Instructor Notes:

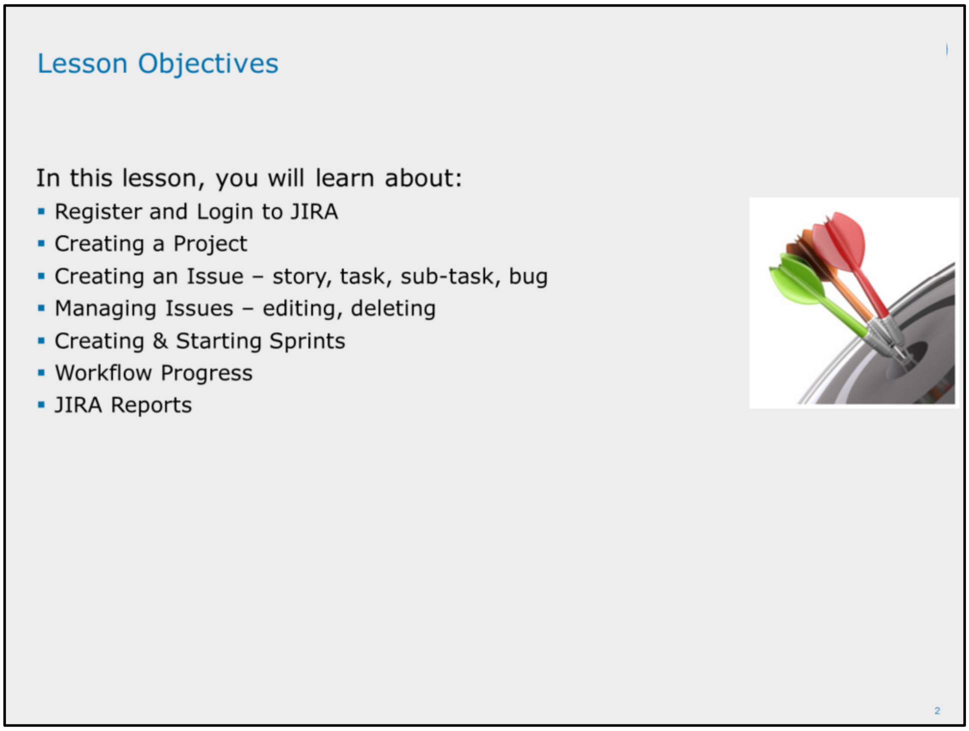
Add instructor notes here.

JIRA

Lesson 03: Hands-on JIRA

Instructor Notes:

Add instructor notes here.



Lesson Objectives

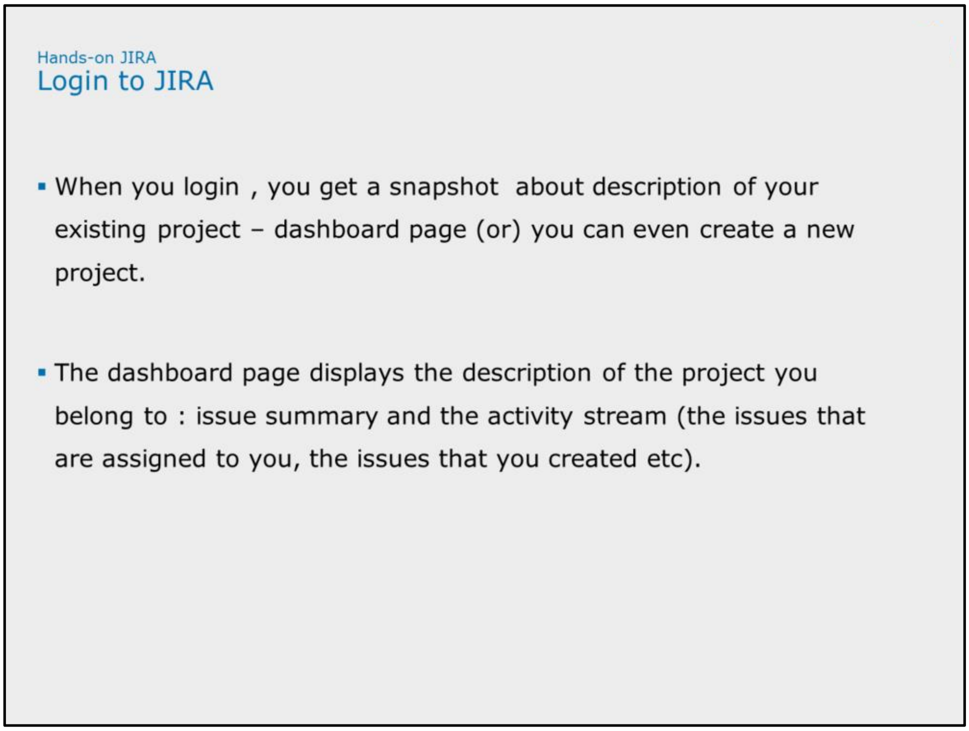
In this lesson, you will learn about:

