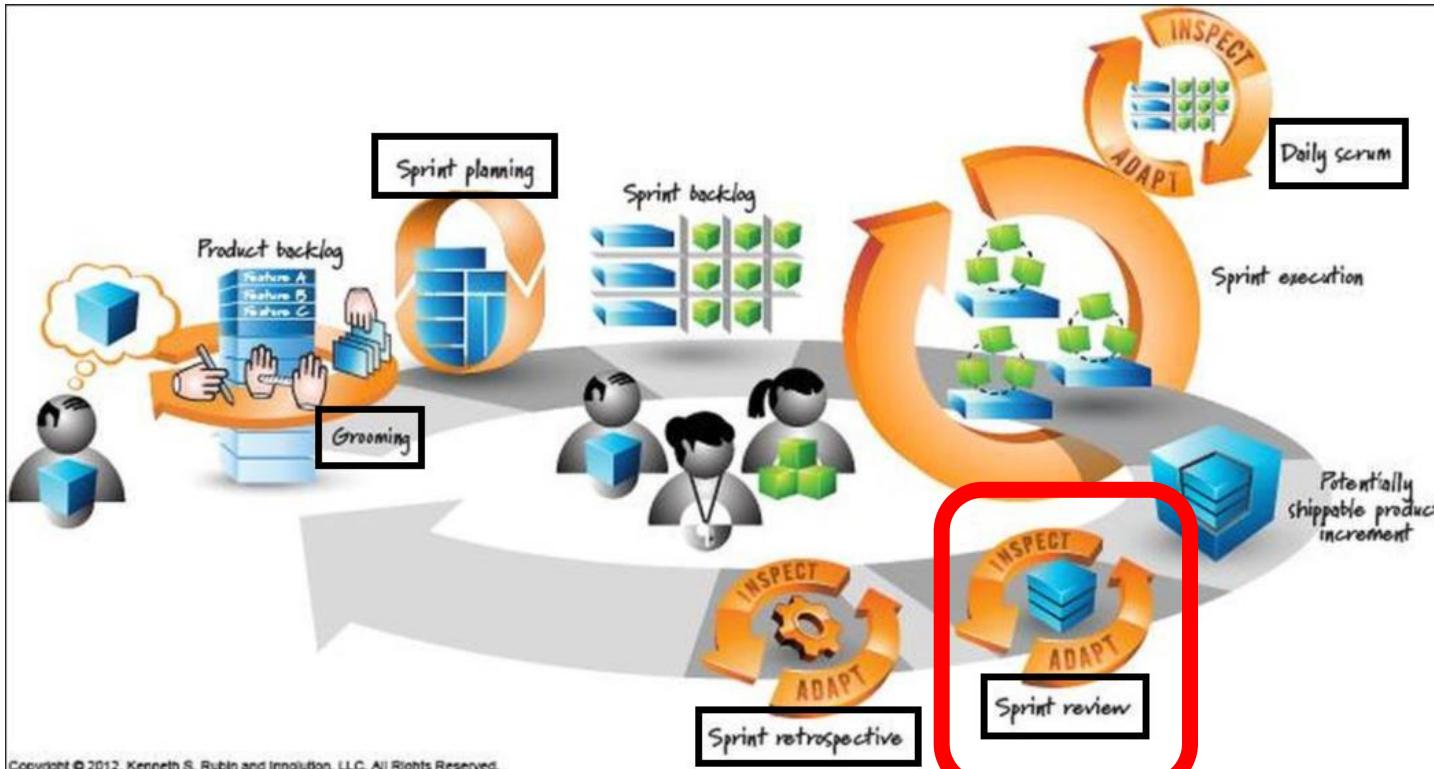
## Pitfalls

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

## 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

# Sprint Review



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

**Sprint Review**

Enterprise Tools  
Engineering  
TechOps

# 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

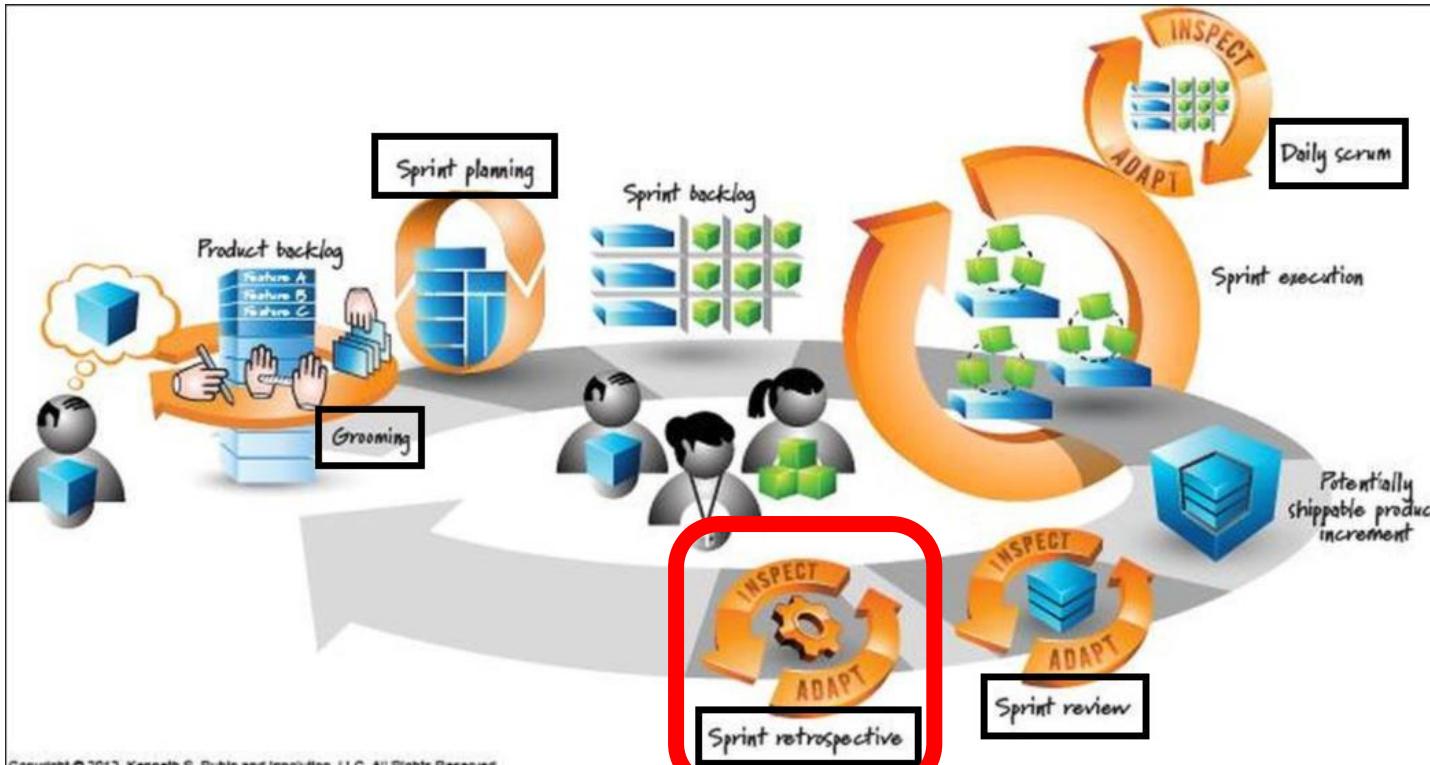
## Success Factors

- 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?

# 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

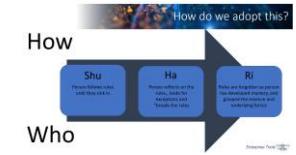
Student

Apprentice

Master

# 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

How do we adopt this?

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 Team
- 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

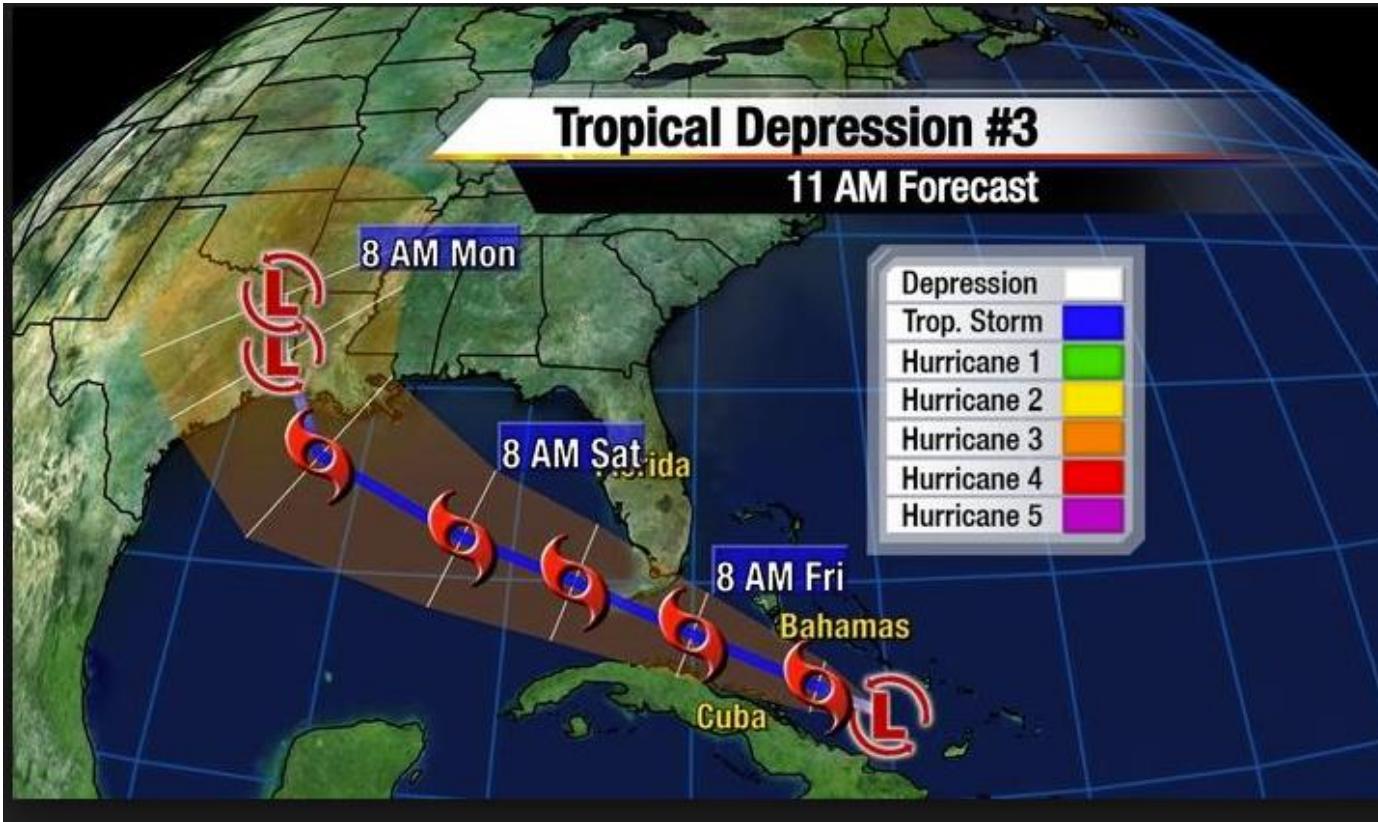


# Team and Scope

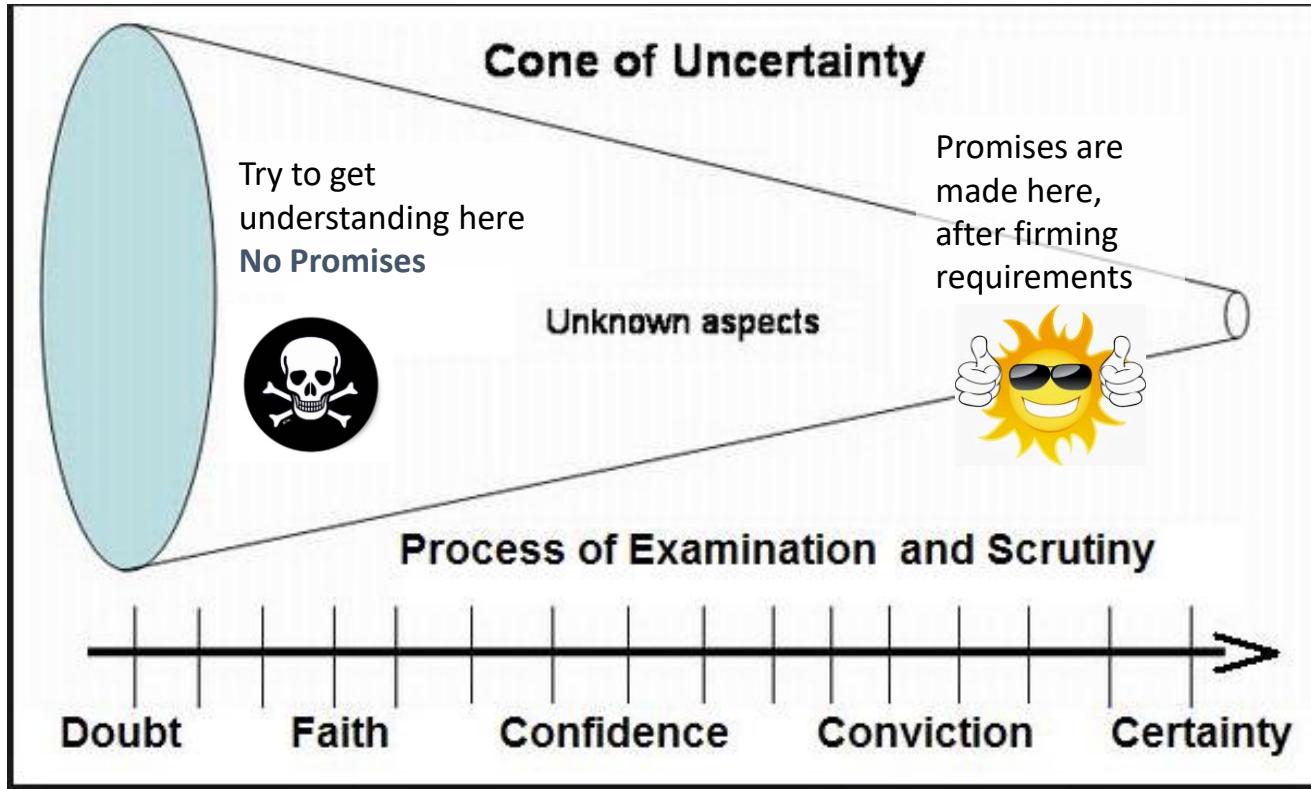
## Team Roles

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

# Cone of Uncertainty



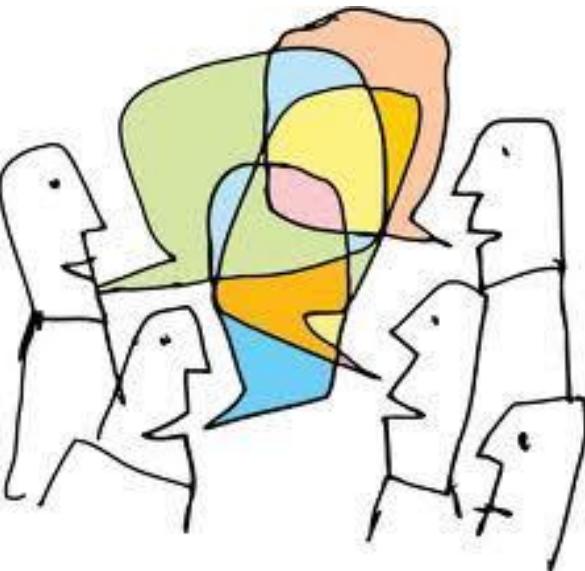
# Cone of Uncertainty





# User Stories

# Helping to solve the Business Problem...



Start with an Epic...



# Story Writing

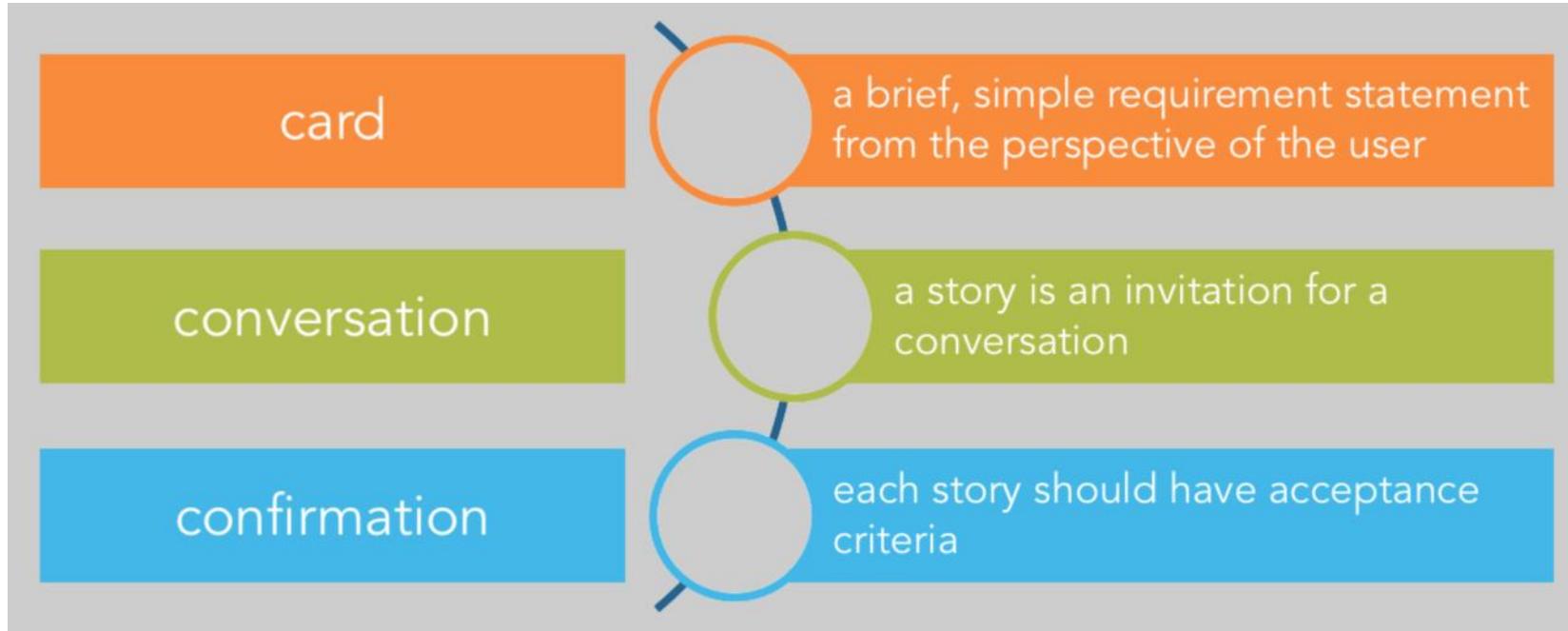
- A user story is not a specification, it is a communication and collaboration tool to help explain the why.
  - *Ron Jefferies - A Card, Conversation and Confirmation.*
- The Product Owner should not hand stories off to the development team:
  - They should rather be part of a conversation together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



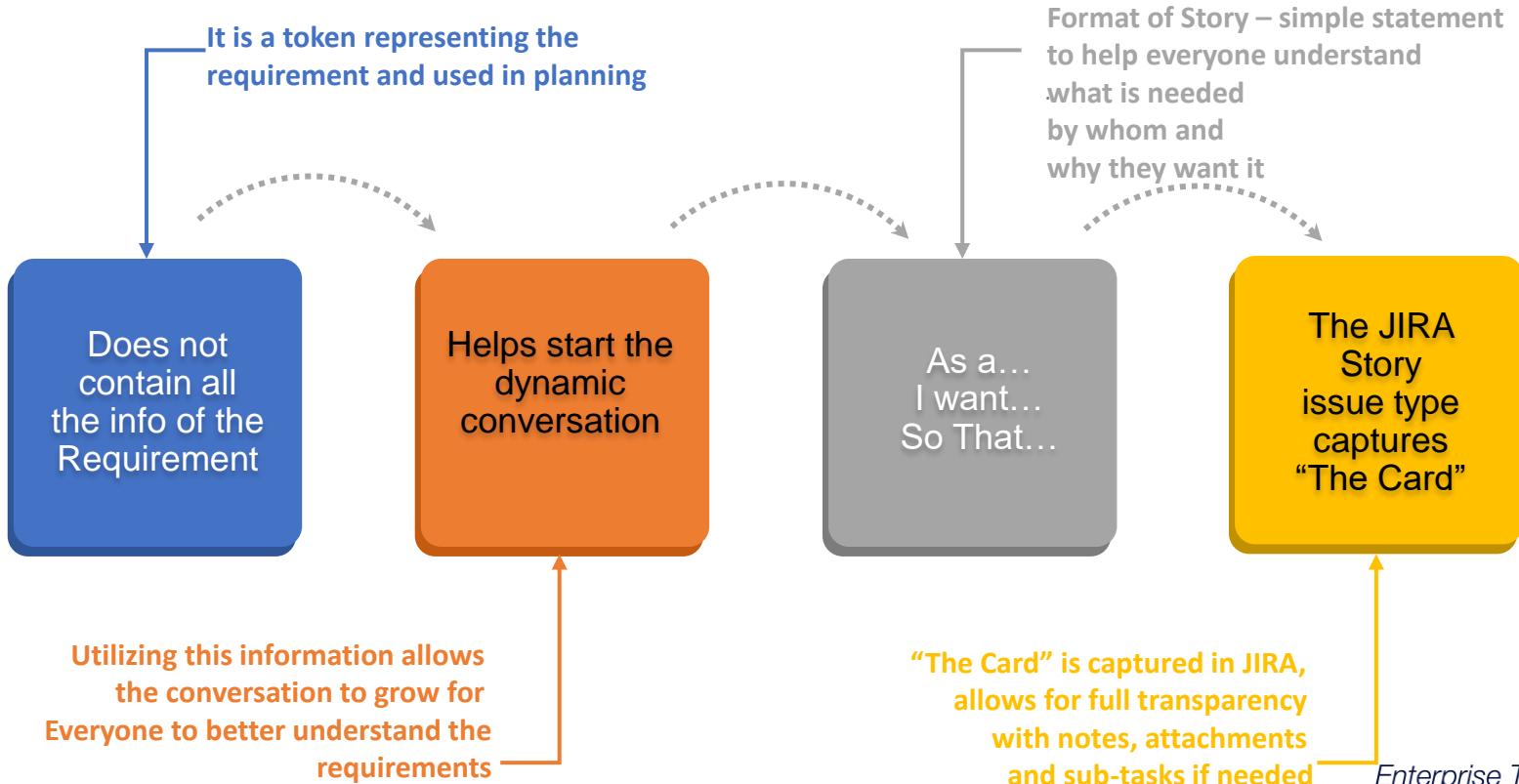
# Structure of a Story INVEST

<u>Independent</u>	Can deliver value by itself. We want to be able to develop in any sequence.
<u>Negotiable</u>	Avoid too much detail; keep them flexible so the team can adjust how much of the story to implement.
<u>Valuable</u>	Users or customers get some value from the story.
<u>Estimatable</u>	The team must be able to use them for planning
<u>Small</u>	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>Testable</u>	Document acceptance criteria, or the definition of done for the story which leads to test cases

# Structure of a Story



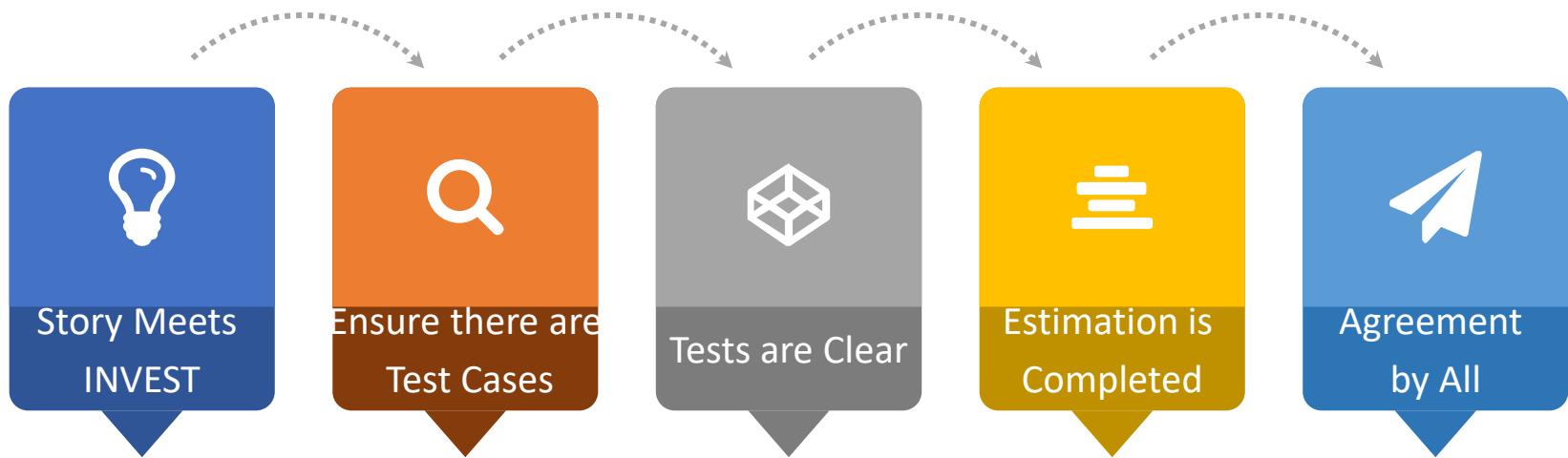
# The Card



# The Conversation



# The Confirmation



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.

## 1<sup>st</sup> Feedback from the field

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



# Story Overview cont.

## 1<sup>st</sup> 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

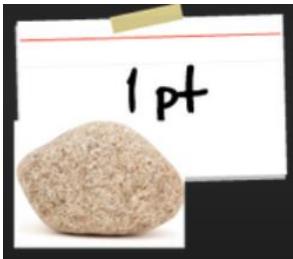
## Agile Estimation (Fibonacci) and Planning

- 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

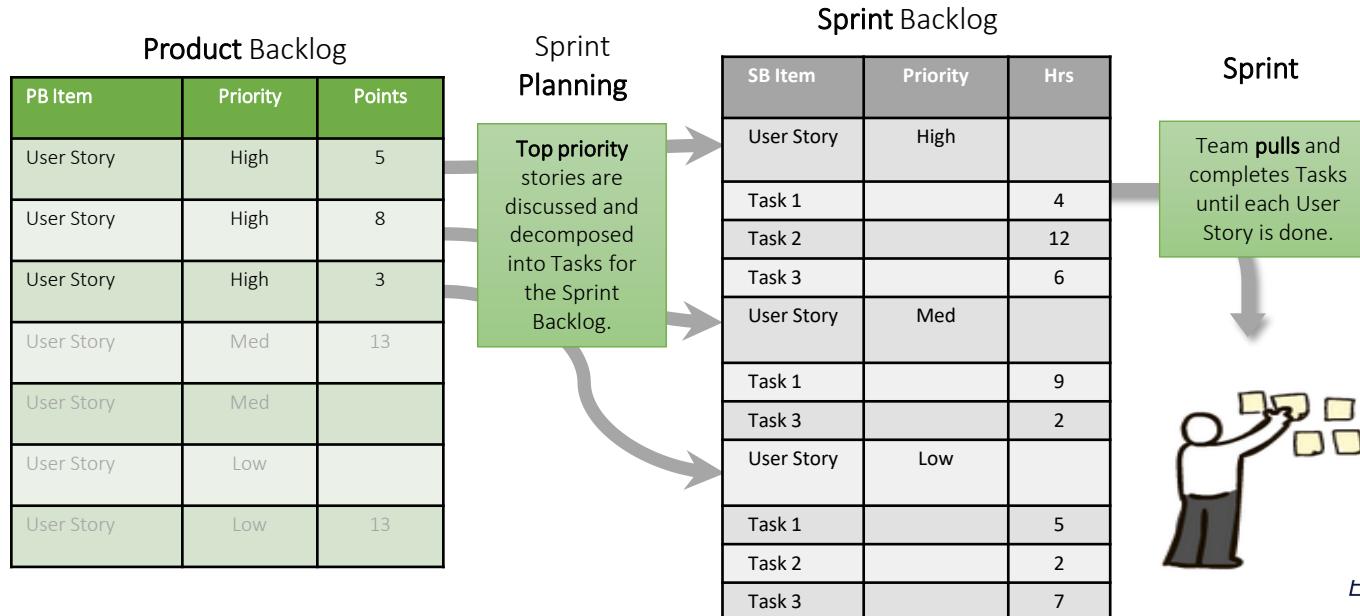
The Sprint Report shows the list of issues in each sprint. It is useful for your Sprint Retrospective meetings, and also for mid-sprint progress checks..

### 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](#)

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

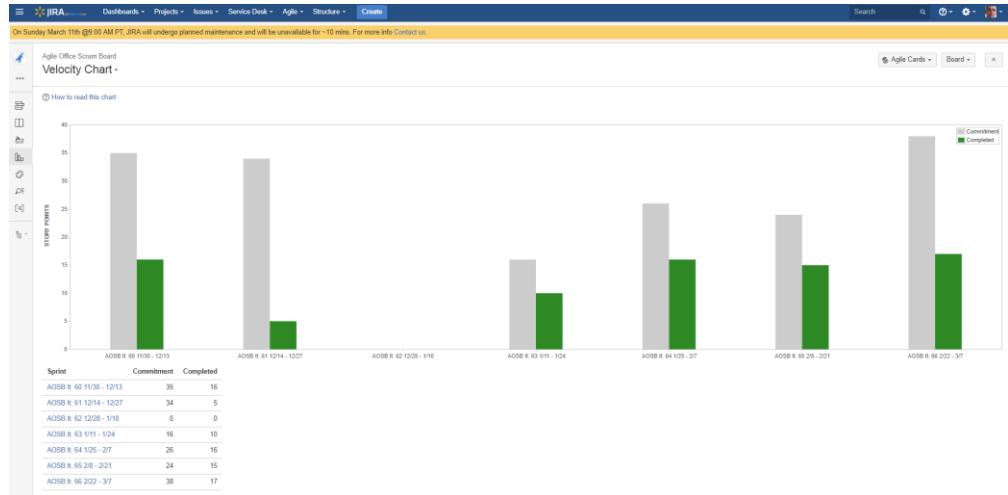
[View in Issue Navigator](#)

**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!



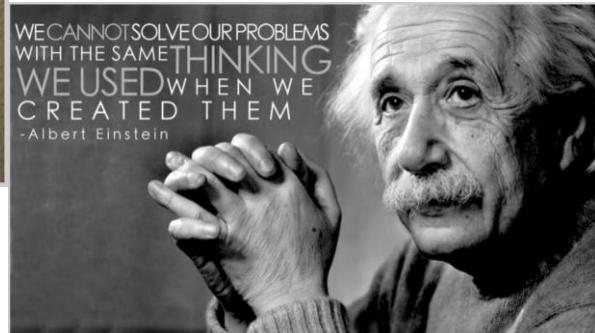
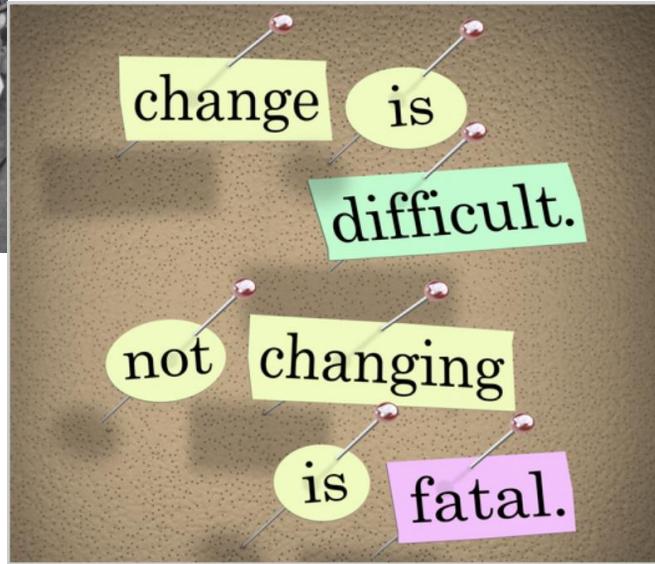
# Why do Retrospectives

**"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

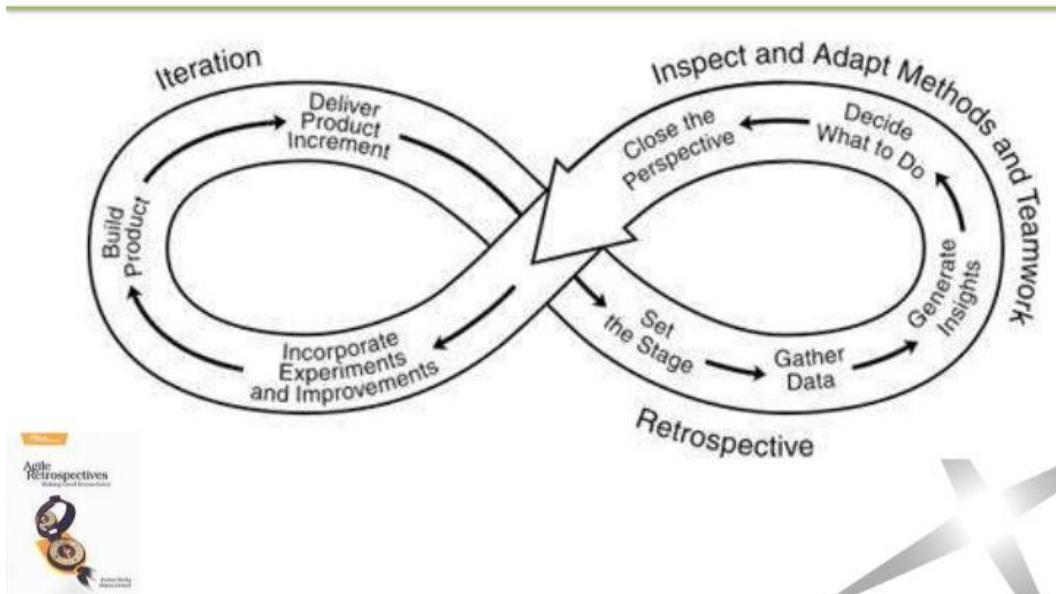


# Tip prior to Retrospective



# 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

Enterprise Tools  
Engineering  
TechOps

# 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

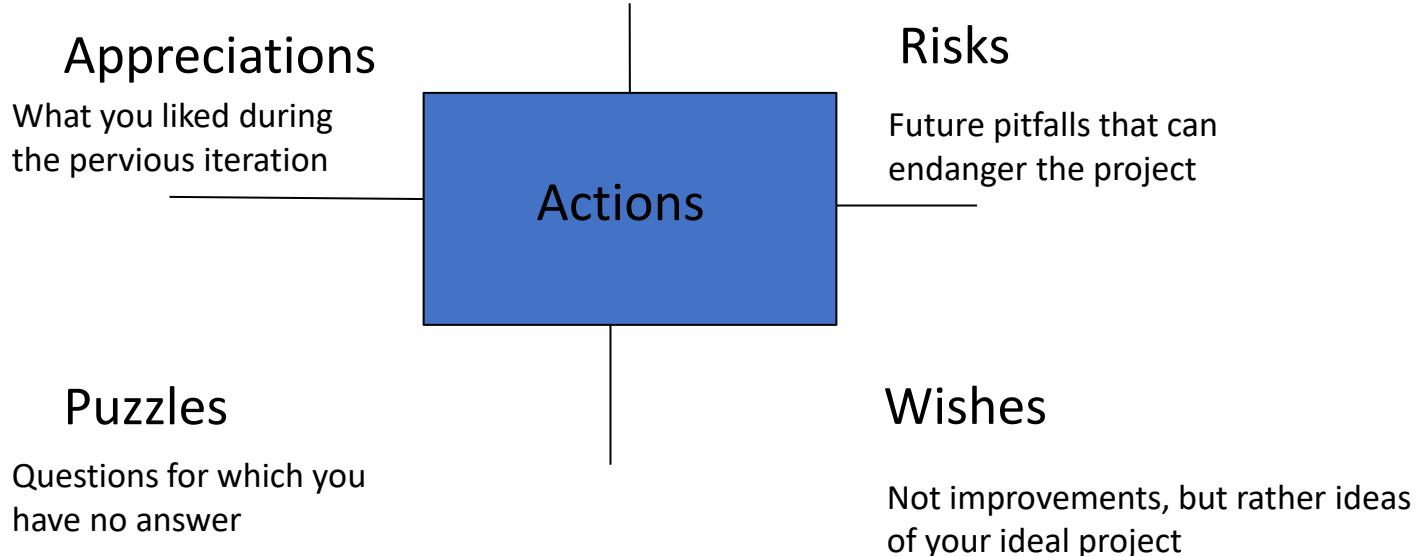


Flower bouquet for people  
you wish to appreciate



Light bulb for new ideas to try

# Gathering Data - Actions



# Gathering Data - Sailboat

## TRADEWINDS



Items that you want  
to capitalize on -  
What is going well and  
should we continue  
doing?



What helps you move  
forward - Who/what  
should we Appreciate?



## SAILS

## STORMS



Items that could  
threaten your  
success - What are  
the Risks?

What is slowing you  
down - What can be  
improved?



## ANCHORS

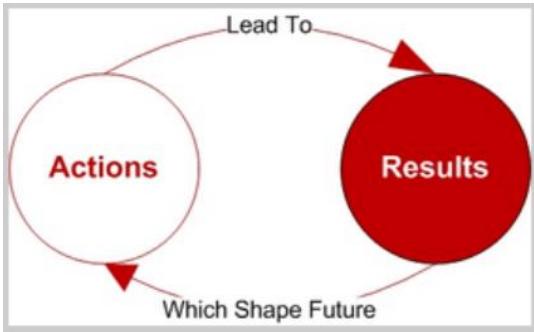
# Gathering Data – Start, Stop, Continue



- 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



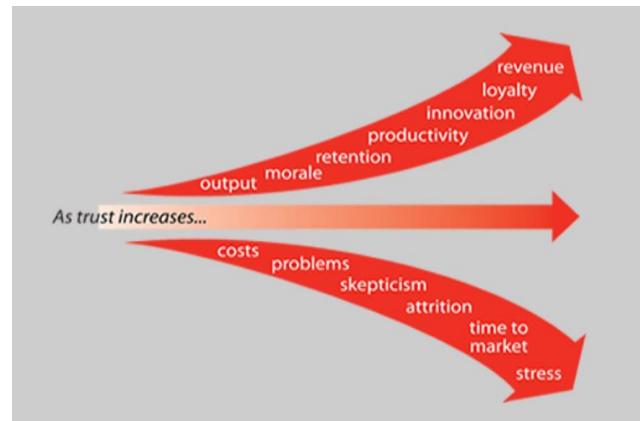
Appreciate contributions



Identify ways to make the next retrospective better

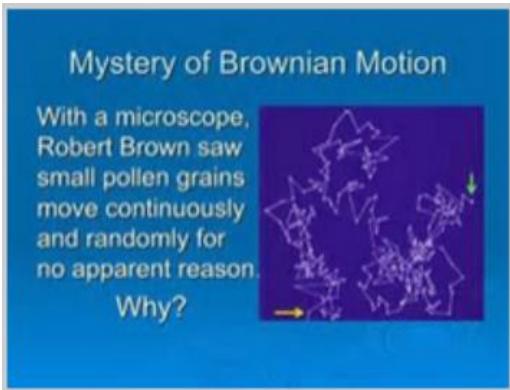


# Benefits of doing Retrospective

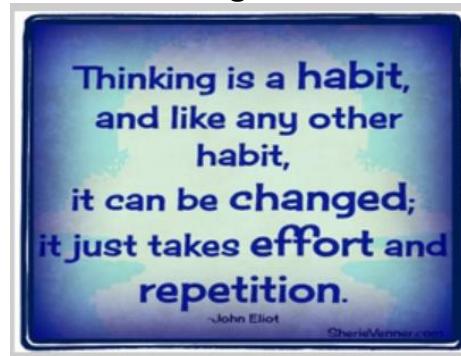


# Leave out...what happens

## Setting the Stage:



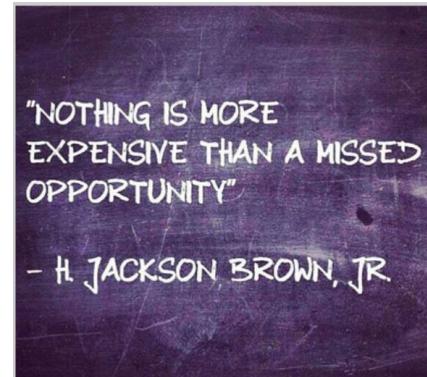
## Generating Discussion:



## Gathering Data:



## Closing the Retrospective:



## Deciding What to Do:



# How

How do we adopt this?

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

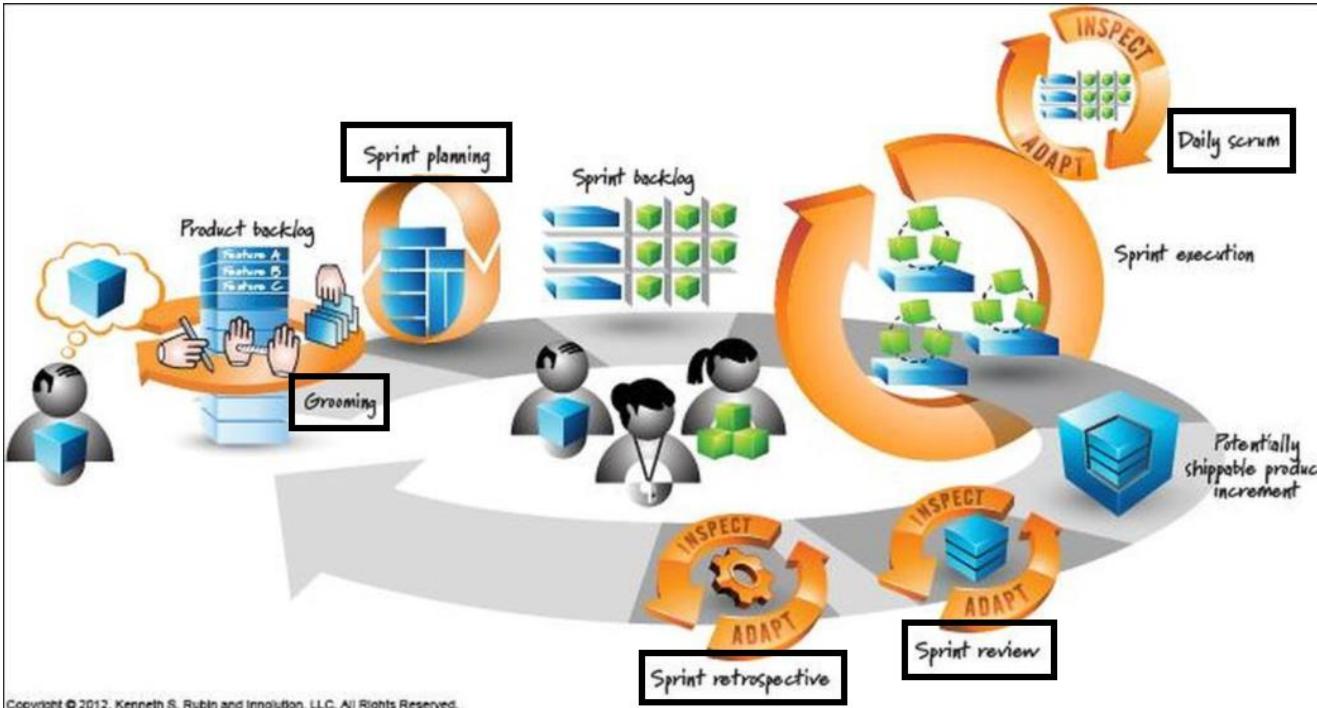
Rules are forgotten as  
person has developed  
mastery, and grasped the  
essence and underlying  
forces

Who  
Student

Apprentice

Master

# Better Understanding - Scrum Framework



# Live teach schedule (1 of 2)

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

# Live teach schedule (2 of 2)

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

# Jira Cloud vs Jira Server/Data Center

A screenshot of the Jira Cloud interface showing a Kanban board. The board has three columns: 'To do', 'In progress', and 'Done'. There are several cards in each column, each with a title, a description, and a small icon. A sidebar on the left contains links for 'Issues', 'Reports', and 'Project settings'. The top navigation bar includes 'Work', 'Projects', 'Filters', 'Dashboards', 'People', and 'Apps'.

Cloud

A screenshot of the Jira Server/Data Center interface showing a Kanban board. The board has three columns: 'To do', 'In progress', and 'Done'. The cards are more detailed than in the Cloud version, showing attachments and more descriptive text. A sidebar on the left contains links for 'Issues', 'Reports', 'Components', and 'Project settings'. The top navigation bar includes 'Work', 'Projects', 'Filters', 'Dashboards', 'People', and 'Apps'.

Server /  
Data Center

# Jira Cloud- classic vs. next-gen projects

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

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

**Next-gen project interface:** A screenshot of the Jira Cloud interface showing a Kanban board titled "Project 1 / prod01 / PROJ board". The board has four columns: "In Progress", "Selected for Development", "Backlog", and "Done". Each column contains three items labeled "add item 1", "add item 2", and "add item 3". The "In Progress" and "Selected for Development" columns also contain a "Details" link.

**Next-gen project interface (bottom):** A screenshot of the Jira Cloud interface showing a Kanban board titled "Project 1 / prod01 / PROJ board". This view shows more detailed cards with icons and labels like "label", "Type", and "Status". The cards are organized into columns: "In Progress", "Selected for Development", "Backlog", and "Done". Each card has a "Details" link.

# Lab 1- Course Overview

- Decide If you want to do the Cloud or Server version of the course

# Agile and Jira Overview



# What will you learn?

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

# Topics

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



# What is Agile?

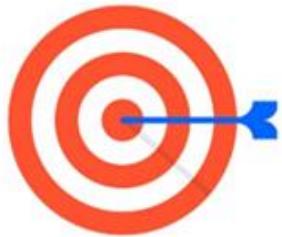
A way of getting things done.