- Register and Login to JIRA
- Creating a Project
- Creating an Issue – story, task, sub-task, bug
- Managing Issues – editing, deleting
- Creating & Starting Sprints
- Workflow Progress
- JIRA Reports

2

Instructor Notes:

Add instructor notes here.



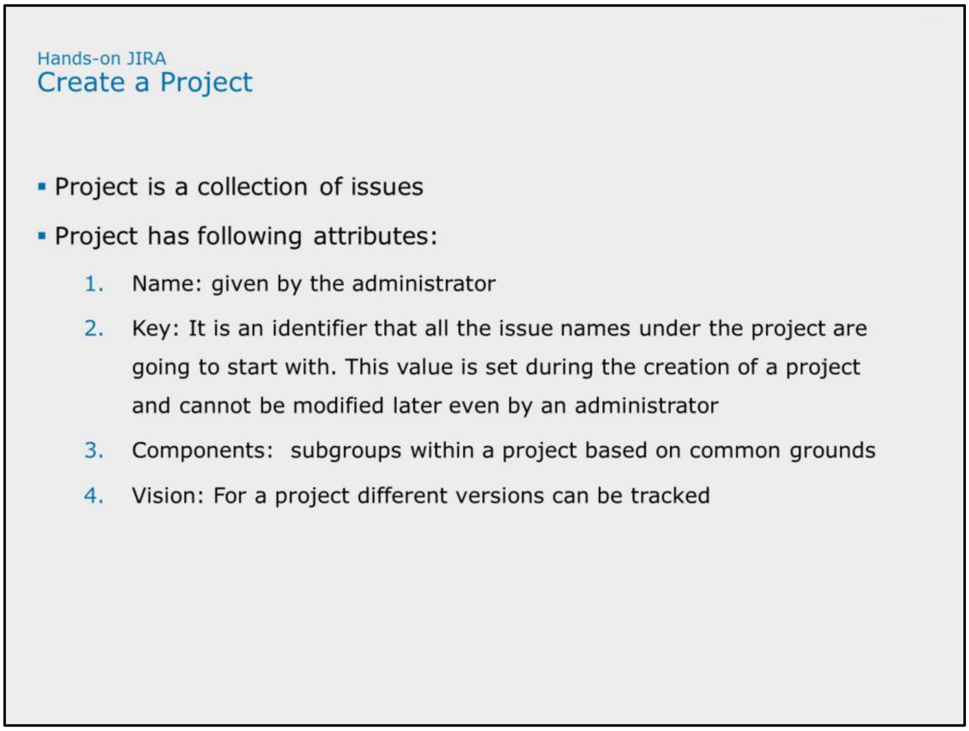
Hands-on JIRA

Login to JIRA

- When you login , you get a snapshot about description of your existing project – dashboard page (or) you can even create a new project.
- The dashboard page displays the description of the project you belong to : issue summary and the activity stream (the issues that are assigned to you, the issues that you created etc).

Once logged in, the dashboard page is displayed (unless otherwise chosen) to the user. The dashboard page gives a snap shot about the description of the project you belong to; issue summary and the activity stream (the issues that are assigned to you, the issues that you created etc).

Instructor Notes:



Hands-on JIRA

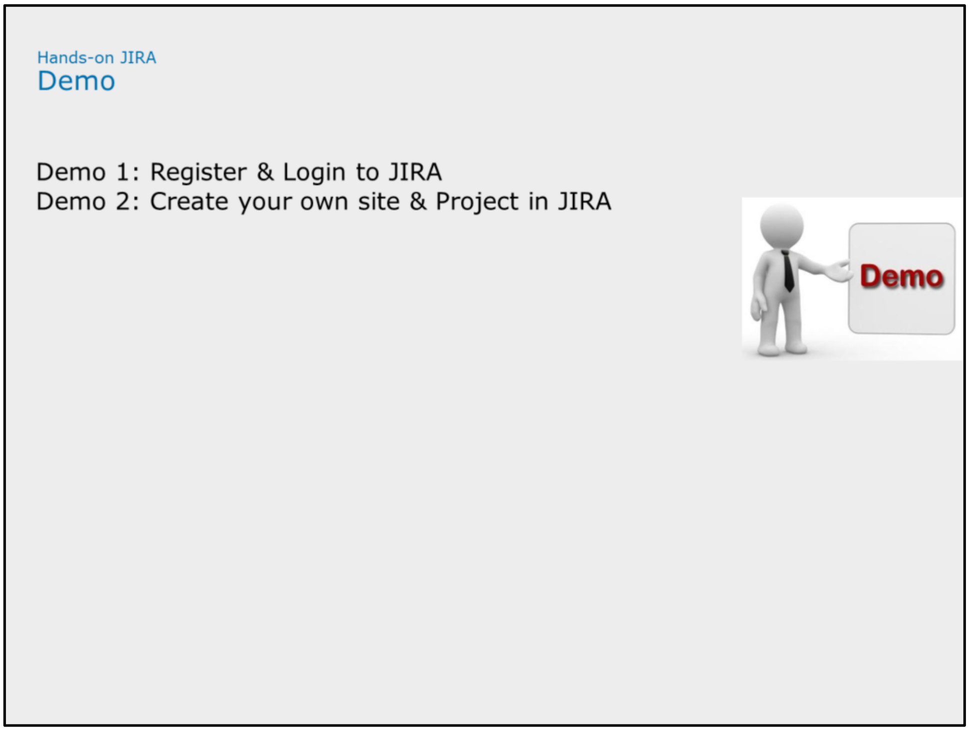
Create a Project

- Project is a collection of issues
- Project has following attributes:
  1. Name: given by the administrator
  2. Key: It is an identifier that all the issue names under the project are going to start with. This value is set during the creation of a project and cannot be modified later even by an administrator
  3. Components: subgroups within a project based on common grounds
  4. Vision: For a project different versions can be tracked

Project: Every issue belongs to a project. You can choose the same by clicking on the drop down and choosing the project that you want this issue to belong to.

Instructor Notes:

Add instructor notes here.



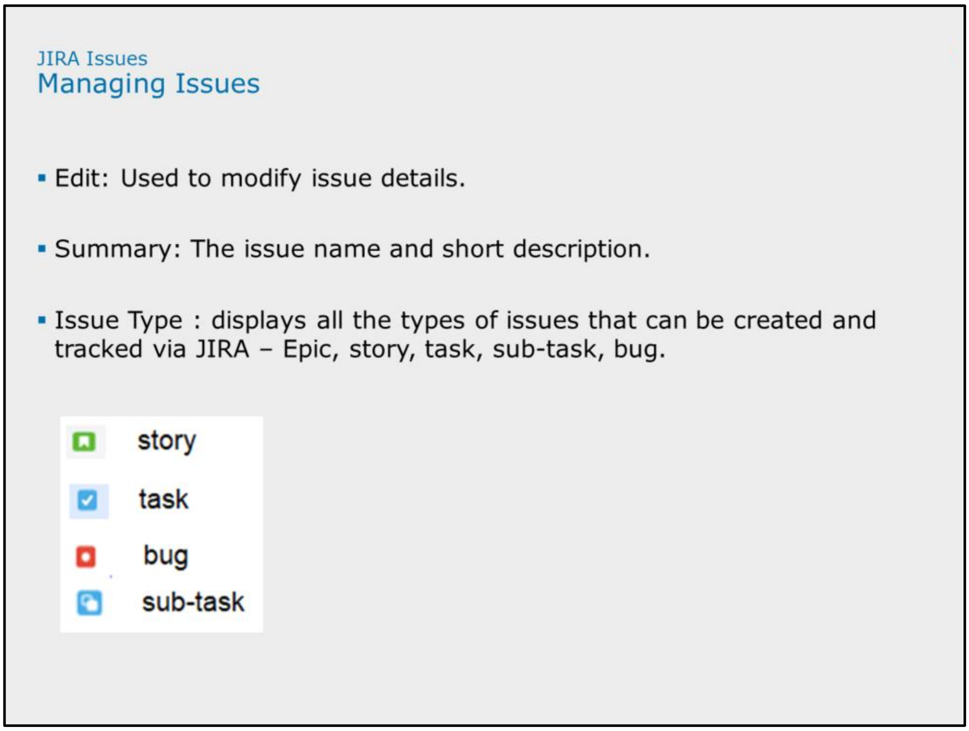
Hands-on JIRA

Demo

Demo 1: Register & Login to JIRA
Demo 2: Create your own site & Project in JIRA

Add the notes here.