An empirical approach to project management

Continuously develop the plan, process and product

A mindset

# Why Agile?



## Effectiveness

Perform better than traditional projects



## Empower the Team

Leverage team knowledge and increase job satisfaction



## Manage Complexity

Simple project management approach to increasing complexity

# What is an agile mindset?

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



# What is an agile mindset?

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



# The agile coach



## BROWSE TOPICS

### Agile manifesto

- › Scrum
- › Kanban
- › Agile project management
- › Product Management
- › Agile at scale
- › Software development

## What is Agile?

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

# Topics

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



# What is Jira

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

# Why Jira?

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

# How does Jira relate to an agile mindset?

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

# Topics

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



# What is a Jira issue?

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



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)



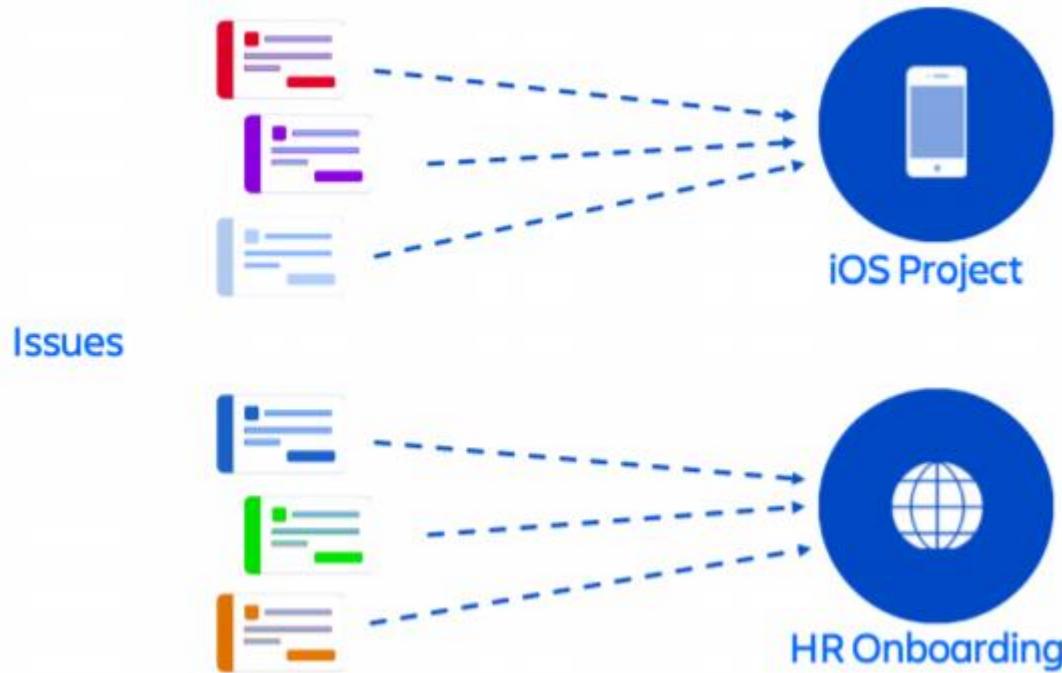
# Issue Key



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

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

# Each issue belongs to one project



# What is a project board?

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



# Jira User Types



Jira Administrator

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



Jira Project  
Administrator

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



Team Member

..... Works on projects

# Takeaways

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

# Lab 2 -Agile and Jira Overview

- Create a project
- Create issues



# Visualize work using project boards



# What will you learn?

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

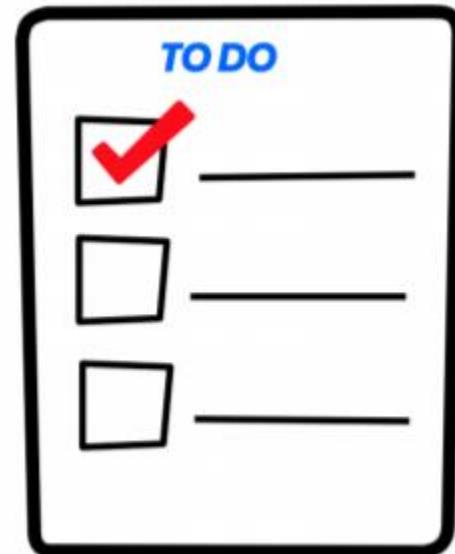
# Topics

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



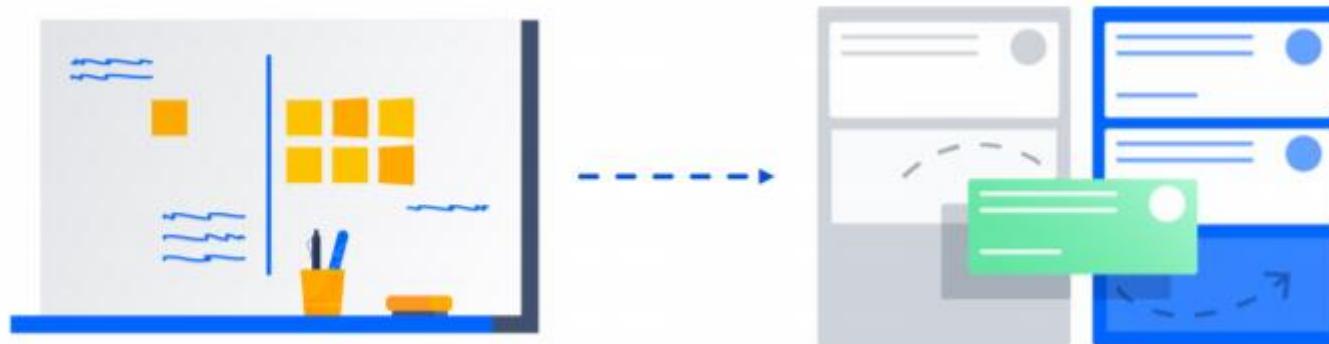
# Virtualizing work: a “to do” list

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



# Visualizing work: a board

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



# Visualizing work: reports and dashboards



Reports



Dashboards

# Why Visualize Work?

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



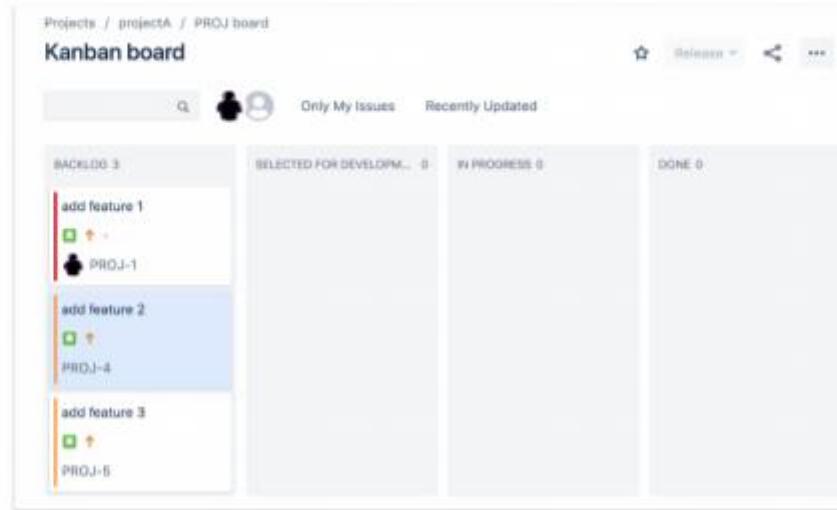
# Topics

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

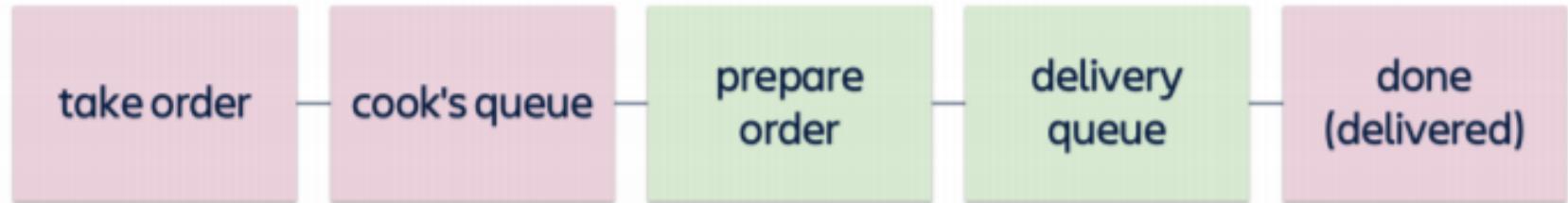


# Workflows

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

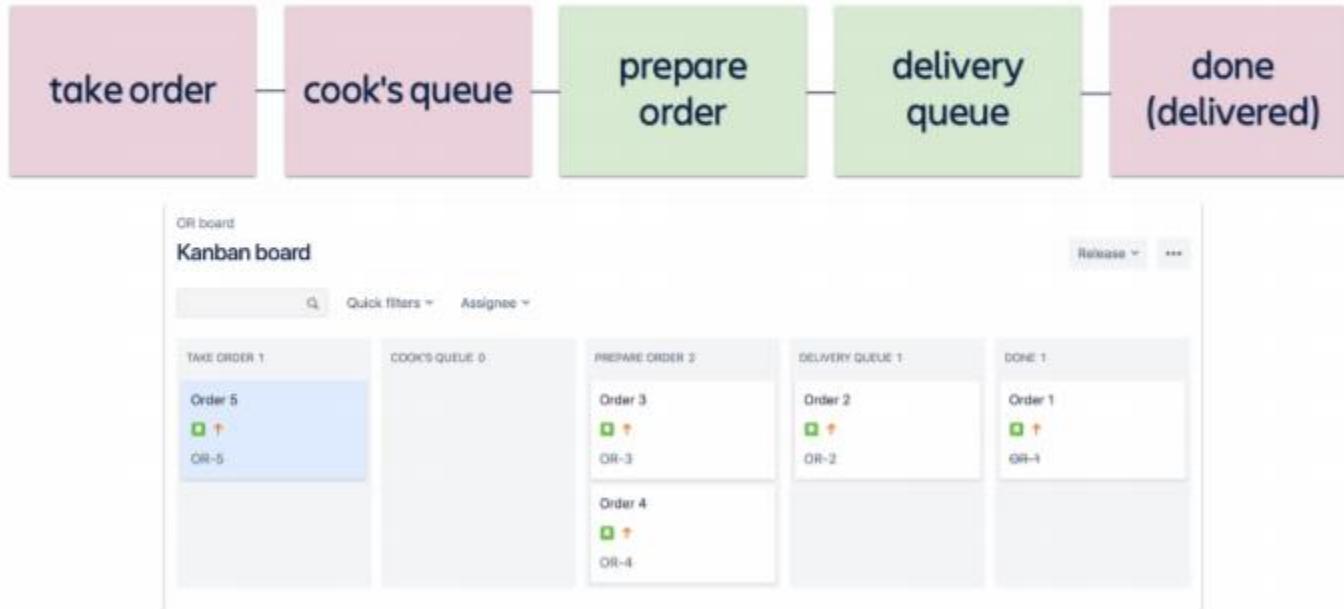


# Example workflow: restaurant order and delivery



# Boards VS Workflows

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



# Topics

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



# How are boards created?

- Automatically
- Create additional boards at any time

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

Projects / projectA / PROJ board

## Kanban board

Only My Issues   Recently Updated

BACKLOG 3

add item 1

PROJ-1

SELECTED FOR DEVELOPME... 0

IN PROGRESS 0

Board settings

Create board

Hide menus

Show detail view

Open issues in sidebar

Expand all swimlanes

# An issue's status field

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

The screenshot shows a Jira interface for creating a new issue. On the left, there is a checkbox labeled "PROJ-1". At the top right, there are icons for "Give feedback" (with 1), a person icon, and a link icon. Below these are buttons for "Backlog" (with a red arrow pointing to it), "SELECTED FOR DEVELOPMENT", "IN PROGRESS", and "DONE".

add item 1

PROJ-1

Give feedback 1

Backlog ▾

SELECTED FOR DEVELOPMENT

IN PROGRESS

DONE

Attachment

Create subtask

Link issue

...

Description

Add a description...

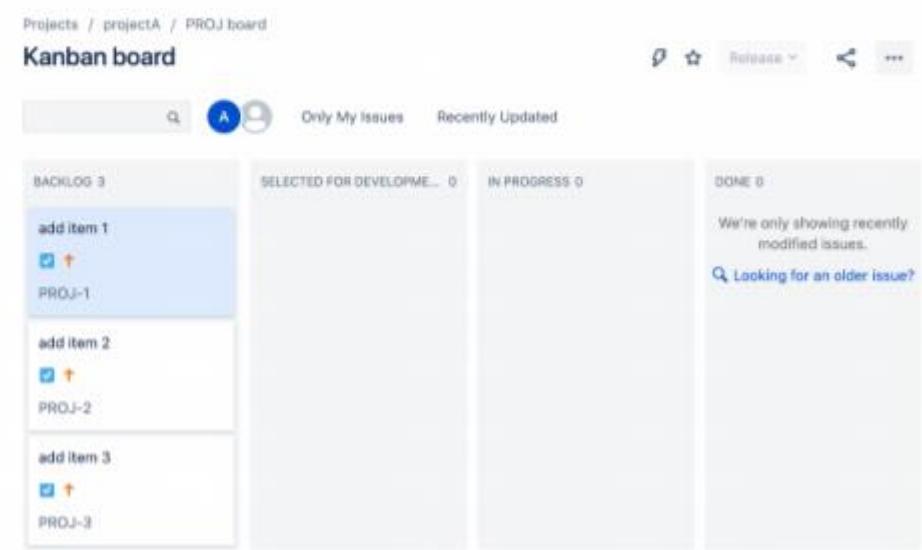
# Boards and status

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

Projects / projectA / PROJ board

### Kanban board

Only My Issues Recently Updated



BACKLOG 3	SELECTED FOR DEVELOPMENT 0	IN PROGRESS 0	DONE 0
add item 1 PROJ-1			We're only showing recently modified issues. <a href="#">Looking for an older issue?</a>
add item 2 PROJ-2			
add item 3 PROJ-3			

# Topics

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



# Adding a board column

Projects / projectA / PROJ board

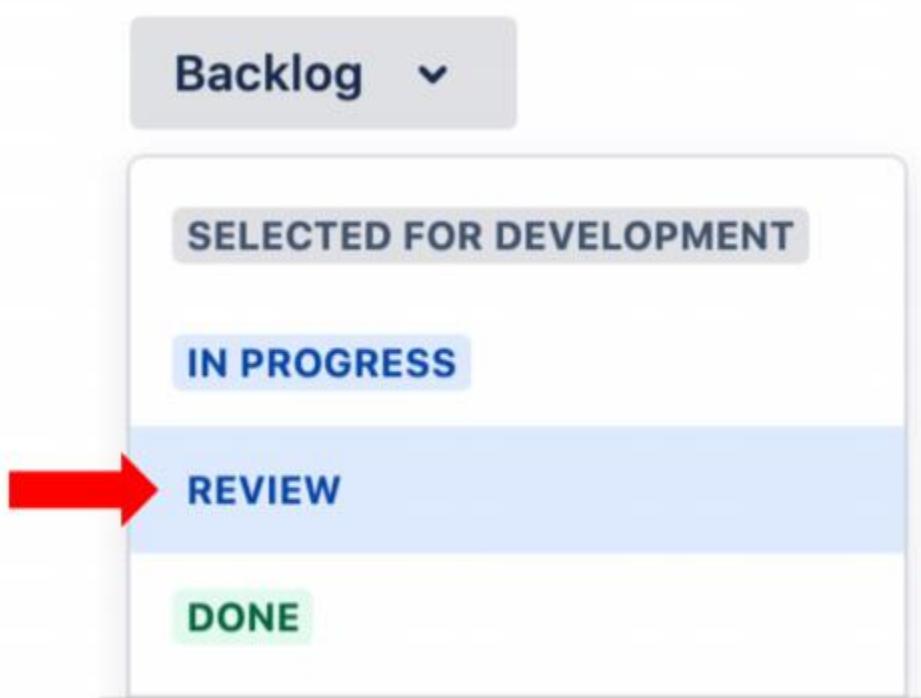
## Kanban board

Release   

  Only My Issues Recently Updated

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

# Viewing the new status



# Takeaways

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

# Lab 3 - Visualize Work Using Boards

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

# Enrich Issues



# What will you learn?

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

# Topics

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



# Issues contain work-related information

## Issue

Summary: Check network jacks

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

Type:  Task

Assignee:



Helena

Priority:  Critical

Status:  IN PROGRESS

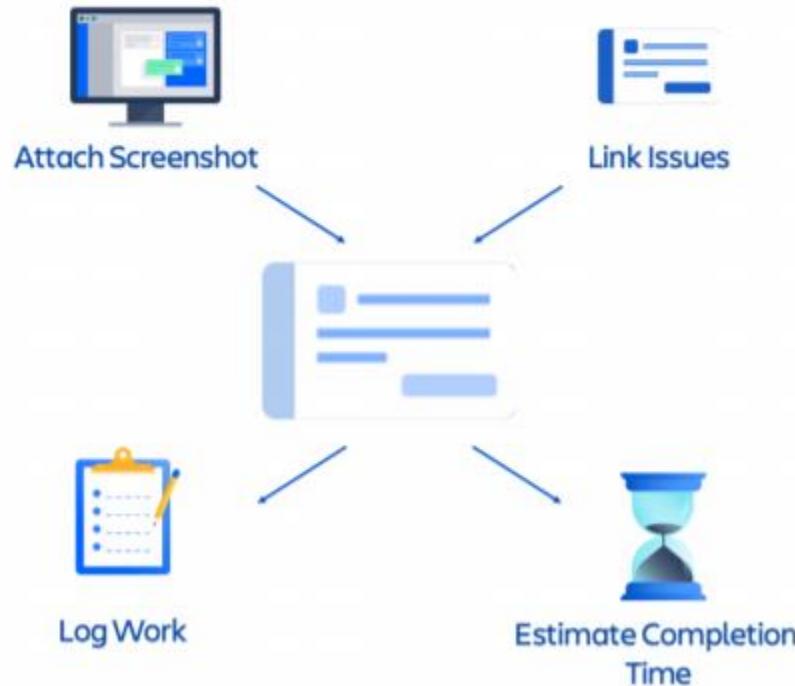
Reporter:



Oliver

Comments: Helena needs the network diagram from IT.

# Enriching issues



# Mention team members



# Topics

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



# The issue type field

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

Create issue

Project\*

projectA (PROA)

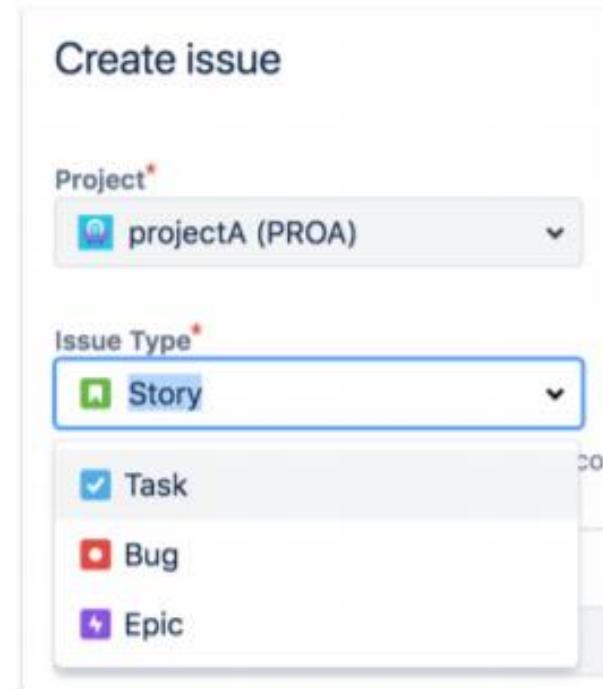
Issue Type\*

Story

Task

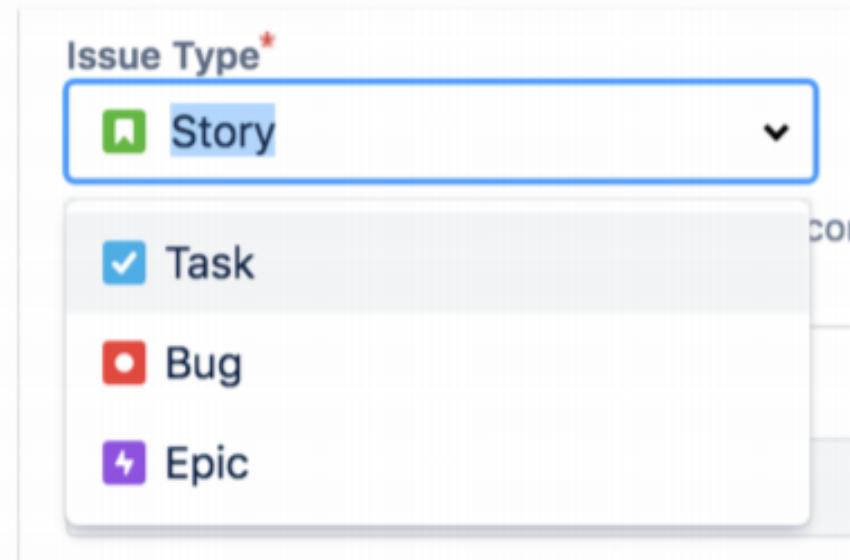
Bug

Epic

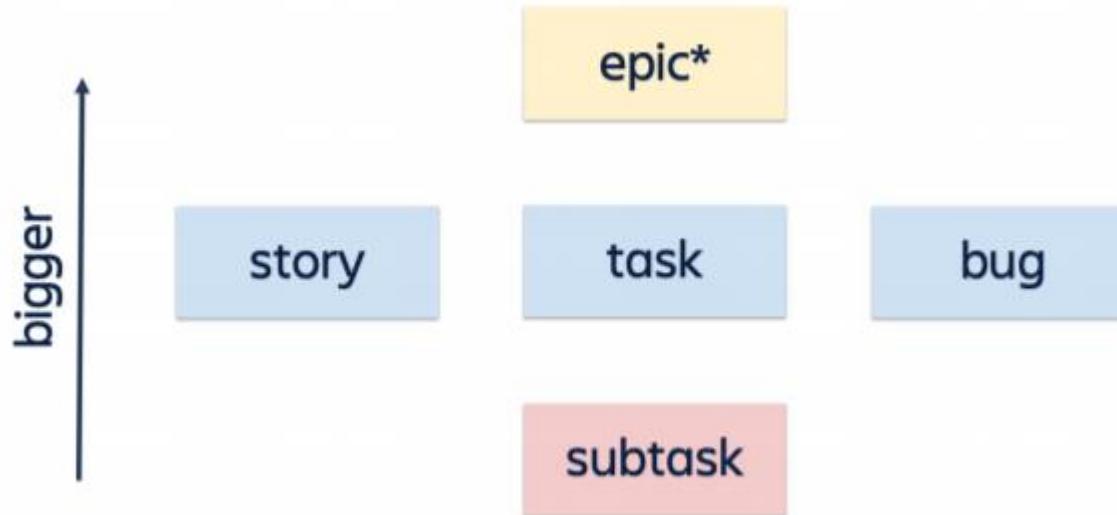


# Why issue types

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



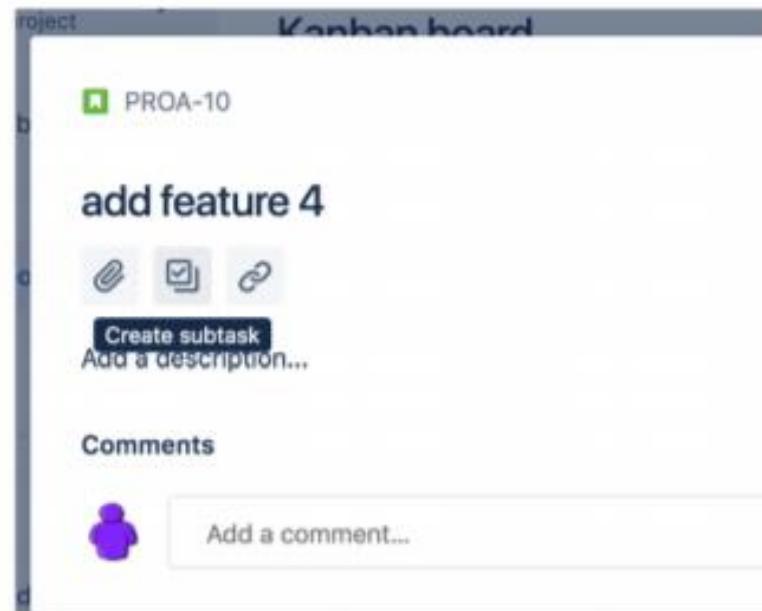
# Jira's issue type hierarchy



\* Epics are discussed later

# Subtasks

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



# Subtask characteristics

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

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

# Topics

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



# Labels

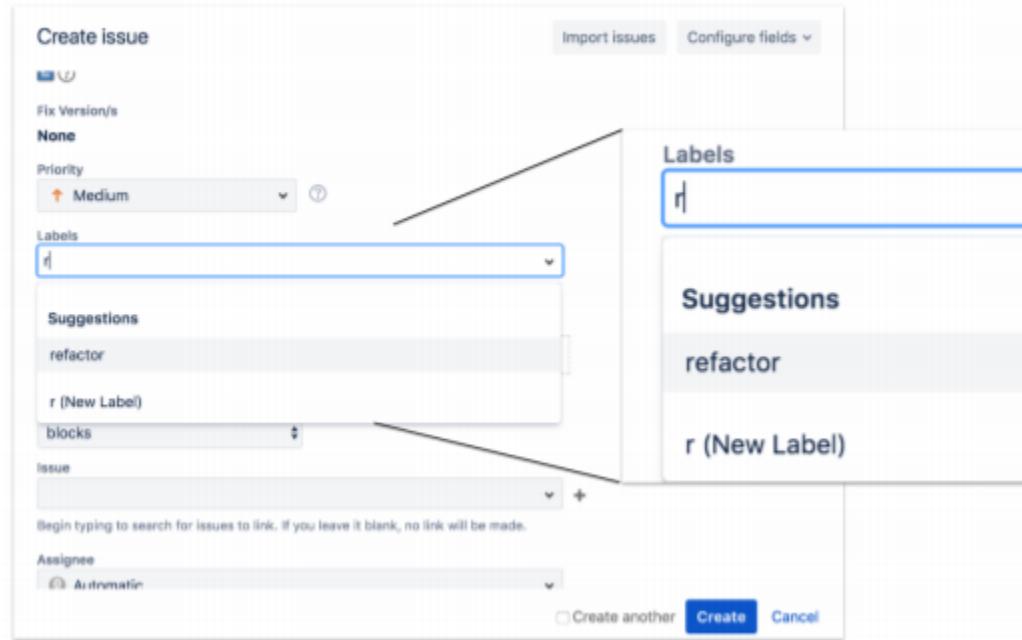
A field used to categorize and search for issues

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

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

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

# Adding or creating a labels



# Searching for a Labeled issues

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

The image shows a Jira interface. On the left, a new issue is being created for 'PROJ-6' with the title 'complete task 1'. The 'Labels' field contains 'database refactor'. On the right, a 'Search' results page displays two issues: 'PROJ-10 PROJ-6 / complete task 1a' and 'PROJ-6 complete task 1', both of which have the 'refactor' label applied.