Instructor Notes:



JIRA Issues

Managing Issues

- Edit: Used to modify issue details.
- Summary: The issue name and short description.
- Issue Type : displays all the types of issues that can be created and tracked via JIRA – Epic, story, task, sub-task, bug.

__Summary__: Give your bug a title here. When used right, this field can be very successful at transmitting a lot of critical information. Some aspects to note here:

A bug/defect is essentially something that is not right. The right way to approach a bug title is to concisely define ‘what’s wrong’.

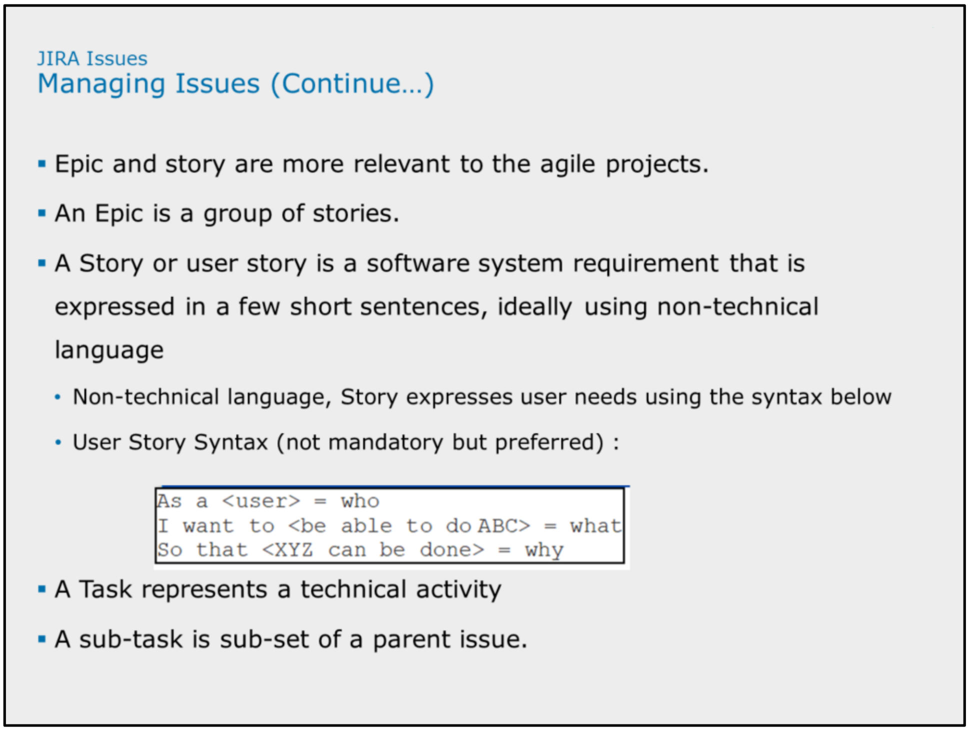
Example of bug summary : “There should be an option to clear the contents on the screen”. When I read this my initial reaction is going to be – “Okay, there should be- but what’s the problem here? Is the option not present at all? Or is the options present and not clearing the content?”

It is also agreed, that when I open this bug and look into it in detail, I am sure I will find the answer to this question.

However, the emphasis here is to use this “Summary” field in the most efficient manner. Therefore, a very apt summary/title would be “The option to clear the contents of the home login page does not clear the fields when clicked.”

In the limited space that this field provides try to write your title in a way that communicates the exact issue without any ambiguity.

Instructor Notes:



JIRA Issues

Managing Issues (Continue…)