Labels

database refactor

Search

labels = "refactor"

1-2 of 2

Key	Summary
PROJ-10 PROJ-6 / complete task 1a	
PROJ-6 complete task 1	

1-2 of 2

# Topics

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



# The issue detail development panel

Issue detail view  
for selected issue



## Development

4 commits

Latest 2 days ago

1 pull request

MERGED

Updated 2 days ago

1 build



Latest 2 days ago

Deployed to QA

Create a branch or view existing  
branches related to this issue



Create branch

# Integration works through the issue key

## Using a Commit Message

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

## Using a Branch Name

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

## For Pull Requests

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

## For Builds and Deployments

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

# Takeaways

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

# Lab 4- Enrich Issues

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

# Kanban Method



# What will you learn?

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

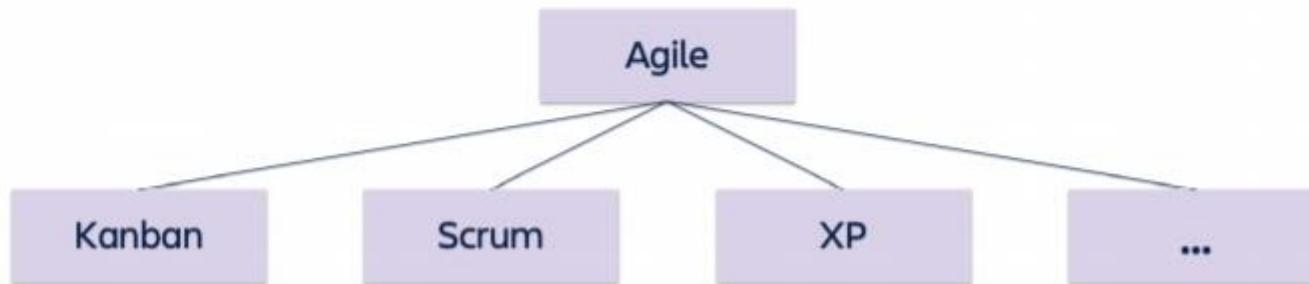
# Topics

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



# Agile methods

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



# What is kanban method?

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

# Why choose the kanban method

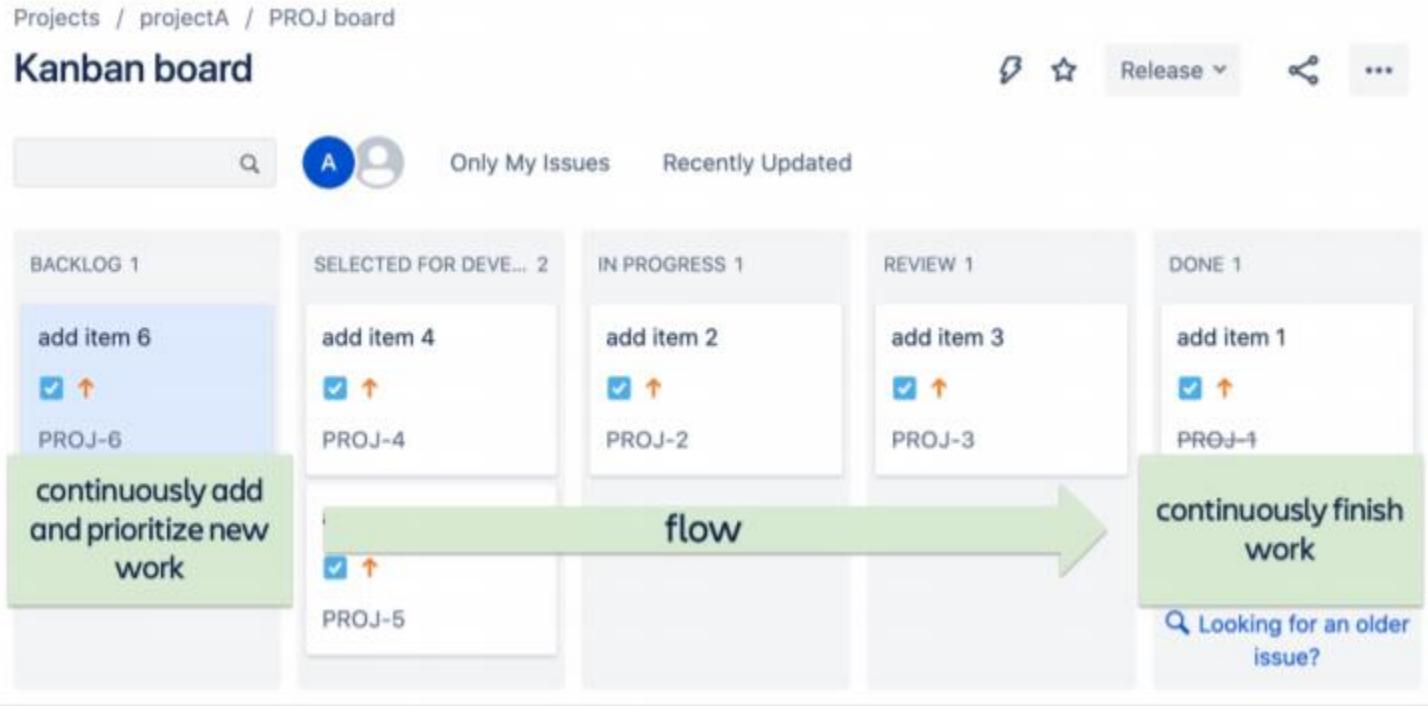
- Very Lightweight and efficient
- Evolutionary approach of transforming to agile
- Works well If the workflow is service-oriented
- operations
- support
- maintenance development
- new hire funnel
- Supports multi-team and multi-project workflows

# Topics

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



# Continuous flow of work items



# Improving flow- limit work in progress (WIP)

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

# Column under minimum limit

Projects / projectA / PROJ board

## Kanban board

Only My Issues Recently Updated

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

Min 2

IN PROGRESS 1

REVIEW 1

DONE 1

We're only showing recently modified issues.

Looking for an older issue?

# Column over maximum limit

Projects / projectA / PROJ board

## Kanban board

The screenshot shows a Jira Kanban board with the following columns and their contents:

- BACKLOG 1**:
  - add item 6  
PROJ-6
  - add item 5  
PROJ-5
- SELECTED FOR DEVEL... 2**:
  - add item 4  
PROJ-4
- IN PROGRESS 0**: (empty)
- REVIEW 3 Max 2**:
  - add item 1  
PROJ-1
  - add item 2  
PROJ-2
  - add item 3  
PROJ-3A red arrow points down to the "Max 2" label above this column.
- DONE 0**:
  - We're only showing recently modified issues.  
Looking for an older issue?

# What should WIP limits be set to?

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

Projects / projectA / PROJ board

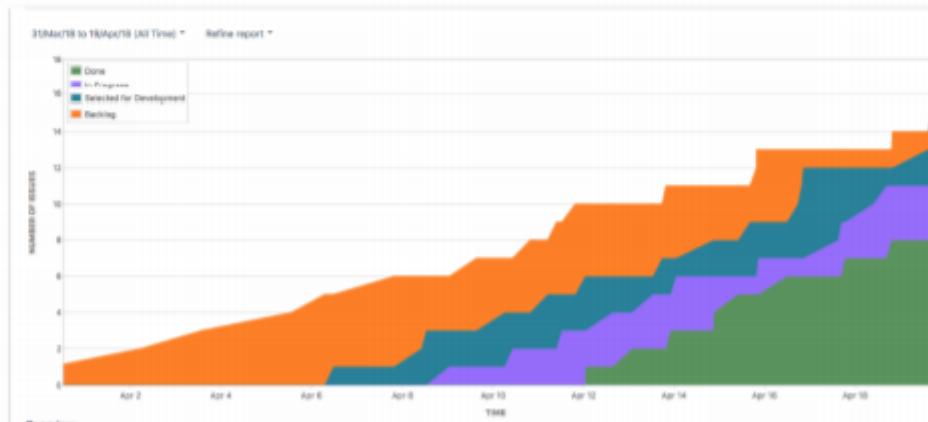
## Kanban board

Only My Issues Recently Updated

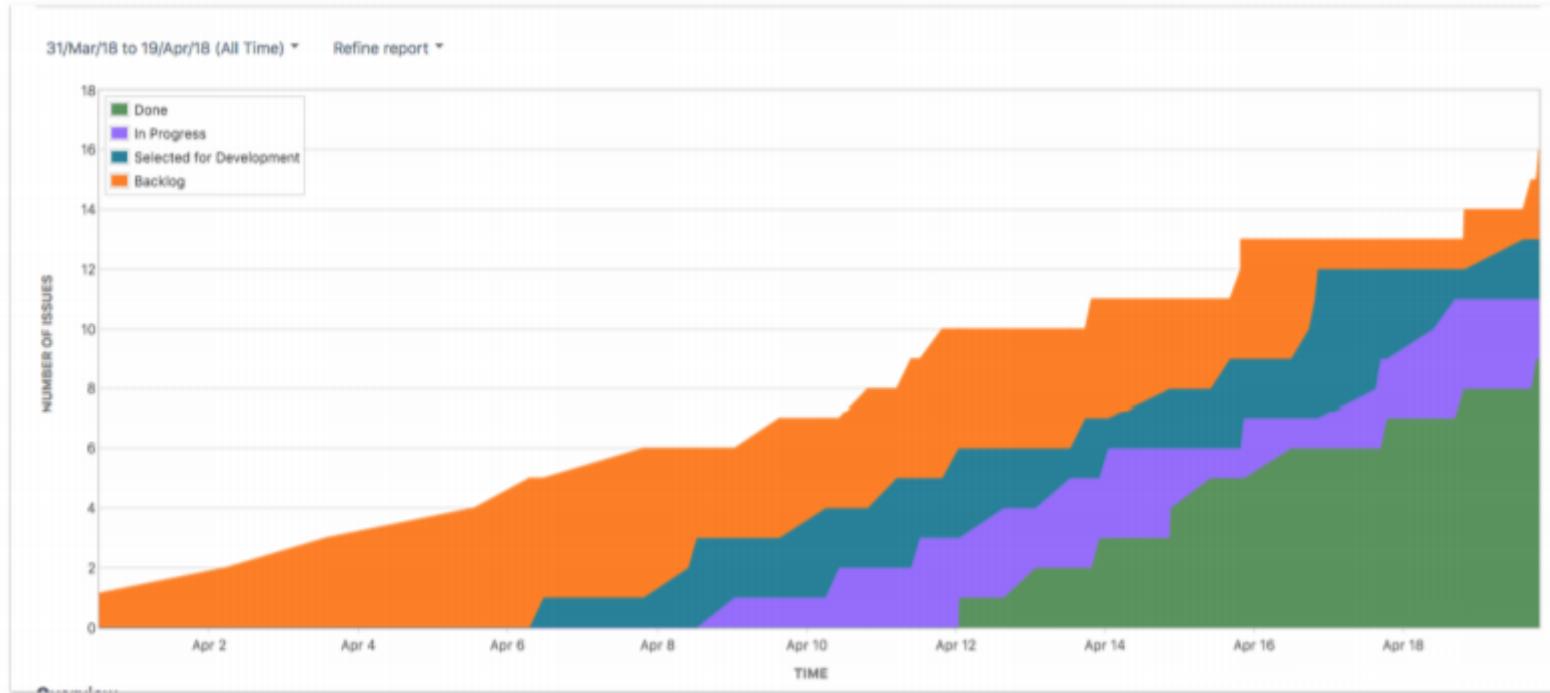
BACKLOG 1	SELECTED FOR DEVELOPMENT 2	IN PROGRESS 1	REVIEW 2	DONE 6
add item 6 PROJ-6	add item 4 PROJ-4	add item 3 PROJ-3	add item 1 PROJ-1	We're only showing recently modified issues. Looking for an older issue?
	add item 5 PROJ-5		add item 2 PROJ-2	

# What agile reports

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



# Cumulative flow



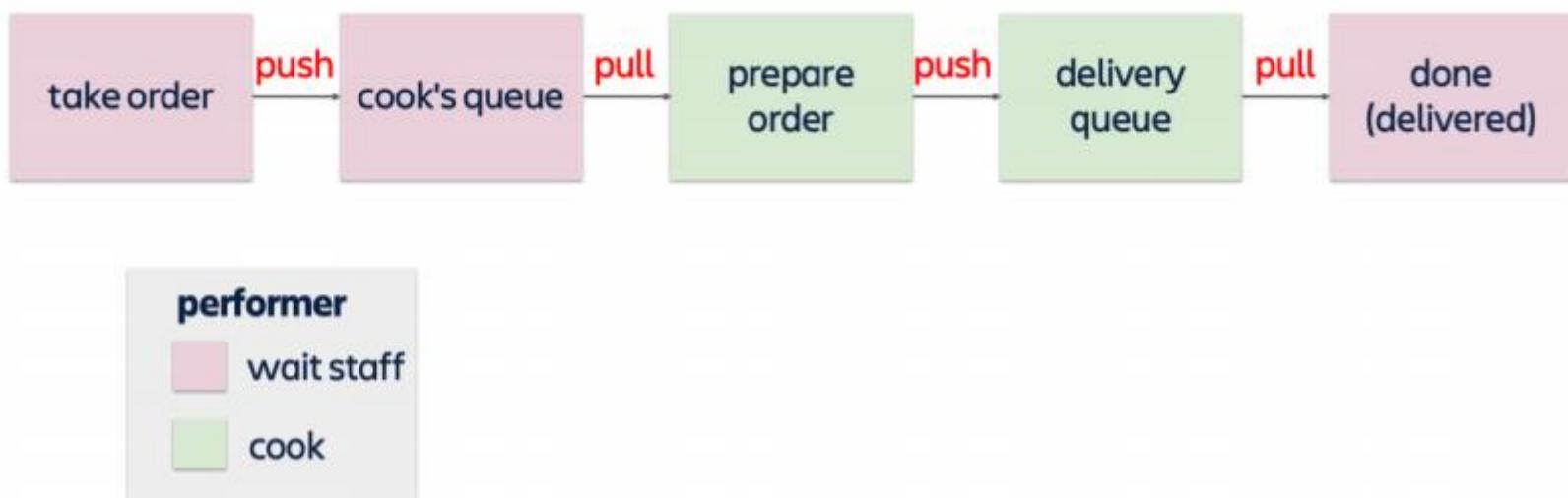
# Topics

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



# Pull vs Push in process steps

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



# Pull vs Push

Projects / projectA / PROJ board

## Kanban board

The screenshot shows a Kanban board with four columns: BACKLOG, SELECTED FOR DEVELOPMENT, IN PROGRESS, and DONE. Red arrows labeled 'push' point from the first three columns to the fourth, indicating a push-based flow where items are moved through stages sequentially. A red arrow labeled 'pull' points from the second column to the third, indicating a pull-based flow where work is requested from the previous stage.

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

push → pull → push

Only showing recently modified issues.  
Looking for an older issue?

# Adding queues to enabled pull (1 of 2)

Projects / projectA / PROJ board

## Kanban board

Release    

Only My Issues Recently Updated

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

# Adding queues to enabled pull (2 of 2)

Projects / projectA / PROJ board

## Kanban board

The Kanban board displays the following information:

- BACKLOG 1:** add item 6 (PROJ-6), add item 5 (PROJ-5)
- SELECTED FOR DEVELOPMENT 2 Min 2:** add item 3 (PROJ-3), add item 4 (PROJ-4)
- IN PROGRESS 1:** add item 4 (PROJ-4)
- DEVELOPMENT DONE 0:** (empty)
- REVIEW 1:** add item 2 (PROJ-2)
- DONE 1:** add item 1 (PROJ-1)

A message in the REVIEW column states: "We're only showing recently modified issues."

At the bottom of the board, red arrows indicate a push-pull flow between columns:

- BACKLOG → SELECTED FOR DEVELOPMENT (push)
- SELECTED FOR DEVELOPMENT → IN PROGRESS (pull)
- IN PROGRESS → DEVELOPMENT DONE (push)
- DEVELOPMENT DONE → REVIEW (pull)
- REVIEW → DONE (push)

A search bar at the bottom right says "Looking for an older issue?"

# Why Pull ?

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

Projects / projectA / PROJ board

## Kanban board

Only My issues Recently Updated

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

**pull**

We're only showing recently modified issues.

Looking for an older issue?

# Topics

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



# Separate Backflow

Projects / projectA / PROJ board

## Kanban board

Only My Issues Recently Updated

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

can move these issues off of the board

We're only showing recently modified issues.

Looking for an older issue?

# Why a Separate Backflow?

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

Projects / projectA / PROJ board

## Kanban board

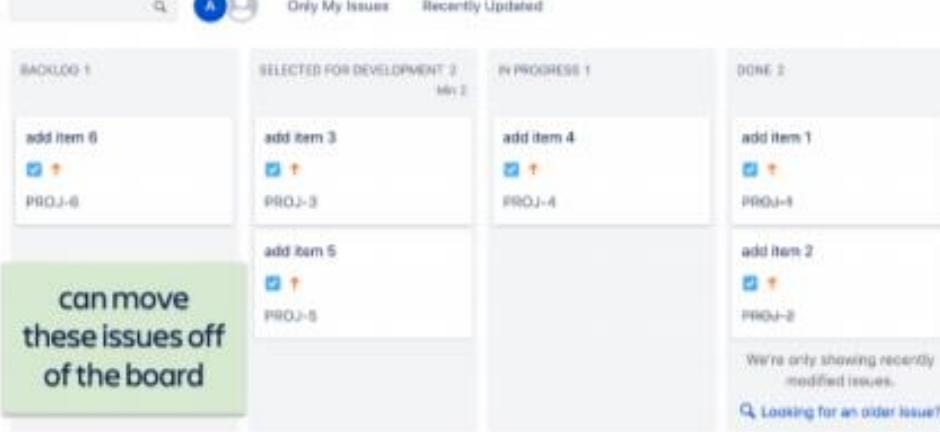
Only My Issues Recently Updated

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

can move these issues off the board

We're only showing recently modified issues.

Looking for an older issue?



# Managing a Separate Backflow

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

**Selected for Development** (MIN 2 2 issues)

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

**Backlog** 1 issue

- add item 6 PROJ-6 ↑

+ Create issue

# Takeaways

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

# Lab 5- Kanban Method

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

# Lean and Agile Principles



# What will you learn?

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

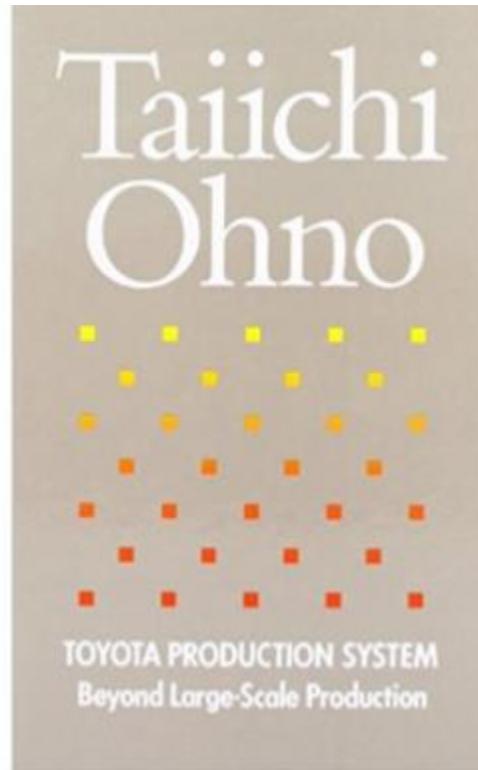
# Topics

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



# Toyota production system

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



# Toyota simplified history

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

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

Taichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Lean principles apply broadly

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

Taiichi Ohno

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

# Topics

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



# What is kanban?

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

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Other examples of kanbans



guest check



coffee cup



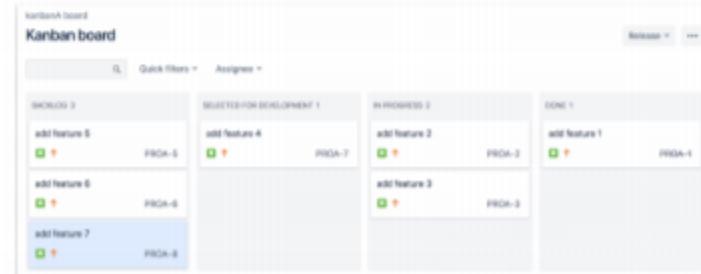
Jira issue

# Kanban systems

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production



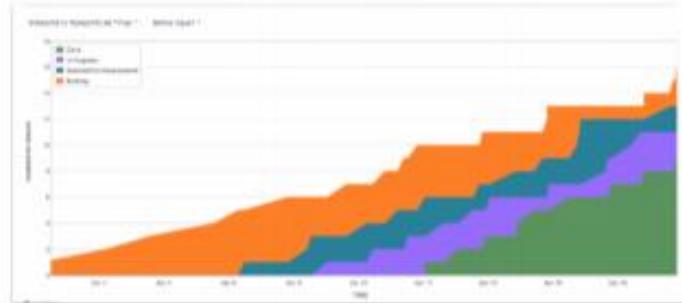
# Benefits of kanban system

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



# Summary – kanban definitions

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



# Topics

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



# Why principles?

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

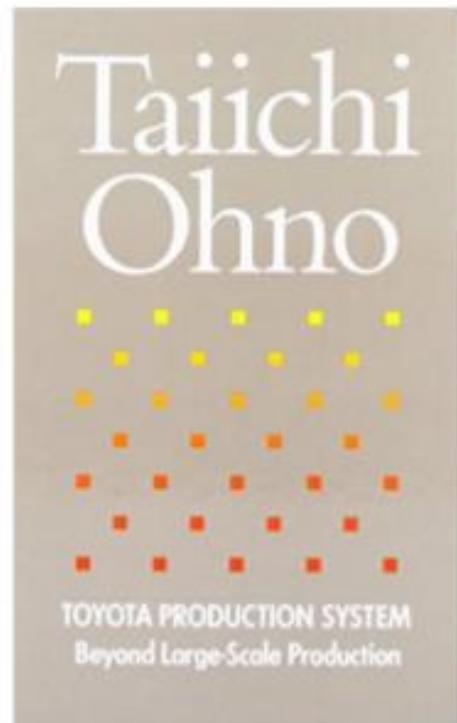
"We should not forget to always use the principles..."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Lean principles

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



# Empower the team

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Empower the team – teamwork

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

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Visualize work

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

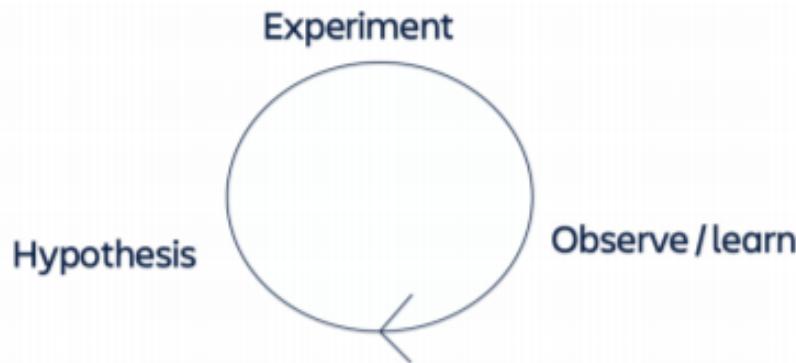
"When one looks up, the andon (the line stop indication board) comes into view, showing the location and nature of trouble situations at a glance."

Taichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Embrace the scientific method

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



# Embrace the scientific method

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

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

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

Taiichi Ohno  
Toyota Production System: Beyond Large-Scale Production

# Embrace the scientific method-embrace change

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

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

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

"Reducing the number of kanban increases their sensitivity."

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Improve the flow - map the value stream

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

# Improve the flow – pull work

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Improve the flow – eliminate waste

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

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Improve the flow – reduce setup times

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Improve the flow – automate what should be automated

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Summary – ways to improve the flow of value

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

"Look straight at the reality."

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Build quality in

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

Taiichi Ohno

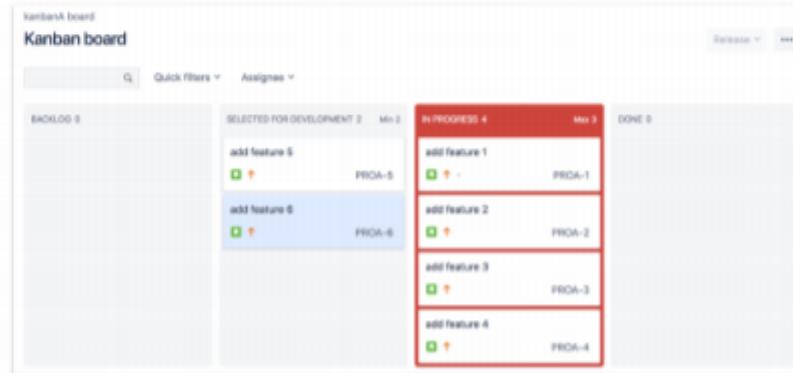
Toyota Production System: Beyond Large-Scale Production

# Build quality in-the process should identify problems

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production



# Build quality in-fix problems when they are discovered

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

# Example-fix problems when they are discovered



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

Taiichi Ohno  
Toyota Production System: Beyond Large-Scale Production

# Build quality in- identify and fix the root cause

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

Taiichi Ohno

Toyota Production System: Beyond Large-Scale Production

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

**Problem:** The orbiter crashed into the planet.

**Question:** Why did the orbiter crash into the planet?

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

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

**Answer:** Because the thrusters did not work properly.

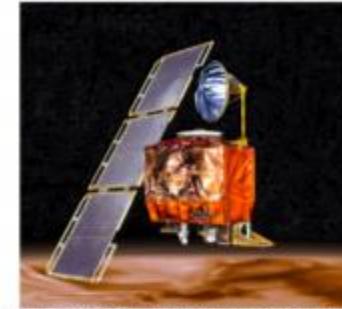
**Question:** Why didn't the thrusters work properly?

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

**Question:** Why was the acceleration data inaccurate?

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

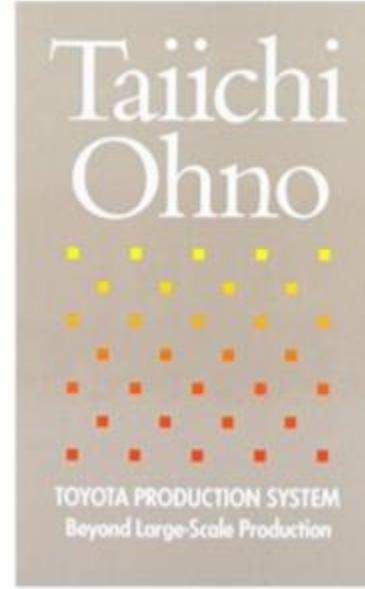
**Solution:** Standardize on a single system of measurement. Ensure pre-launch tests fail in similar circumstances.



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

# Lean principles

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



# Topics

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



# Manifesto for Agile software development

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

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

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

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

© 2001, the above authors

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

<https://agilemanifesto.org>

# Agile manifesto values statements-takeaway principles

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

# Principle behind the Agile Manefesto

## Principles behind the Agile Manifesto

*We follow these principles:*

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

# Principle and ideas from the Agile Manefesto

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

# Topics

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



# Lean vs Agile

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

# Lean and Agile principles

## Lean

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

## Agile

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

# Combined Lean and Agile principles

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

# Scrum Overview 1- Artifacts



# What will you learn?

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

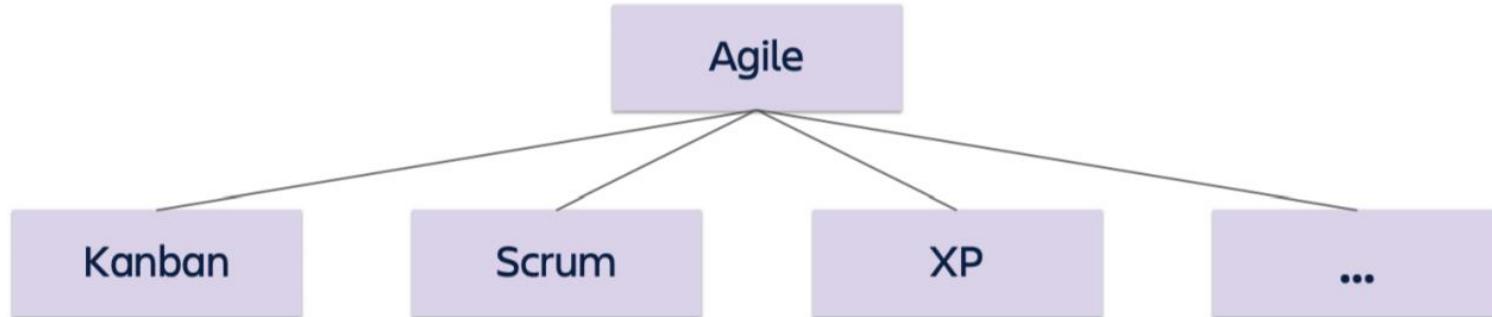
# Topics

- What is scrum?
- Scrum Artifacts

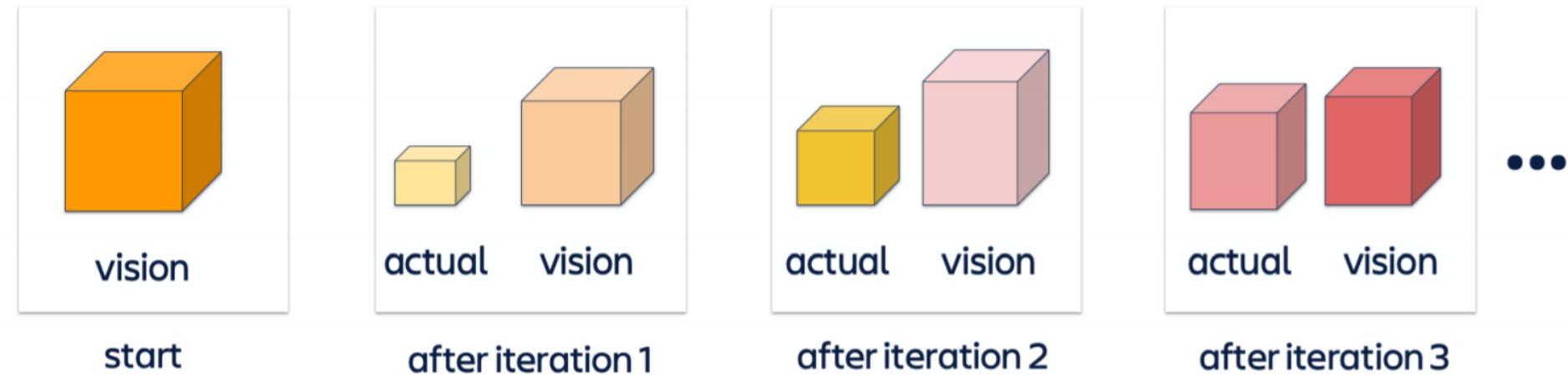


# What is scrum?

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

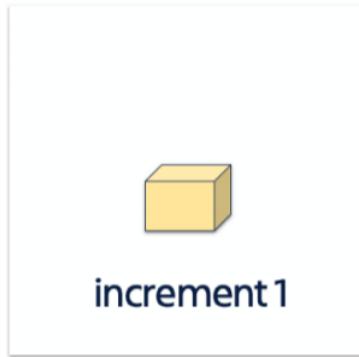


# Continuous learning



# Increment

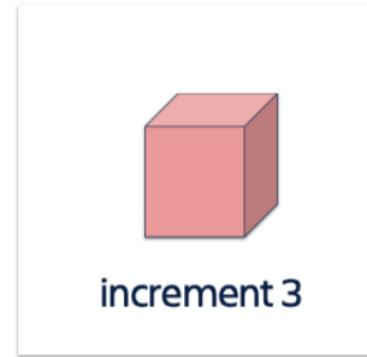
- A usable product that may be given to the customer
- Meets the organization's "definition of done"
- Contains the work of the current iteration, as well as all prior iterations



after iteration 1



after iteration 2



after iteration 3

...

# Sprint

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

- Usually 1-4 weeks (typically 2 weeks)



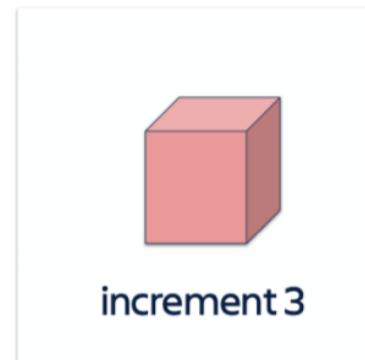
increment 1

after sprint 1



increment 2

after sprint 2



increment 3

after sprint 3

• • •

# Parts of the scrum framework

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

# Topics

- What is scrum?
- Scrum Artifacts



# Scrum Artifacts

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

Projects / projectB / PRJ board

## Backlog

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

Backlog	3 issues
add item 1	
add item 2	
add item 3	

+ Create issue

# Product Backlog

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

Projects / projectB / PRJ board

## Backlog

Share ...

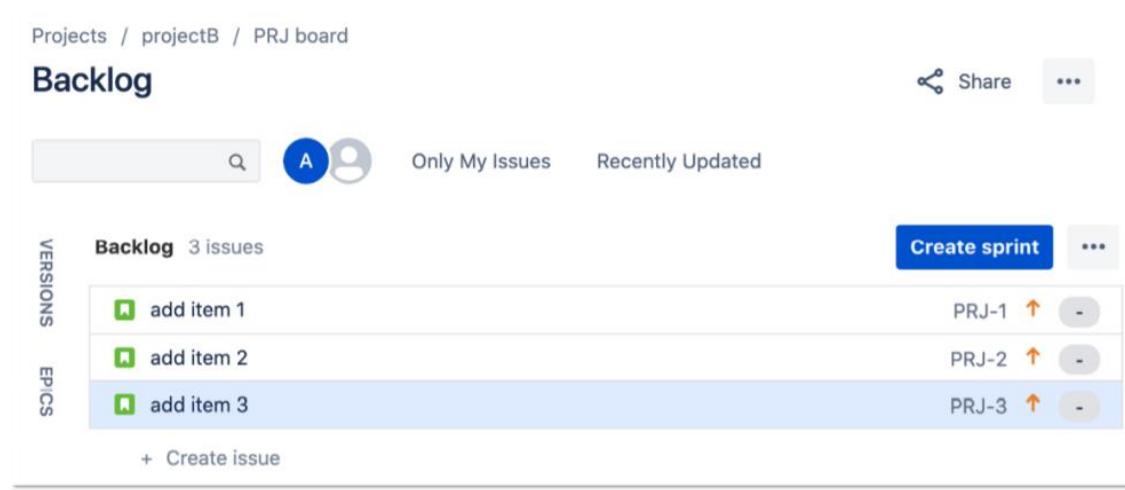
Only My Issues Recently Updated

SEARCH A 8

VERSIONS Backlog 3 issues Create sprint ...

ITEM	VERSION	STATUS	OPERATIONS
add item 1	PRJ-1	↑ -	
add item 2	PRJ-2	↑ -	
add item 3	PRJ-3	↑ -	

+ Create issue



# Creating a sprint with Jira

Projects / projectB / PRJ board

## Backlog

Share ...

Only My Issues Recently Updated

A

PRJ Sprint 1 0 issues

Start sprint Linked pages 0 ...

VERSIONS

EPICS

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

+ Create issue

0 issues Estimate 0

Create sprint ...

Backlog 3 issues

+ Create issue

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

add item 1  
add item 2  
add item 3

# 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

A

PRJ Sprint 1 2 issues

Start sprint Linked pages 0 ...

VERSIONS EPICS

sprint backlog

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

+ Create issue

2 issues Estimate 3

Backlog 1 issue

Create sprint ...

add item 3	PRJ-3 ↑ 3
------------	-----------

+ Create issue

# Estimation-story point

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

Projects / projectB / PRJ board

## Backlog

Share ...

Only My Issues Recently Updated

Versions

PRJ Sprint 1 2 issues

Start sprint Linked pages 0 ...

EPICS

add item 1 PRJ-1 ↑ 1

add item 2 PRJ-2 ↑ 2

+ Create issue

2 issues Estimate 3

Backlog 1 issue

Create sprint ...

+ Create issue

PRJ-3 ↑ 3

Description Add a description...

Story Points

1

The screenshot shows a Jira backlog board for a project named 'projectB'. At the top, there are buttons for 'Start sprint' and 'Linked pages'. Below this, the 'PRJ Sprint 1' section contains two items: 'add item 1' (PRJ-1, 1 point) and 'add item 2' (PRJ-2, 2 points). A 'Create issue' button is also present. The 'Backlog' section contains one item: 'add item 3' (PRJ-3, 3 points). A 'Create sprint' button is located here as well. On the right side, there is a sidebar titled 'Story Points' with a value of '1'. The interface includes standard Jira navigation elements like 'Share', 'Only My Issues', and 'Recently Updated'.

# Sprint Details

Start sprint

2 issues will be included in this sprint.

Sprint name: \*  
PRJ Sprint 1

Duration: \*  
2 weeks

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

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

Sprint goal:

# Sprint Goal

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

Start sprint

2 issues will be included in this sprint.

Sprint name: \* PRJ Sprint 1

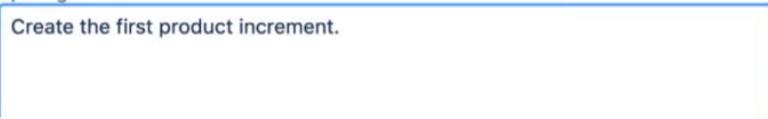
Duration: \* 2 weeks

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

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

Sprint goal:

Create the first product increment.

# Why have a sprint goal

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

Start sprint

2 issues will be included in this sprint.

Sprint name: \* PRJ Sprint 1

Duration: \* 2 weeks

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

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

Sprint goal:

Create the first product increment.

# Sprint board

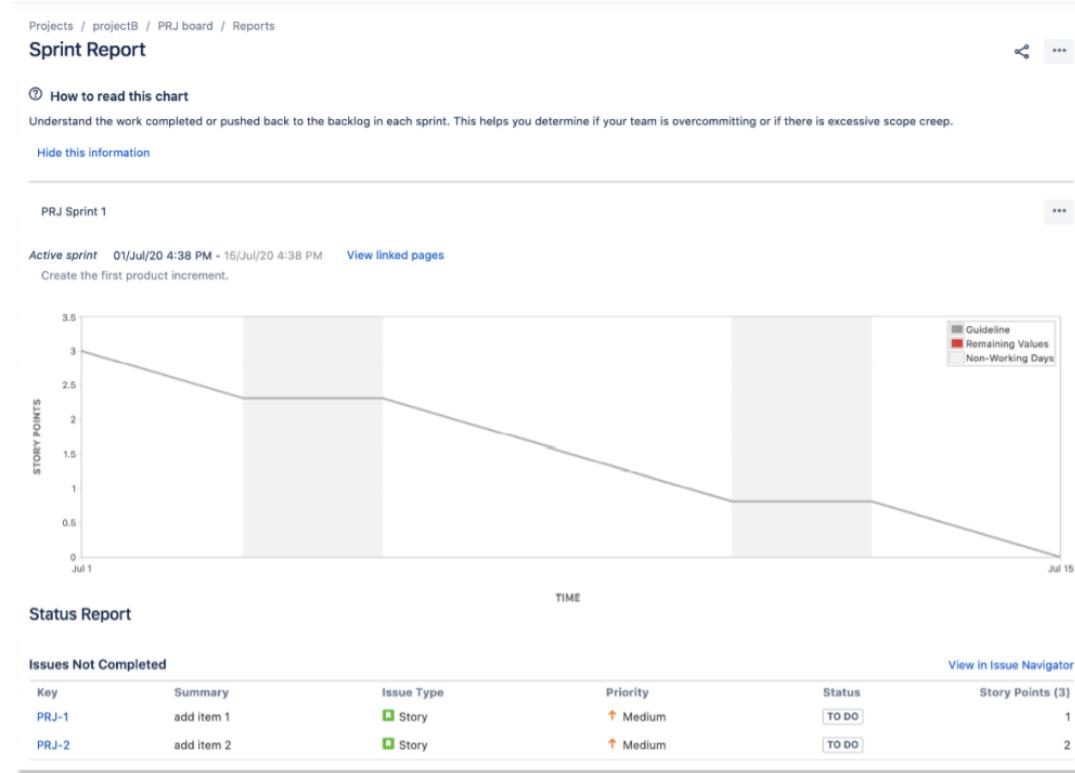
Only contains issues from the sprint backlog

The screenshot shows a Jira Sprint Board interface for a project named "projectB". The board is titled "PRJ Sprint 1" and has a goal: "Create the first product increment." The board is divided into three columns: TO DO, IN PROGRESS, and DONE. The TO DO column contains two items: "add item 1" and "add item 2". Both items have a green "New" icon, an orange "Up" arrow icon, and a circular progress bar indicating completion status. The first item is associated with "PRJ-1" and the second with "PRJ-2". A sidebar on the left provides navigation links for the project, including "PRJ board", "Backlog", "Active sprints" (which is selected), and "Reports". The top right corner shows a user profile, a search bar, and buttons for "Only My Issues" and "Recently Updated". A timer indicates "9 days remaining" and a button to "Complete sprint".

# Scrum Reports – Burndown chart



# Scrum Reports – Sprint Report



# Velocity

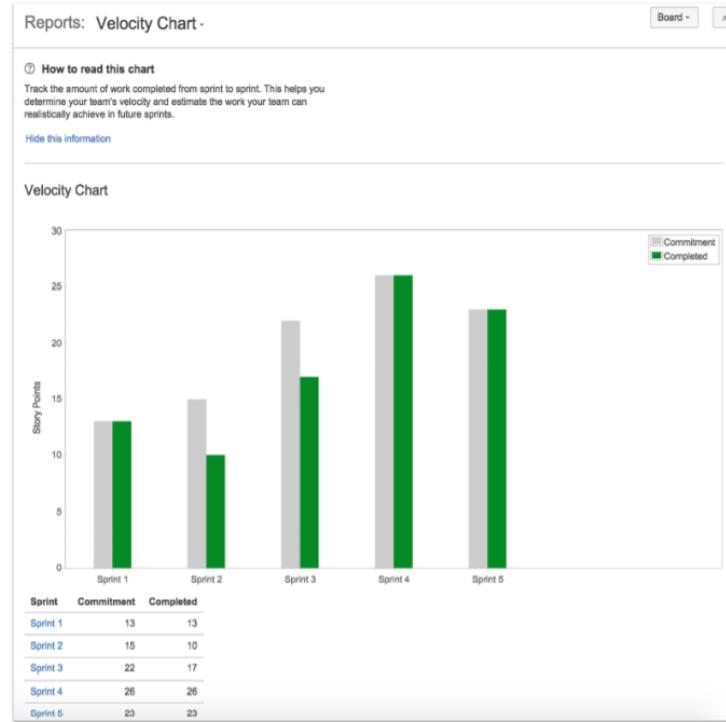
Represents the rate at which the team accomplishes work

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



# Scrum Reports – Velocity Charts

Shows the estimated and actual velocity of the team over time



# Takeaways

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

# Scrum Overview 11- Roles and Events



# What you will learn

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

# Topics

- Scrum roles
- Scrum Events



# Scrum Team

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



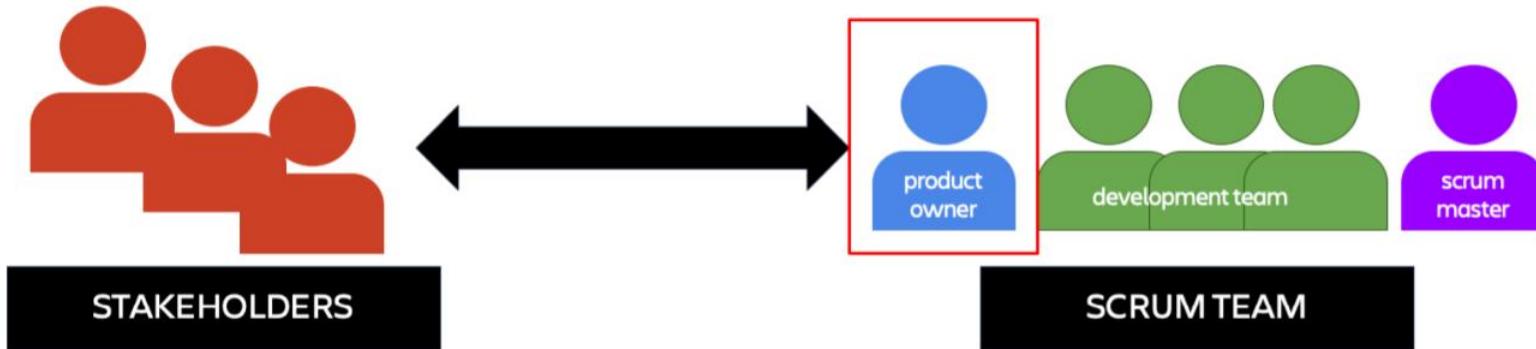
# Stakeholders

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



# Product Owner

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



# Scrum Master

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



# Product Owner VS Scrum Master



## Why separate roles?

- Divide and concur
- Checks and balances

# Development team

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



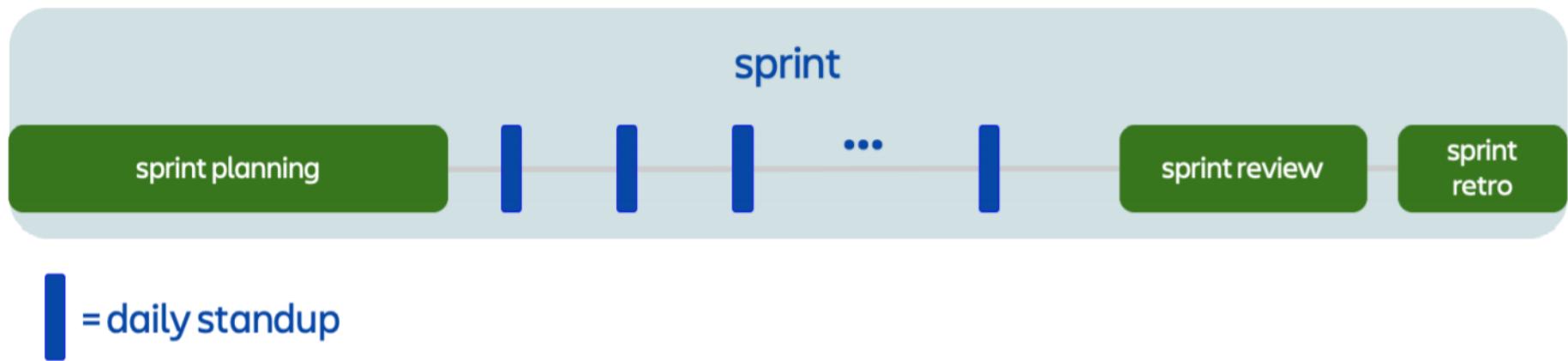
DEVELOPMENT TEAM

# Topics

- Scrum roles
- Scrum Events

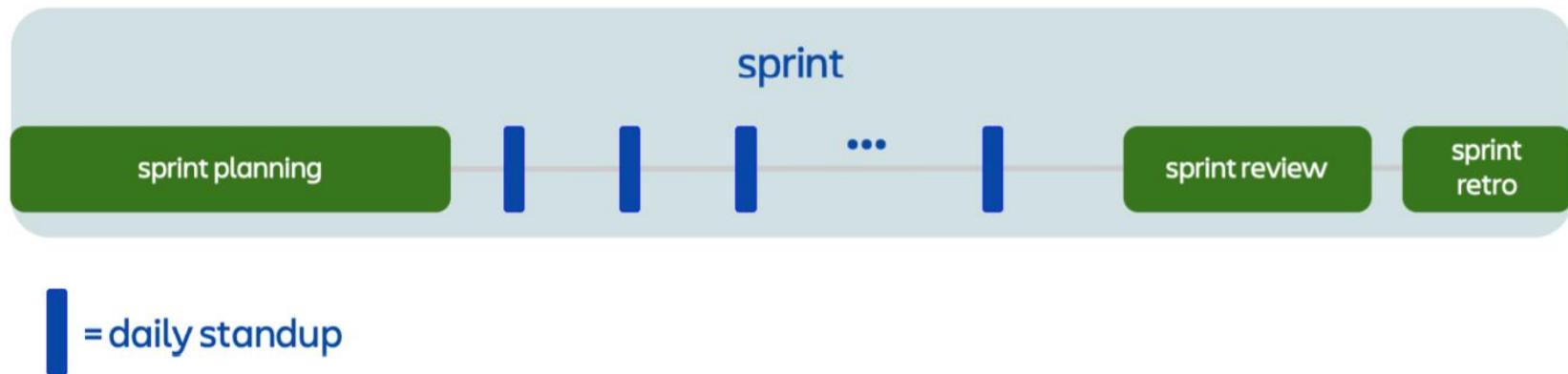


# Scrum Events



# Characteristics of all scrum meetings

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



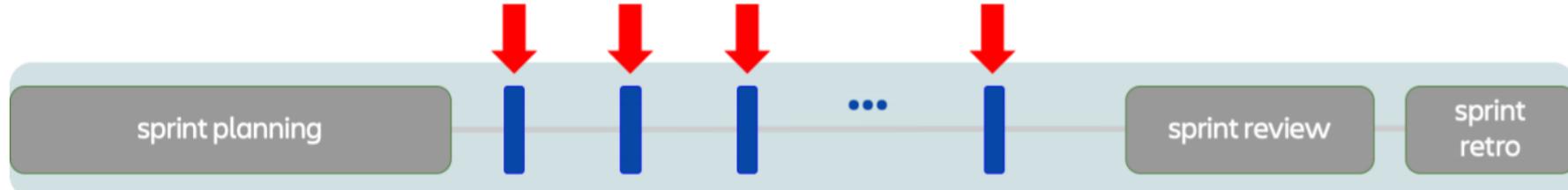
# Sprint planning meeting

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



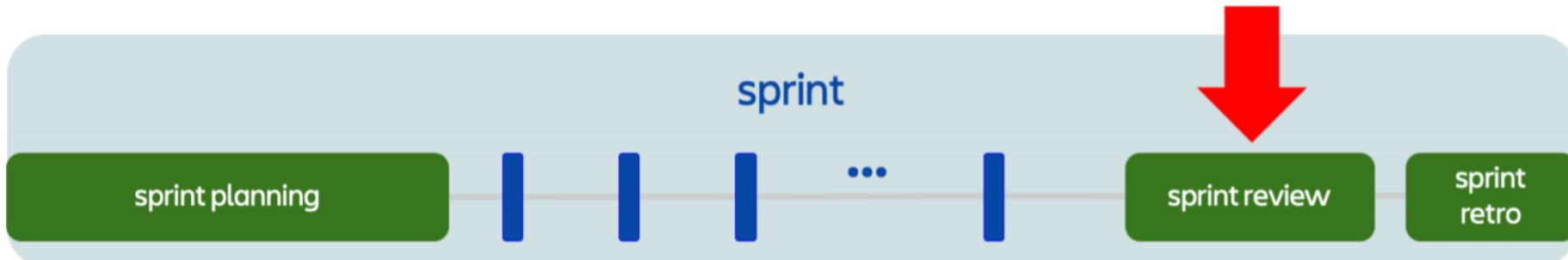
# Daily Standup

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



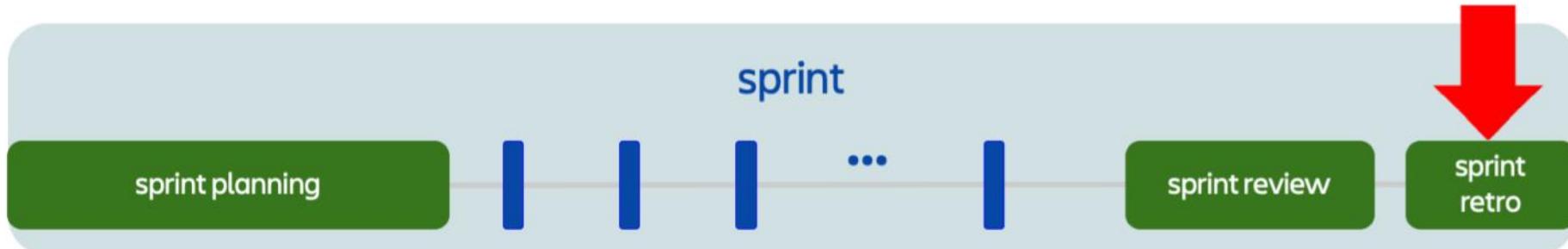
# Sprint Review

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



# Sprint Retrospective

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

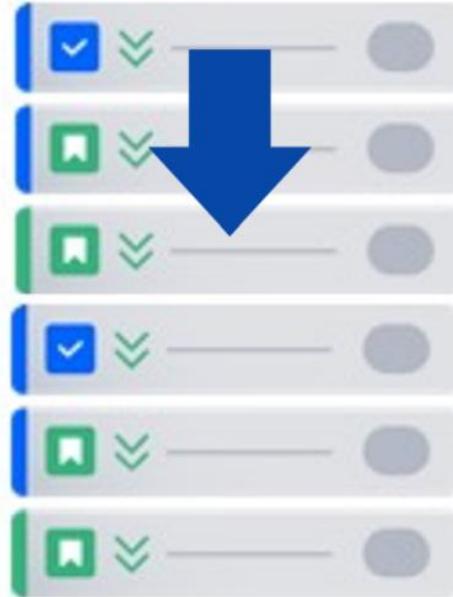


# Scrum meetings - summary

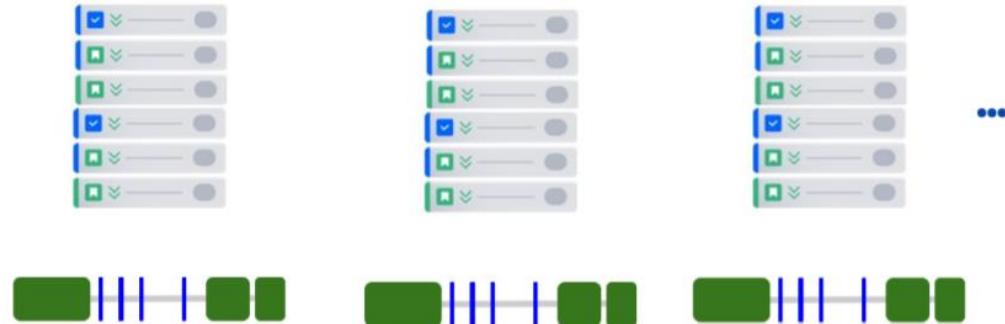
	Sprint Planning	Daily Standup	Sprint Review	Retrospective
<b>Attendees</b>	Scrum team	Development team (primarily)	Scrum team and stakeholders	Scrum team
<b>Duration*</b>	4 hours	15 minutes	2 hours	90 minutes
<b>Purpose</b>	Plan the work of the sprint	Inspect recent work, plan today	Inspect increment,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

\* typical max duration assuming a two week sprint

# Kanban vs Scrum



Kanban



Scrum

# Takeaways

Scrum roles include:

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

Scrum meetings include:

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

# Lab 8 - Scrum Overview II

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

# Quick Search and Basic Search



# What you will learn

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

# Topics

- Searching overview
- Quick search
- Basic search



# Viewing a project's progress



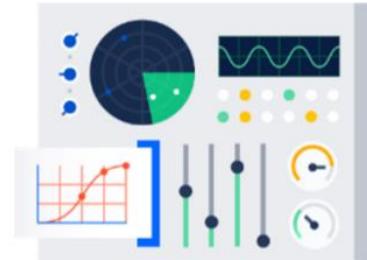
Boards



Search



Reports



Dashboards

# Searching

## 1. Quick search

The screenshot shows the Jira search interface. At the top is a search bar with the placeholder "Search Jira". Below it is a 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 preceded by a small green icon. Below this is a section titled "BOARDS, PROJECTS AND FILTERS" which lists two boards: "PROJ board" and "PRJ board", followed by two projects: "projectA (PROJ)" and "projectB (PRJ)", each preceded by a small colored icon. At the bottom of the sidebar are buttons for navigating to "Issues", "Boards", "Projects", "Filters", and "People".

Search Jira

Advanced search for issues

RECENTLY VIEWED ISSUES

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

BOARDS, PROJECTS AND FILTERS

- PROJ board
- PRJ board
- projectA (PROJ)
- projectB (PRJ)

Go to all: Issues Boards Projects Filters People

# Searching

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

All issues [Save as](#)

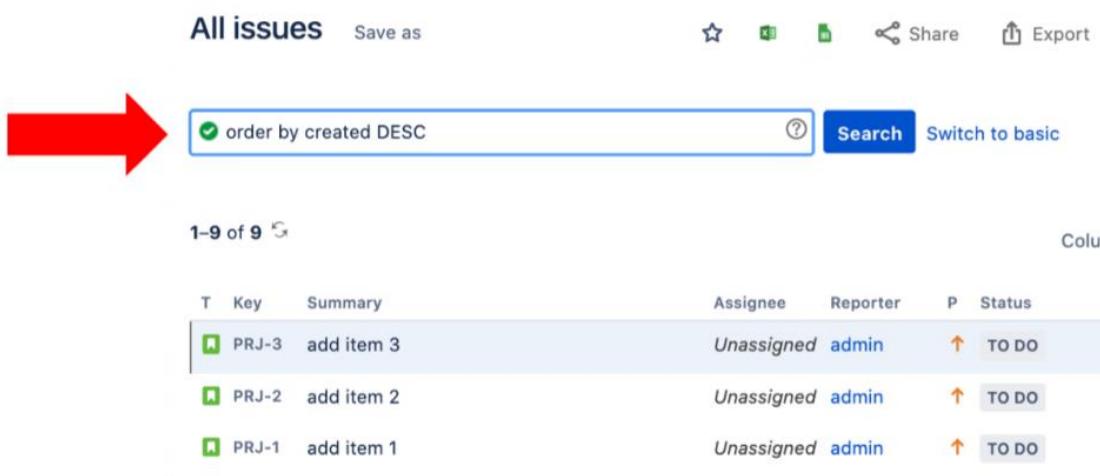
 Project: All ▾ Type: All ▾ Status: All ▾ Assignee: All ▾ + More  [Search](#) [Switch to JQL](#)

1–9 of 9 [S](#)

T	Key	Summary	Assignee	Reporter	P	Status
	PRJ-3	add item 3	Unassigned	admin		<a href="#">TO DO</a>
	PRJ-2	add item 2	Unassigned	admin		<a href="#">TO DO</a>
	PRJ-1	add item 1	Unassigned	admin		<a href="#">TO DO</a>

# Searching

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

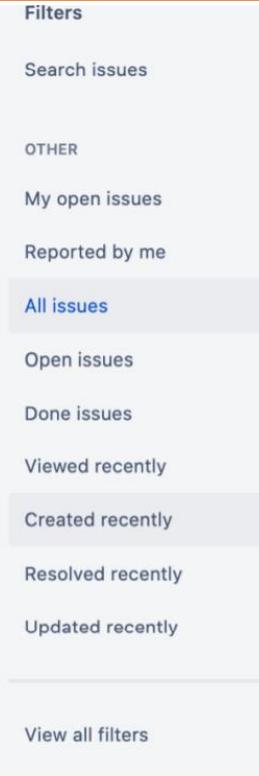


The screenshot shows a Jira search results page titled "All issues". At the top, there are several icons: a star, a green square, a blue square, a share button, and an export button. Below these are three buttons: "Save as", "Share", and "Export". A red arrow points to a search bar containing the query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

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

# Searching

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



# Searching

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

Projects / projectA / PROJ board

Kanban board

The screenshot shows a Jira Kanban board for projectA. At the top, there's a navigation bar with a search icon, a user profile (labeled 'A'), and filter options ('Only My Issues' and 'Recently Updated'). Below the bar are three columns: 'SELECTED FOR DEVELOPMENT 2' (Min 2), 'IN PROGRESS 2', and 'REVIEW 0'. Each column contains two items, each with a title, a checkmark icon, and a progress indicator (blue up arrow, orange down arrow, grey circle). The first item in the 'Selected for Development' column is 'add item 3' (PROJ-3) and the second is 'add item 5' (PROJ-5). The first item in the 'In Progress' column is 'add item 2' (PROJ-2) and the second is 'add item 4' (PROJ-4).

# Why is searching important?

Adapt your Jira experience to your team's processes

Projects / projectA / PROJ board

## Kanban board

The screenshot shows a Jira Kanban board for projectA/PROJ board. At the top, there is a search bar, a user icon (A), and a group icon. To the right of the icons are links for "Only My Issues" and "Recently Updated". Below the header, there are three columns: "SELECTED FOR DEVELOPMENT 2" (Min 2), "IN PROGRESS 2", and "REVIEW 0". Each column contains several items, each with a checkbox, upvote/downvote buttons, and a PROJ-X identifier.

Column	Item Description	Status
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

# Topics

- Searching overview
- Quick search
- Basic search



# Quick search

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

**Jira Cloud (Left):**

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

**Jira Server (Right):**

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

Cloud

Server

# Quick search- with search terms and keywords

feature

ISSUES

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

View all matching issues

feature NOT 1

ISSUES

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

View all matching issues

feature OR sample

ISSUES

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

# Topics

- Searching overview
- Quick search
- Basic search



# Basic search

Search Save as

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

---

Search Save as

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

Find Projects...

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

1-6 of 6

# Contains Text

Search Save as

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

1-5 of 5 

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

1-5 of 5 



# Searching more fields

A screenshot of a search interface, likely from a project management tool. At the top, there are filters for 'projectA', 'Type: All', 'Status: All', and 'Assignee: All'. To the right of these are buttons for '+ More', 'Contains text', 'Search', and 'Switch to JQL'. A large red arrow points down to a modal window titled 'All Criteria'.

The modal contains a search bar and a list of criteria:

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

At the bottom of the modal, it says "...excluding 9 hidden".

The main search results table shows 5 items:

T	Key	Summary	Assignee	Status	Due Date
<input checked="" type="checkbox"/>	PROJ-4	add item 4	Unassigned	In Progress	2023-10-05
<input checked="" type="checkbox"/>	PROJ-2	add item 2	Unassigned	In Progress	2023-10-05
<input checked="" type="checkbox"/>	PROJ-3	add item 3	Unassigned	Approved	2023-10-05
<input checked="" type="checkbox"/>	PROJ-6	add item 6	Unassigned	Pending	2023-10-05
<input checked="" type="checkbox"/>	PROJ-5	add item 5	Unassigned	Approved	2023-10-05

Below the table, it says "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

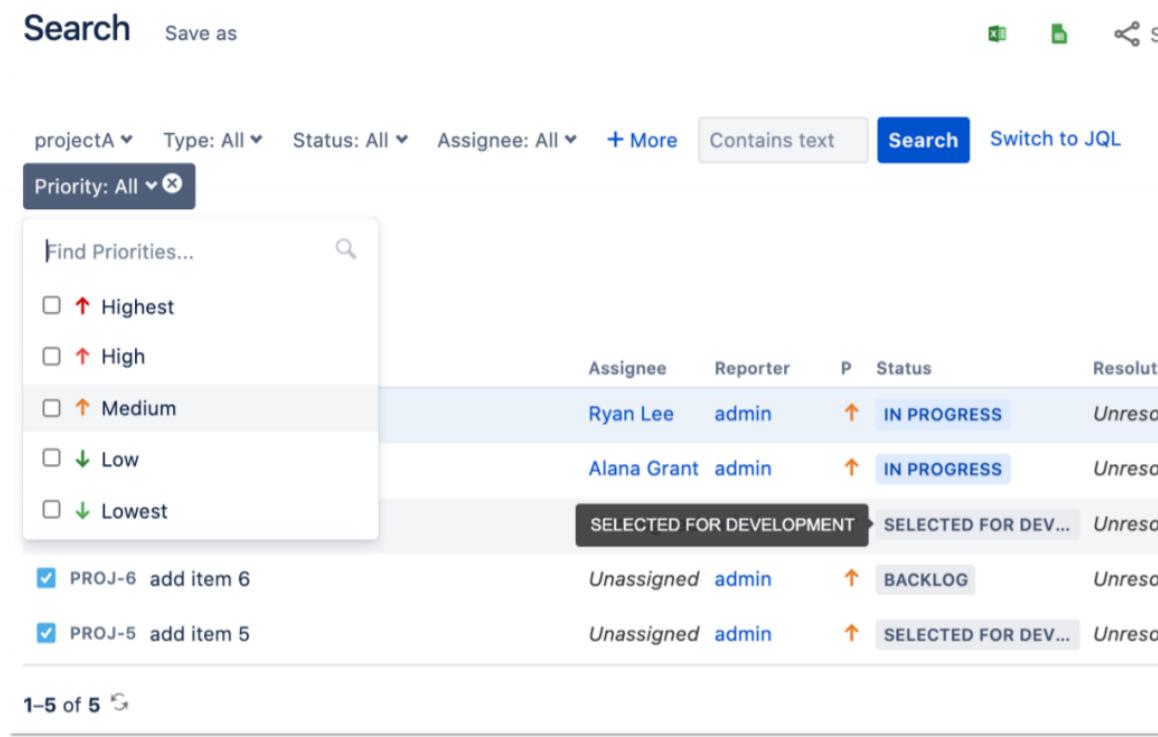
Priority: All

Find Priorities...

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

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

1-5 of 5 



# Example- Specifying an updated date

Search Save as

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

Updated Date: All X

Within the last 3 days ▼

More than minutes ago ▼

Between 11-Jan-2019 and 30-Jan-2019 ▼

In range -3w 4d to 3w 4d ▼

Update Close

PROJ-5 add item 5

1-5 of 5 G

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

# Takeaways

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

# Lab 9 - Quick Search and Basic Search

- Perform quick searches
- Perform basic searches

# JQL



# What will you learn?

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

# Topics

- JQL Overview
- Autocomplete
- Functions



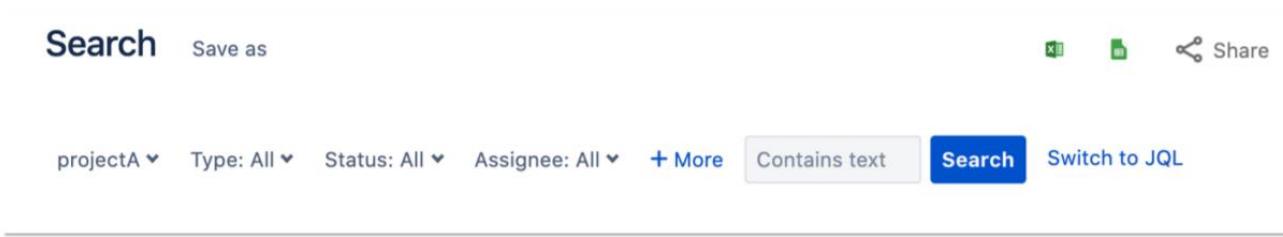
# Basic and advanced/JQL Search

basic

Search Save as

X E Share

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



advanced/JQL

Search Save as

E Share Export

project = PROJ order by lastViewed DESC

?

Search Switch to basic

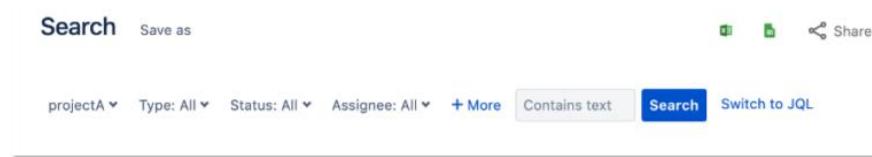


All issues      Save as

order by created DESC

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

# Basic vs. advanced/JQL Search



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

## Basic search

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



## Advanced/JQL search

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

# Writing “JQL” the easiest way

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

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

1-6 of 6

project = PROJ ORDER BY created DESC

1-6 of 6

# The two main parts of JQL query

 project = PROJ order by created DESC

search clause - selects a  
subset of issues

order by clause - orders the  
results

# The Simplest JQL Query



All issues — Edited Save as ▾

1-29 of 29 ⏪ Columns ▾

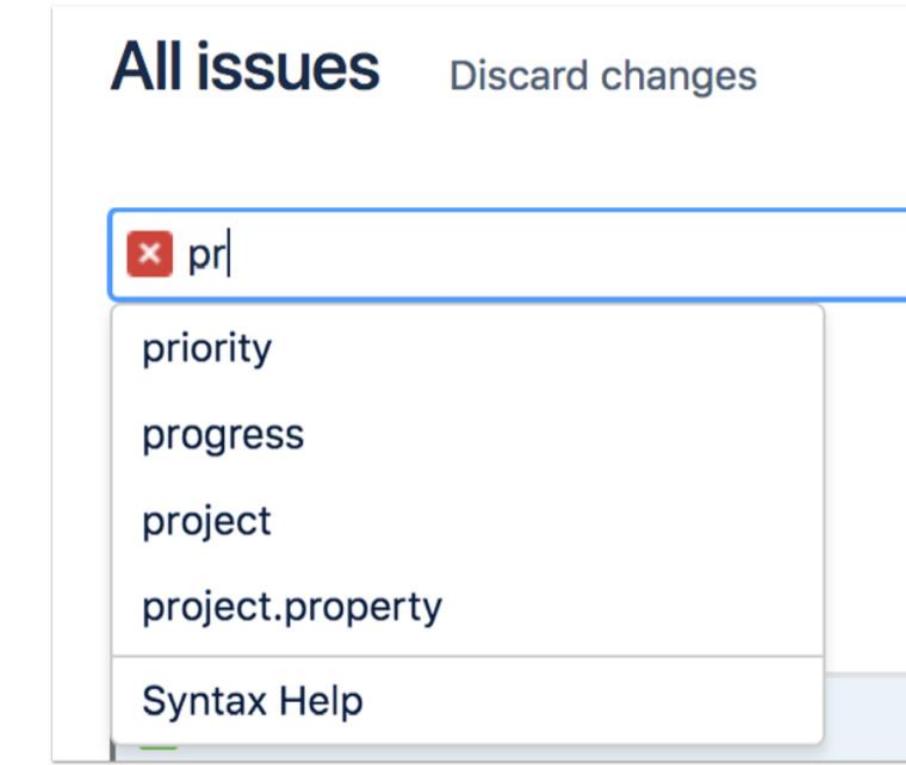
T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	SAM-23	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	09/Jun/18	...
	SAM-22	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done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

# Topics

- JQL Overview
- Autocomplete
- Functions



# JQL with autocomplete - fields



# Advanced searching fields reference

The screenshot shows a blue header bar with the Jira Software Support logo, navigation links for 'Get started', 'Documentation', '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 area has a title 'Advanced searching - fields reference'. A paragraph explains what JQL is and how clauses work. Below this is a section titled 'Affected version' with a table of its properties. To the right is a sidebar titled 'On this page' listing various JQL fields.

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

## Affected version

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

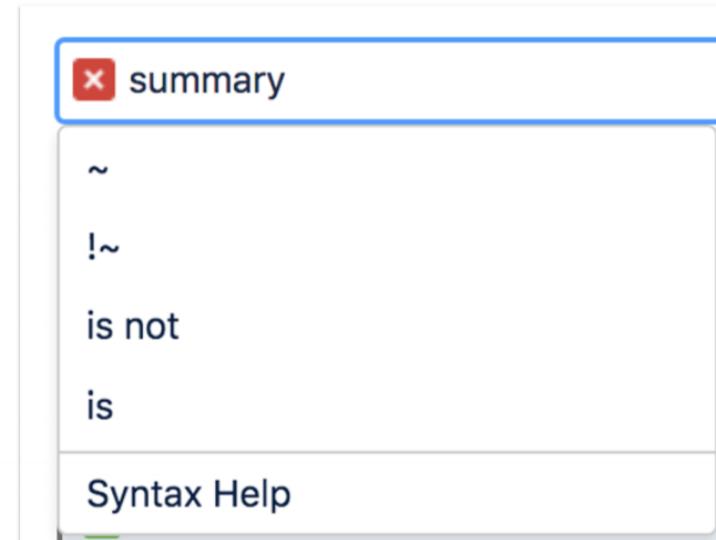
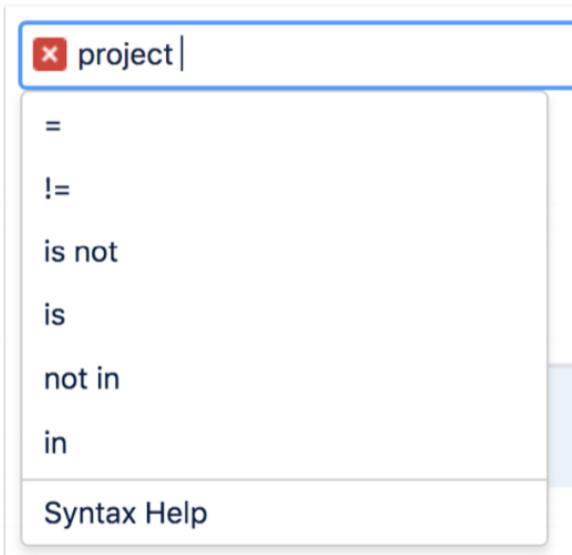
Syntax	affectedVersion
Field Type	VERSION
Auto-complete	Yes
Supported operators	= , != , > , >= , < , <= IS , IS NOT , IN , NOT IN Note that the comparison operators (e.g. ">") use the version order that has been set up by your project

**On this page**

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

# Operator autocomplete

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



# Boolean Operators

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

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

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

NOT status = Backlog
status != Backlog

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

# Advanced searching – operators reference

The screenshot shows a Jira Core Support documentation page. The header includes a navigation bar with 'Jira Core Support' (selected), 'Documentation' (current page), 'Resources', a search icon, and a user profile icon. Below the header, the breadcrumb trail shows 'Atlassian Support / Jira Core / Documentation / ... / ... / Advanced searching'. A 'Cloud' button is also present. The main content title is 'Advanced searching - operators reference'. A sub-section title 'EQUALS: =' is shown, followed by a detailed description of the operator and examples. To the right, a sidebar titled 'On this page' lists various operators: EQUALS: =, NOT EQUALS: !=, GREATER THAN: >, GREATER THAN EQUALS: >=, LESS THAN: <, LESS THAN EQUALS: <=, IN, NOT IN, CONTAINS: ~, DOES NOT CONTAIN: !=~, IS, IS NOT, and WAS.

## Advanced searching - operators reference

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

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

### EQUALS: =

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

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

Examples

- Find all issues that were created by jsmith:

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

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

### On this page

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

# Executing the query

Search [Save as](#)



project = projectA



1-6 of 6

Color

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

Search [Save as](#)

The associated basic search

projectA

Type: All

Status: All

Assignee: All

+ More

Contains text

Search

Switch to JQL

# Topics

- JQL Overview
- Autocomplete
- Functions



# Search clauses

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

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



## Advanced searching - functions reference

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

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



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

### approved()

*Only applicable if Jira Service Desk is installed and licensed.*

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

### On this page

[approved\(\)](#)

[approver\(\)](#)

[breached\(\)](#)

[cascadeOption\(\)](#)

[closedSprints\(\)](#)

[completed\(\)](#)

[componentsLeadByUser\(\)](#)

[currentLogin\(\)](#)

[currentUser\(\)](#)

[earliestUnreleasedVersion\(\)](#)

[elapsed\(\)](#)

[endOfDay\(\)](#)

[endOfMonth\(\)](#)

# Time based Functions

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

Issues created since the start of today

created > startOfDay()

# The unit qualifier

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

Tip: Use basic search to create the query

**created in the last 2 days (48 hours):**  
created > -2d

**created since the start of day 2 days ago:**  
created > startOfDay (-2d)

**created since the 15th of this month**  
created > startOfMonth (+14d)

# Takeaways

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

# Lab10-JQL

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

# Filters



# What will you learn?

- Create filters
- Describe board filters
- Use quick filters

# Topics

- Filters
- Board filters
- Quick filters



# Filters

Filters

Search issues

OTHER

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

[View all filters](#)

## My open issues

Save as

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

Search Switch to basic

1-1 of 1

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

# Saving a search creates a 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

Edited Save as

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

?

Search Switch to basic

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

1-1 of 1

# Save the filters

The screenshot shows the Jira interface for managing issues. On the left, there's a sidebar with a 'Filters' section containing various pre-defined filters like 'My open issues', 'Reported by me', etc. The 'My open issues' filter is currently selected. The main area displays a list of issues titled 'My open issues' with the query: `assignee = currentUser() AND statusCategory = "In Progress" order by updated DESC`. A modal window titled 'Save Filter' is open in the center, prompting the user to enter a 'Filter Name'. The input field contains the text 'My in progress'. At the bottom of the modal are 'Submit' and 'Cancel' buttons.

# The newly created filter

A screenshot of a Jira interface showing a newly created filter. On the left, a sidebar lists various filters, with 'My in progress' highlighted by a red arrow. The main area shows a search bar with the query 'assignee = currentUser() AND statusCategory = "In Progress" ORDER BY updated DESC'. Below it, a table displays one issue: PROJ-2 add item 2, assigned to admin, reported by admin, in IN PROGRESS status, unresolved, created on 01/Jul/20, updated on 02/Jul/20.

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   Save as   Details

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

1-1 of 1

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

1-1 of 1

# Edit filter details

The screenshot shows the Jira interface with a sidebar on the left containing filters like 'Search issues', 'STARRED', and 'My in progress'. The main area displays the 'My in progress' filter details. A red arrow points from the 'Details' link in the top right of the filter card to the 'Edit filter' dialog on the right.

**My in progress** (Owned by admin)

1-1 of 1

T	Key	Summary
<input checked="" type="checkbox"/>	PROJ-2	add item 2

1-1 of 1

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

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

**Edit filter**

Name \*

Description

Access  
 Private Only you

Project  
 Group  
 My organization...  
 Private

**Filter Subscription**

Recipients

Schedule  
 Daily  
 Days per Week  
 Days per Month  
 Advanced

Interval  
  
at

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

Email this filter, even if there are no issues found

# View all filters

Filters

Search issues

STARRED

My in progress

OTHER

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

[View all filters](#)

## Filters

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

[Create filter](#)

[Manage subscriptions](#)

[Copy filter](#)

[Edit](#)

[Delete](#)

# Edit filters query

Filters

Search issues

STARRED

My in progress

OTHER

My open issues

Reported by me

All issues

Open issues

Done issues

Viewed recently

Created recently

Resolved recently

Updated recently

[View all filters](#)

My in progress   Edited   Save   Details

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

1-1 of 1

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

1-1 of 1

# Topics

- Filters
- Board filters
- Quick filters



# Board filters

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

## Filter

Saved Filter

Filter for PRJCT board

[Edit Filter Query](#)

Filter Query

project = PRJCT ORDER BY Rank ASC

# Example Board filter

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

## Filter

- Saved Filter
- Two Projects
- Edit Filter Query

Shares

No shares

Edit Filter Shares

Filter Query

project in (PRJCT, PRJ) ORDER BY Rank ASC

## Kanban board

The screenshot shows a Jira Kanban board with two columns: 'SELECTED FOR DEVELOPMENT 2' and 'IN PROGRESS'. In the 'SELECTED FOR DEVELOPMENT 2' column, there are two items: 'add item 1' and 'add item 2'. Each item has a checkbox, a priority icon (blue for add item 1), and a three-dot menu. Red arrows point down to the first and second items respectively, indicating they are the ones being filtered by the query shown on the left.

# Topics

- Filters
- Board filters
- Quick filters



# Filtering a board: quick filters

Further filters issues displayed on a board

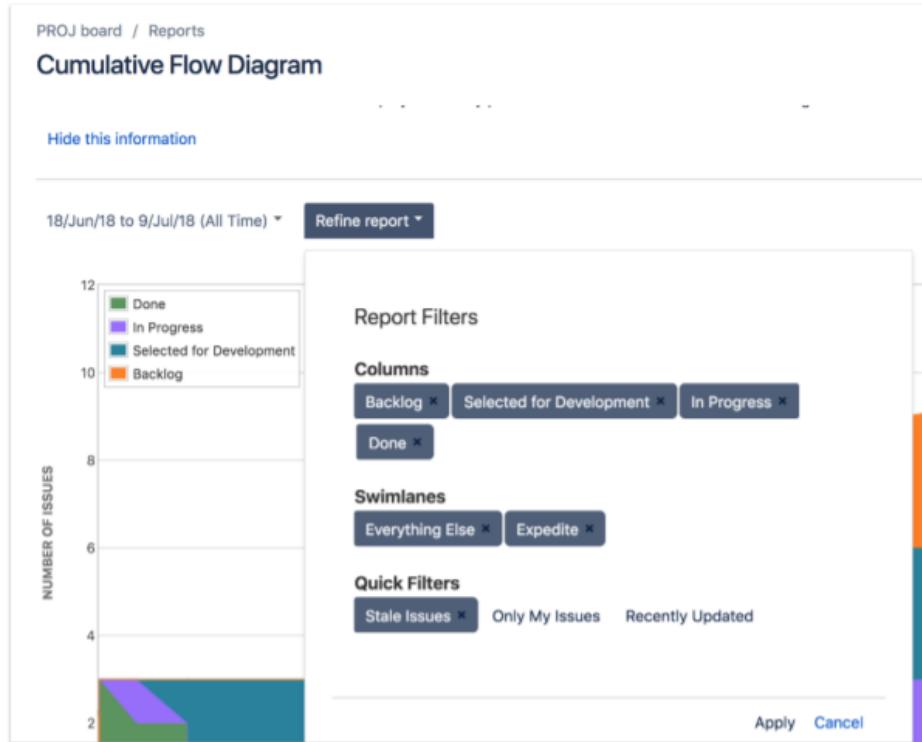
Projects / projectA / PROJ board

## Kanban board

The screenshot shows a Jira Kanban board titled "Kanban board" under "Projects / projectA / PROJ board". The board has three columns: "SELECTED FOR DEVELOPMENT" (2 items), "IN PROGRESS" (2 items), and "REVIEW" (0 items). Each column contains two cards, each with a title, a checkmark icon, a move icon, and a progress bar. The first card in the "IN PROGRESS" column is associated with "PROJ-3" and the second with "PROJ-4". The first card in the "SELECTED FOR DEVELOPMENT" column is associated with "PROJ-5". At the top of the board, there is a navigation bar with a search field, user icons, and filter buttons labeled "Only My Issues" and "Recently Updated". A red box highlights this top navigation area.

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

# Quick filters and report



# Takeaways

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

# Lab 11- Filters

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

# Epics



# What will you learn?

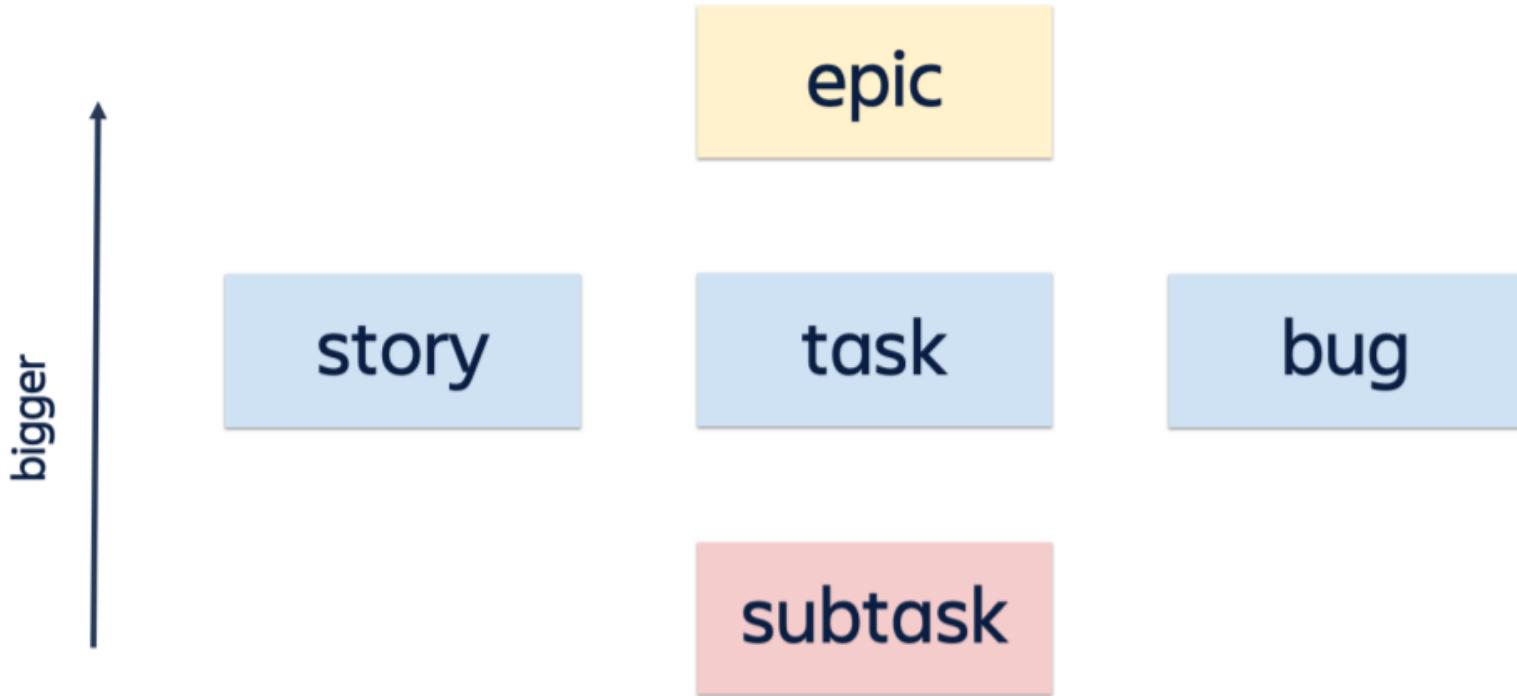
- Describe epics
- Work with epics
- Manage epics in the backlog

# Topics

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



# Jira's issue type hierarchy



# Epics

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

Create issue

Import issues Configure fields ▾

Project\* projectA (PROJ)

Issue Type\* Epic

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

Epic Name\* Big Feature A

Provide a short name to identify this epic.

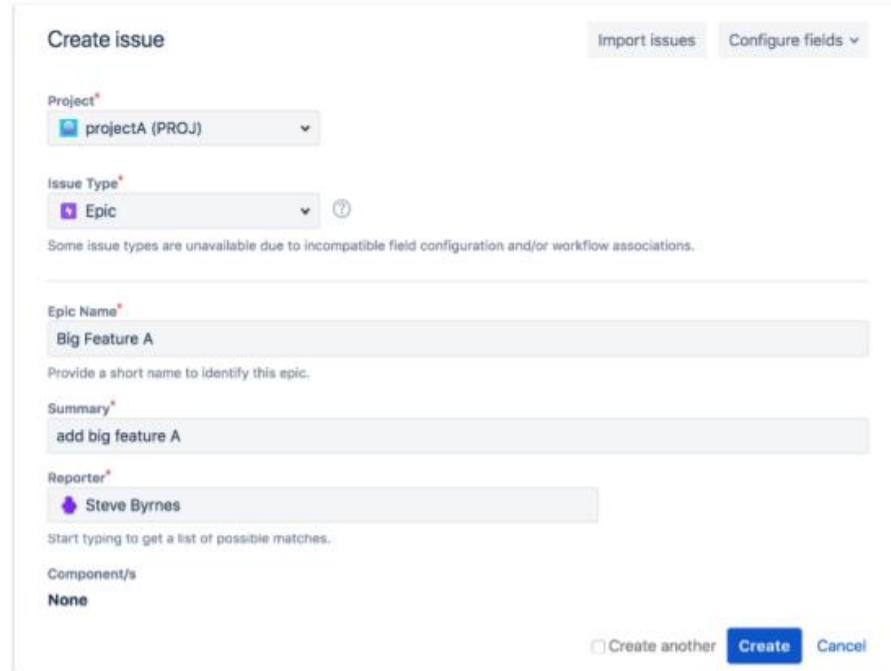
Summary\* add big feature A

Reporter\* Steve Byrnes

Start typing to get a list of possible matches.

Component/s None

Create another Create Cancel



# Why Epics?

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

# Topics

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



# Creating an epic

### Create issue

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

---

**Project\***

projectA (PROJ) ▾

**Issue Type\***

Epic ▾ ⓘ

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

---

**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 v <sup>v</sup> ~~A~~ <sup>v</sup> ~~v~~ |

**Issue Type\***

Epic ▾ ⓘ

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

---

**Epic Name\***

Feature A

Provide a short name to identify this epic.

**Summary\***

add feature A

Create another

**Create** Cancel

# Managing the issue of an epic

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

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

**Epic Information:**

- Description:** Add feature A
- Assignee:** Unassigned
- Reporter:** admin

**Issues in this epic:**

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

**Comment Section:**

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

**Feature A (Issue Details):**

- Story Points: None
- Original Estimate: 0m

# Epic link

A field in the child issue pointing to the parent epic

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

PROJ-7 / PROJ-B

add feature A task 1

Attach Create subtask Link issue ...

Description  
Add a description...

Activity  
Show: Comments History Work log

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

Backlog

Assignee  
Unassigned

Reporter  
admin

Labels  
None

Epic Link

Feature A

Created 6 minutes ago Updated 6 minutes ago Configure

# Adding an existing issue to an epic

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

The screenshot shows a Jira issue page for 'add item 2' in project PROJ-2. The status is 'In Progress'. The 'Epic Link' field is highlighted with a red arrow pointing to it. The 'Select Epic' dropdown menu is open, showing 'Feature A' and 'PROJ-7'.

PROJ-2

In Progress

Assignee: admin

Reporter: admin

Description: Add a description...

Activity: Show: Comments (selected), History, Work log

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

Epic Link: Select Epic

Feature A  
PROJ-7

# Searching for issues of an epic

Search Save as

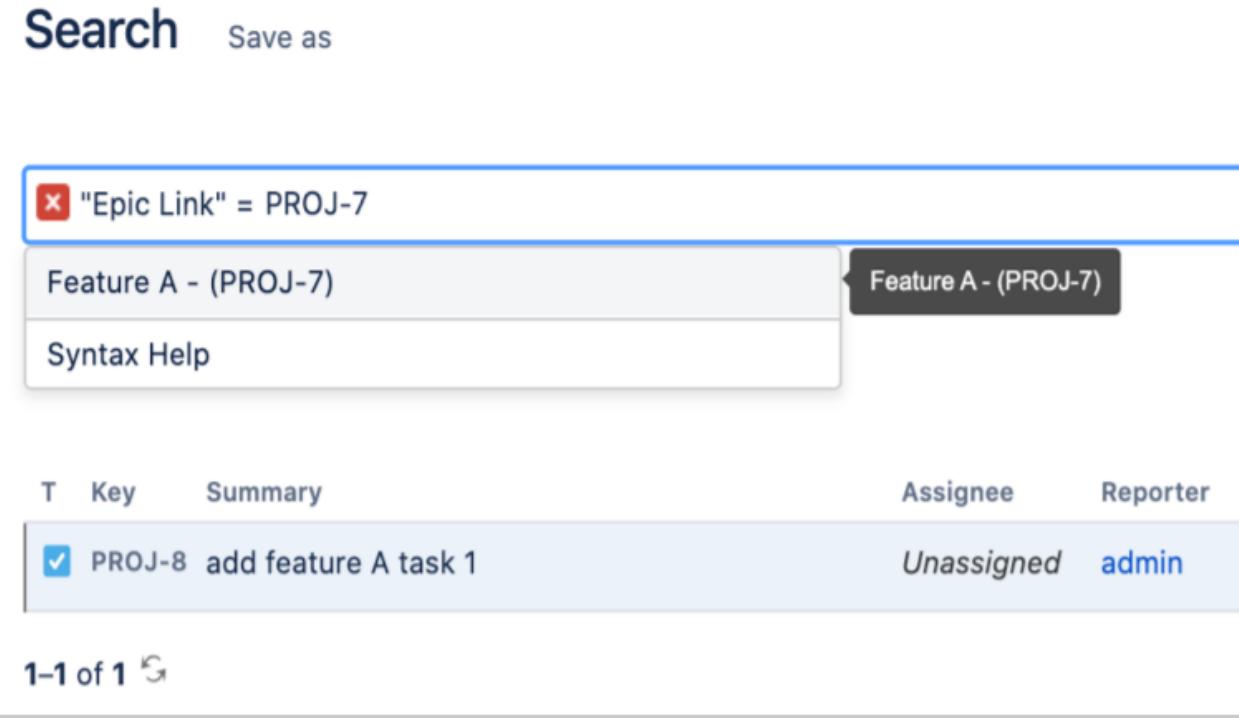
"Epic Link" = PROJ-7

Feature A - (PROJ-7) Feature A - (PROJ-7)

Syntax Help

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

1-1 of 1 ⌂



# Epics labels on a board

Projects / projectA / PROJ board

## Kanban board

The screenshot shows a Jira Kanban board for projectA. The board has three columns: BACKLOG 3, SELECTED FOR DEVELOPM... 2, and another column partially visible. In the BACKLOG 3 column, there are two cards:

- add feature A**  
Feature A  
PROJ-7
- add feature A task 1**  
Feature A  
PROJ-8

Both cards have red arrows pointing to the "Feature A" label. In the SELECTED FOR DEVELOPM... 2 column, there are two cards:

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

At the top right of the board, there is a search bar, a user icon (A), a filter icon (8), and the text "Only My Issues".

# Epic based swimlanes

Projects / projectA / PROJ board

## Kanban board

Projects / projectA / PROJ board

Kanban board

BACKLOG 3    SELECTED FOR DEVELOPM... 2    IN PROGRESS 2

Only My Issues    Recently Updated

Feature A 1 issue

- add feature A task 1
- Feature A
- 
- PROJ-B

Issues without epics 6 issues

- add item 6
- 
- PROJ-6

- add item 3
- 
- PROJ-3

- add item 2
- 
- PROJ-2

- add item 5
- add item 4

# Topics

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



# Creating an epic from a backlog

Projects / projectB / PRJ board

## Backlog

The screenshot shows a Jira backlog interface. At the top, there is a navigation bar with 'Projects / projectB / PRJ board' and a search bar. Below the navigation is a header with 'Backlog' and user profile icons. A red arrow points upwards from the bottom of the 'VERSIONS' dropdown menu towards the 'Create epic' button. The 'VERSIONS' dropdown has two options: 'EPICS' and 'All issues', with 'All issues' currently selected. To the right of the dropdown is a 'Create epic' button and an 'X' button. The main content area displays a section titled 'PRJ Sprint 1' which contains 2 issues. Below this is a 'Backlog' section containing 1 issue. Each section has a 'add item' button.

VERSIONS

EPICS Create epic X

All issues

PRJ Sprint 1 2 issues

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

A ...

add item 1

add item 2

Backlog 1 issue

add item 3

+ Create issue

# Epics panel

Can monitor the status of issues in the epic

Projects / projectB / PRJ board

## Backlog

The screenshot shows the Jira Backlog interface for the 'PRJ board' under 'projectB'. At the top, there are filters for 'Only My Issues' and 'Recently Updated'. A red box highlights the 'EPICS' section on the left, which includes a 'Create epic' button and a dropdown menu for 'All issues'. Below this are sections for 'feature B' (with 'PRJ-4 add feature B'), 'Issues' (3), 'Completed' (2), 'Unestimated' (0), and 'Estimate' (6). There are also links for 'Create issue in epic' and 'View linked pages'. To the right, there are three main sections: 'PRJ Sprint 1' (2 issues), 'Backlog' (1 issue), and another 'Backlog' section. Each section has a 'Create sprint' button and a 'Create issue' link at the bottom.

EPICS Create epic

All issues

feature B  
PRJ-4 add feature B

Issues 3

Completed 2

Unestimated 0

Estimate 6

Create issue in epic

View linked pages

Issues without epics

PRJ Sprint 1 2 issues

0 0 3 Linked pages 0 ...

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

A ...

add item 1 feature B PRJ-1 ↑ 1

add item 2 feature B PRJ-2 ↑ 2

Backlog 1 issue Create sprint ...

add item 3 feature B PRJ-3 ↑ 3

+ Create issue

# Mark an epic as done

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

Projects / projectB / PRJ board

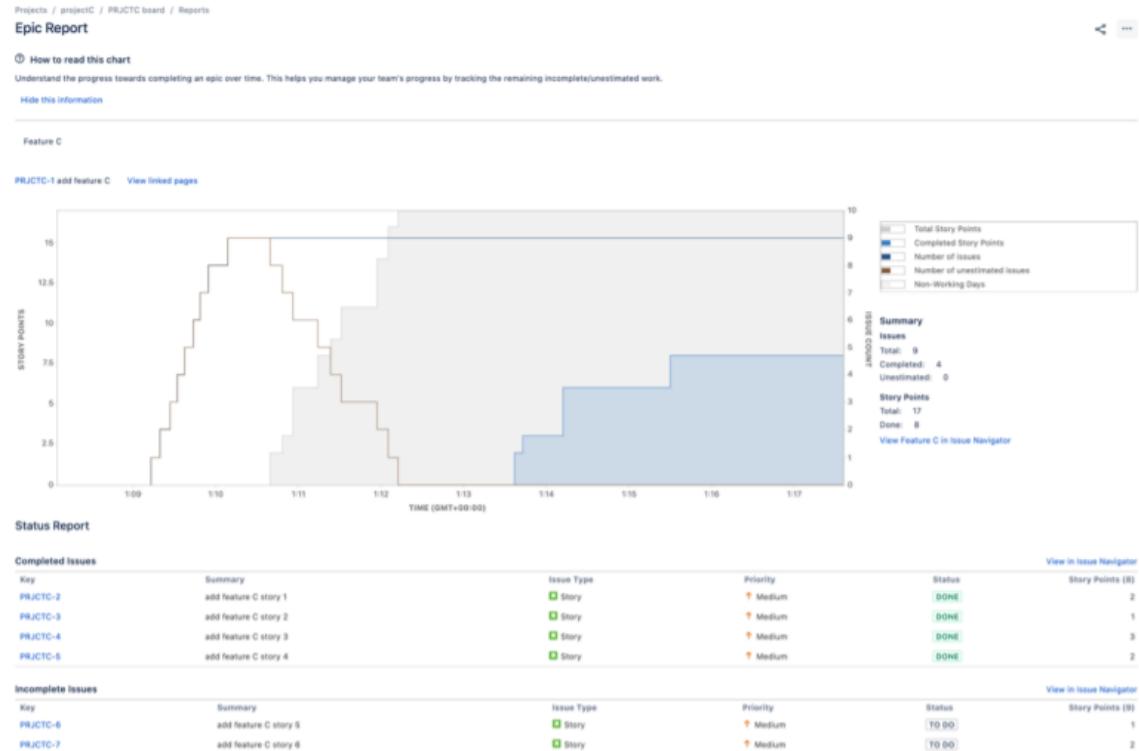
## Backlog

The screenshot shows the Jira Backlog interface for the 'PRJ board'. A modal window is open over the backlog, specifically for an epic titled 'feature B'. The modal has tabs for 'EPICS' (selected), 'Create epic', and 'x'. It displays the following information:

- All issues
- Issues: 3
- Completed: 3
- Unestimated: 0
- Estimate: 6

Below this, there are links for 'Create issue in epic', 'View linked pages', and 'Issues without epics'. On the right side of the modal, there is a 'Color' section with a grid of colored squares and a 'Mark as Done' button at the bottom, which is highlighted with a red border.

# Epics Report



# Scrum - Epics burndown report



# Takeaways

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

# Lab12- Epics

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

# Dashboards



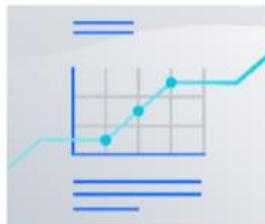
# What will you learn?

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

# Visualizing work



**Boards**



**Reports**



**Search**



**Dashboards**

# Dashboards

The image shows two screenshots of a dashboard interface.

**Left Screenshot: Alana's dashboard**

- Sprint Health Gadget:** Displays "PROJB Sprint 3 - PROJB board".
  - Overall sprint progress (Story Points): 5 days left, 5 Story Points.
  - Time elapsed: 0 %
  - Work complete: 0 %
  - Scope change: 0 %
  - Blocker: 0
  - Flagged: 0
- Sprint Burndown Gadget:** A burndown chart from Nov 3 to Nov 9. The Y-axis is Story Points (0 to 6). The X-axis is TIME. A red vertical line marks the end of the sprint on Nov 9 at 5 Story Points. A legend indicates: Guideline (blue), Remaining Values (red), and Non-Working Days (grey).

**Right Screenshot: Activity Stream and Assigned to Me**

- Activity Stream:** Shows activity history for "Your Company JIRA".
  - Alana Grant updated the Sprint of PROJB-6 - add item 8 (1 minute ago)
  - Alana Grant updated the Sprint of PROJB-7 - add item 7 (1 minute ago)
  - Alana Grant updated the Story Points of PROJB-7 - add item 9 (1 minute ago)
  - Alana Grant created PROJB-7 - add item 9 (2 minutes ago)
  - Alana Grant updated the Story Points of PROJB-6 - add item 8 (2 minutes ago)
- Assigned to Me:** A list of tasks assigned to the user.

Key	Summary	Due Date
TIS-57	Buttons need to be red	Nov 10
TIS-58	Add Features for Flight School Teachers	Nov 10
TIS-59	Enhancements for Flight School	Nov 10
PROJ-B	fix bug 1	Nov 10
PROJ-11	add big feature A	Nov 10
PROJ-F-1	add item 1	Nov 10
PROJ-F-2	Create initial release	Nov 10
PROJ-F-3	add item 6	Nov 10
PROJB-5	add item 7	Nov 10

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

# Types of Gadgets

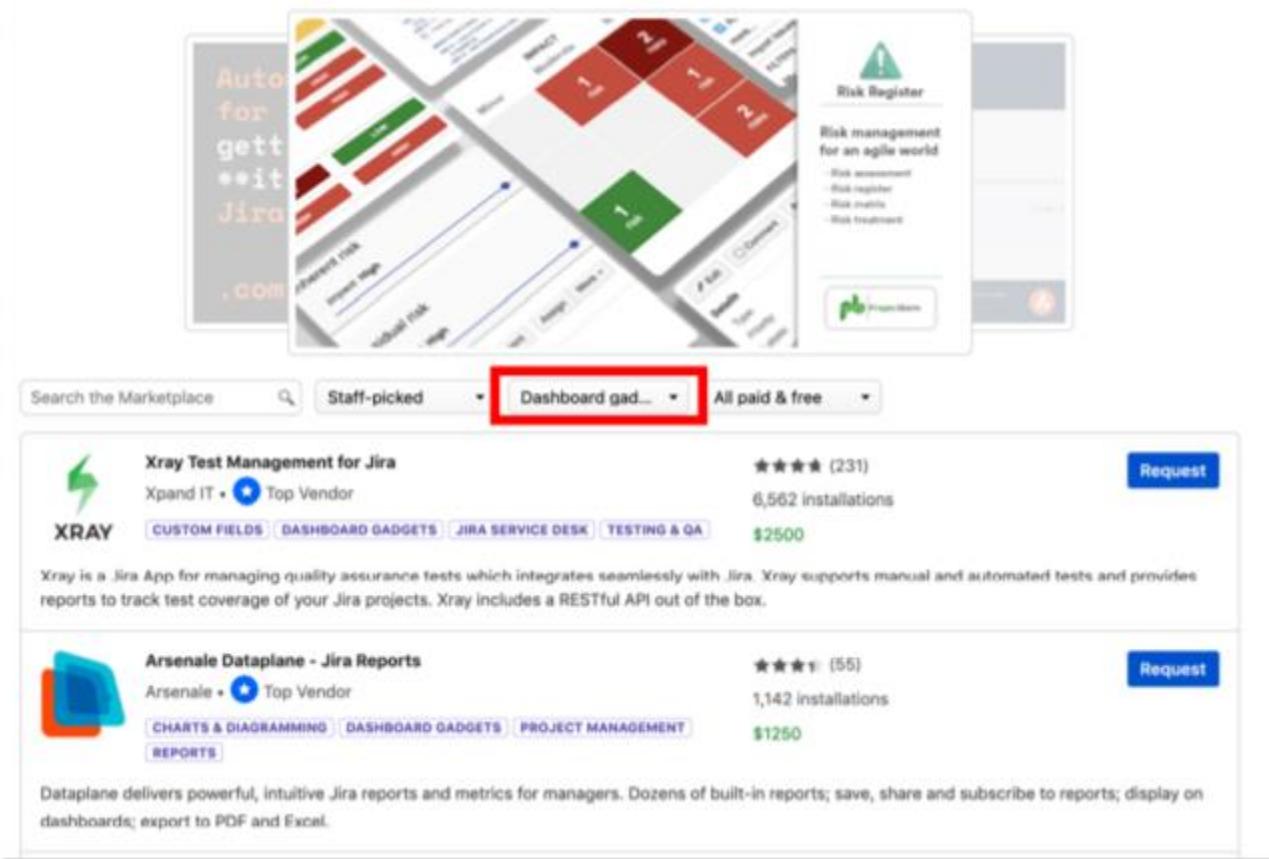
## Add a gadget

Search

CATEGORIES

All	31
Charts	8
Jira	27
Other	3
Wallboard	7

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



The screenshot shows the Atlassian Marketplace interface. At the top, there's a search bar labeled "Search the Marketplace" with a magnifying glass icon, followed by dropdown menus for "Staff-picked", "Dashboard gad...", and "All paid & free". A red box highlights the "Dashboard gad..." dropdown. Below the search area, two app cards are displayed:

**Xray Test Management for Jira**  
Xray Test Management for Jira integrates with Jira for quality assurance tests. It has a 4-star rating (231 reviews), 6,562 installations, and costs \$2500. It includes tags for CUSTOM FIELDS, DASHBOARD GADGETS, JIRA SERVICE DESK, TESTING & QA, and REPORTS. A "Request" button is available.

**Arsenale Dataplane - Jira Reports**  
Arsenale Dataplane provides Jira reports and metrics. It has a 3-star rating (55 reviews), 1,142 installations, and costs \$1250. It includes tags for CHARTS & DIAGRAMMING, DASHBOARD GADGETS, PROJECT MANAGEMENT, and REPORTS. A "Request" button is available.

- Contains more dashboard gadgets
- Select “Dashboard gadgets”

# Adding gadgets

Add gadget   Edit layout   ...

**Add a gadget**

Search

CATEGORIES

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

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

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

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

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

# Configuring a gadget

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

Minimize

Delete

Edit

Sprint Health Gadget

Board PROJB board

Show board name

Show sprint name

Show assignees

Sprint Next Sprint Due (auto)

Auto refresh  Update every 15 minutes

Save Cancel

# Sharing Dashboards

The screenshot shows a user interface for managing a dashboard. On the left, there's a form with fields for Name (Team Sprint Dashboard), Description (empty), Favorite (star icon), and Shares (Not shared). Below these are dropdown menus for Add Shares (Project, Teams in Space) and a status message: "Shared with everyone with permission to browse the 'Teams in Space' space". At the bottom are Update and Cancel buttons. A context menu is open on the right, listing options like Copy Dashboard, Edit Dashboard, Share Dashboard (which is selected and highlighted in blue), Delete Dashboard, Find Dashboards, Create Dashboard, View as wallboard, View wallboard slideshow, and Set up wallboard slideshow.

Name \* Team Sprint Dashboard

Description

Favorite ★

Shares Not shared

Add Shares Project ▾ Teams in Space ▾

Shared with everyone with permission to browse the 'Teams in Space' space

Update Cancel

Share Dashboard

Copy Dashboard

Edit Dashboard

Delete Dashboard

Find Dashboards

Create Dashboard

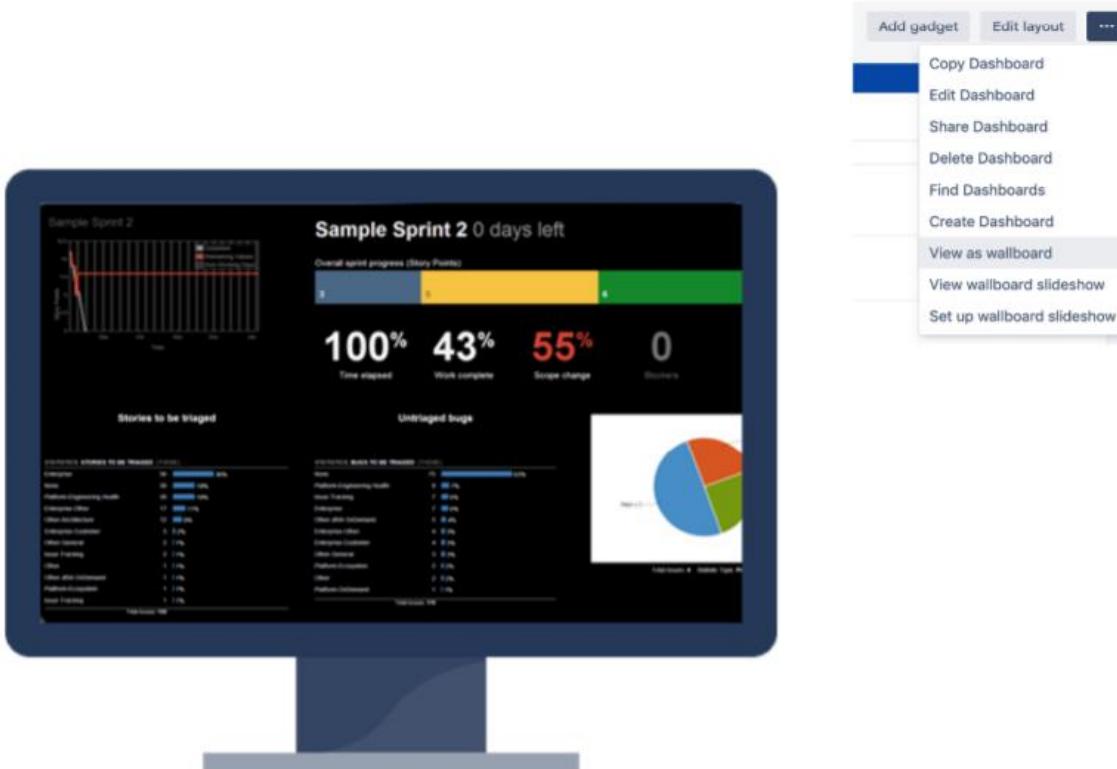
View as wallboard

View wallboard slideshow

Set up wallboard slideshow

# Wallboards

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

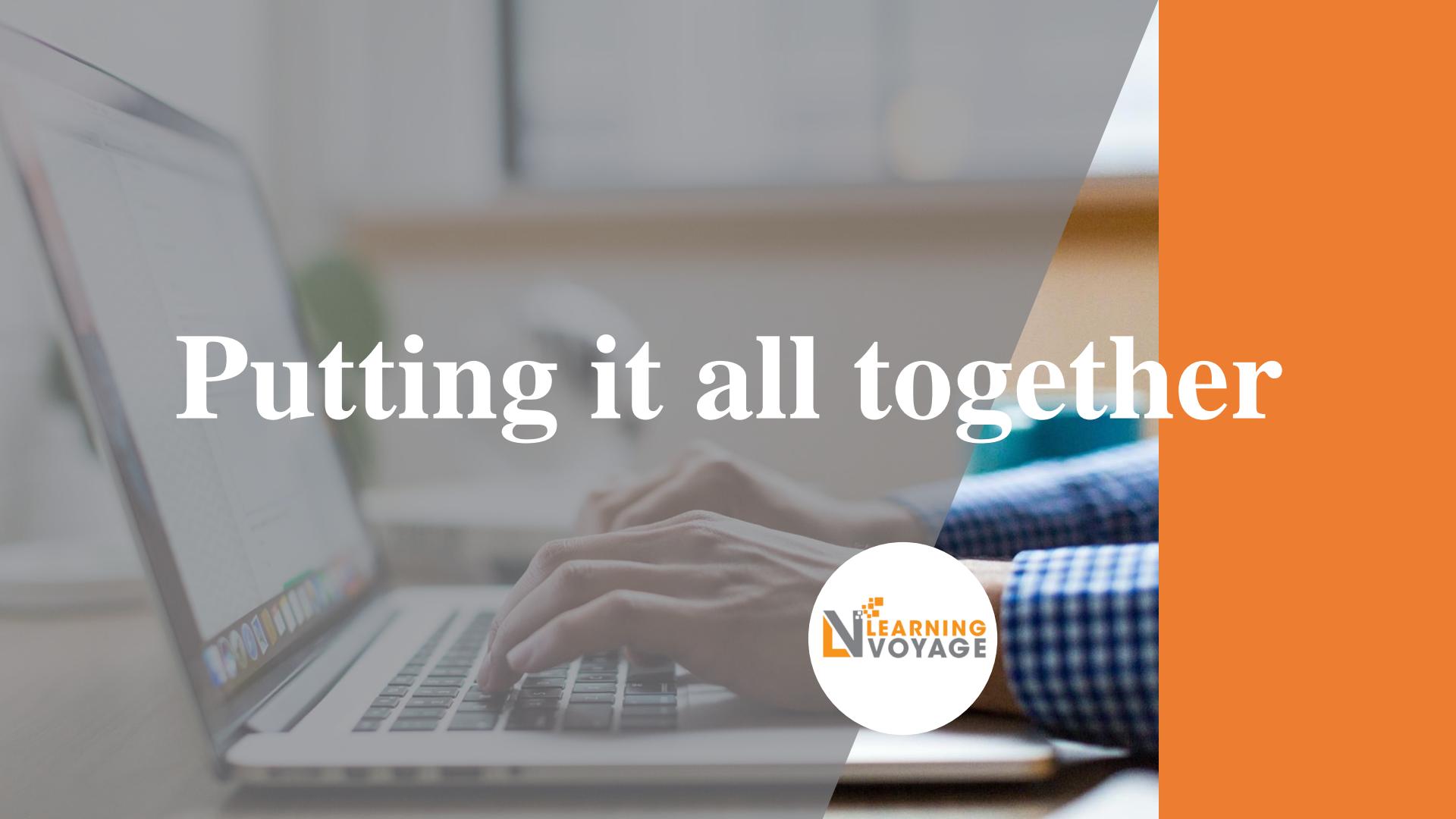


# Takeaways

- 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 13 - Dashboards

- Create a dashboard
- Display a dashboard as a wallboard

A blurred background image of a person's hands typing on a laptop keyboard. The laptop screen shows some text, though it's not legible. To the right of the laptop, there's a vertical orange bar.

# Putting it all together



# Topics

- Course overview
- Jira family
- Wrap up



# Combined lean and agile principles

## 1. Empower the team

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

## 2. Visualize work

## 3. Experiment using the scientific method

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

## 4. Plan, develop and deliver incrementally

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

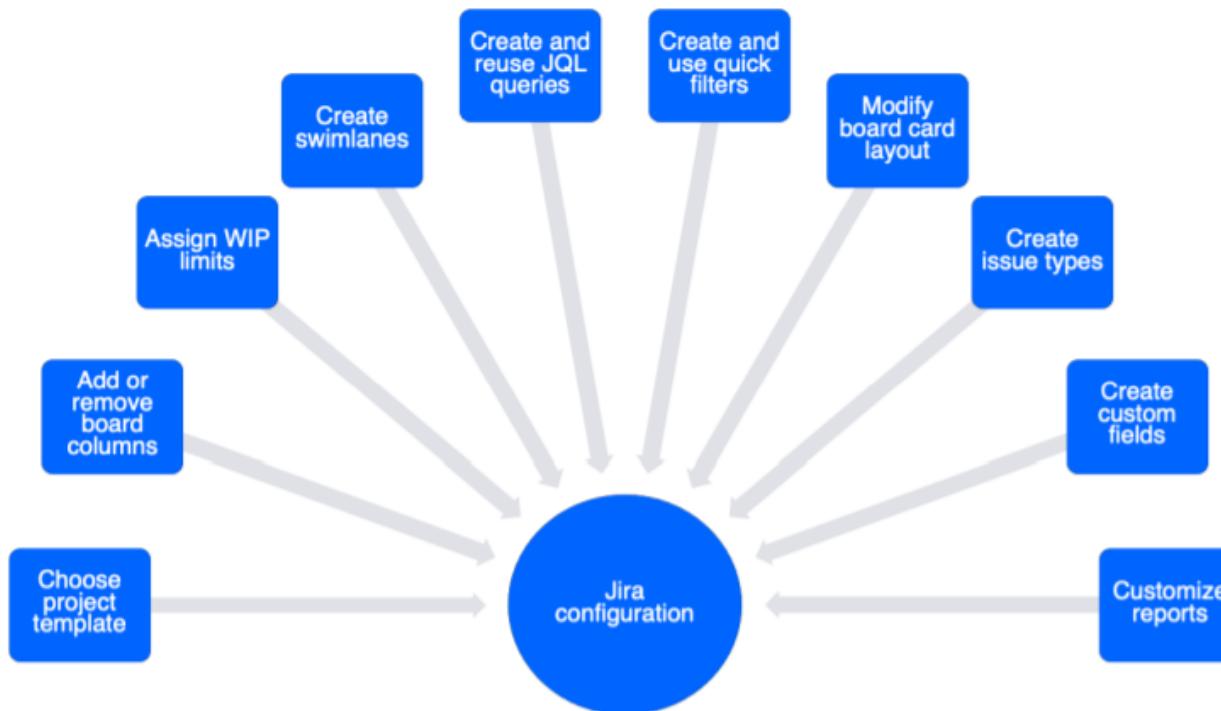
## 5. Improve the "flow" of value

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

## 6. Build quality in

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

# Ways to configure jira to match your team's process



# Topics

- Course overview
- Jira family
- Wrap up



# Jira family of products



**Jira Core**  
Business project management  
software

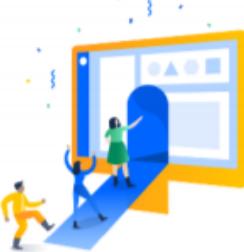


**Jira Software**  
Plan, track, and release  
software



**Jira Service Desk**  
Service desk software for IT  
teams

## Onboarding



## Tasks



Create email account



Acquire Laptop



Video chat with new employee



Jira Core

Create & manage

Track progress



## Mobile App Development



Manage backlog



Backlog



Jira Software



Manages board



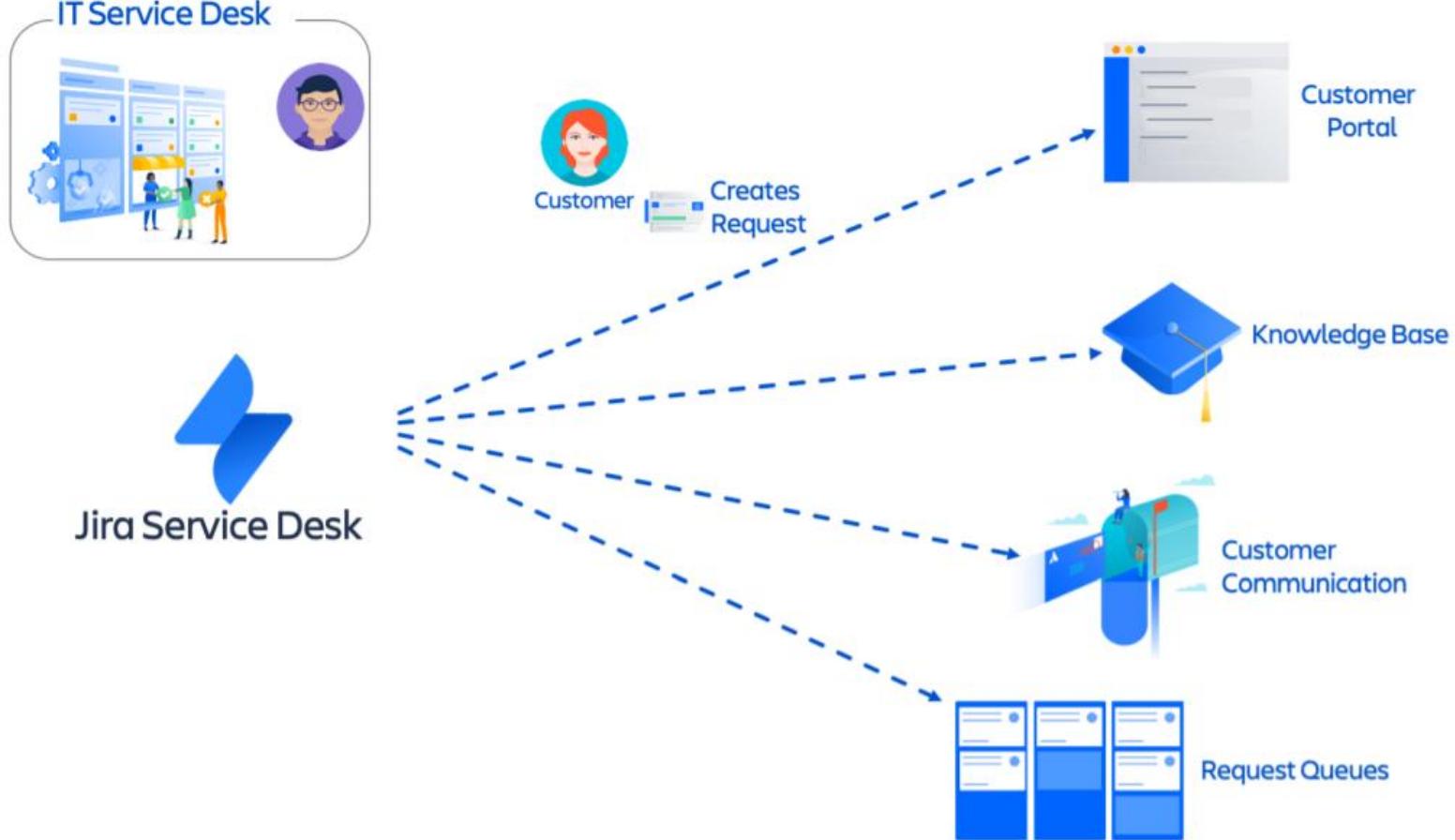
Board



Monitor pull requests



Code Repository



# Lab 14- Final project (optional)

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

# Topics

- Course overview
- Jira family
- Wrap up



# Congratulations on completing the course!