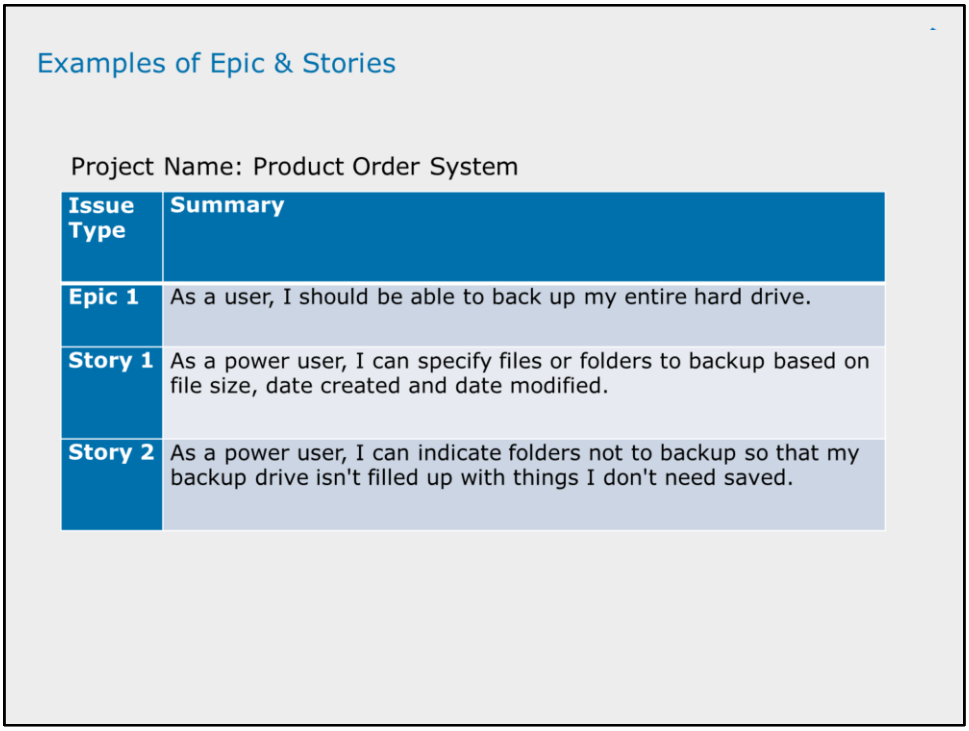
- Epic and story are more relevant to the agile projects.
- An Epic is a group of stories.
- A Story or user story is a software system requirement that is expressed in a few short sentences, ideally using non-technical language
  - Non-technical language, Story expresses user needs using the syntax below
  - User Story Syntax (not mandatory but preferred) :

```
As a <user> = who
I want to <be able to do ABC> = what
So that <XYZ can be done> = why
```

- A Task represents a technical activity
- A sub-task is sub-set of a parent issue.

Example of Story : As a customer, I want to filter the product catalog by product attributes such as price, range, brand, colour, size etc. so that I can make a correct purchase decision.

Instructor Notes:

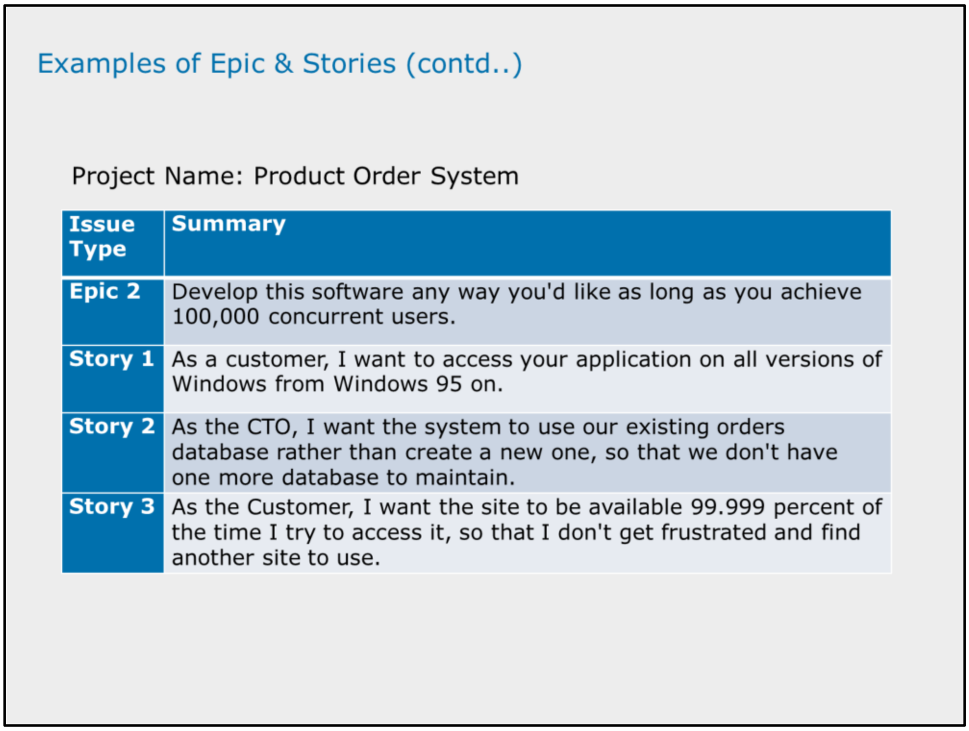


Examples of Epic & Stories

Project Name: Product Order System

Issue Type	Summary
Epic 1	As a user, I should be able to back up my entire hard drive.
Story 1	As a power user, I can specify files or folders to backup based on file size, date created and date modified.
Story 2	As a power user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved.

Instructor Notes:

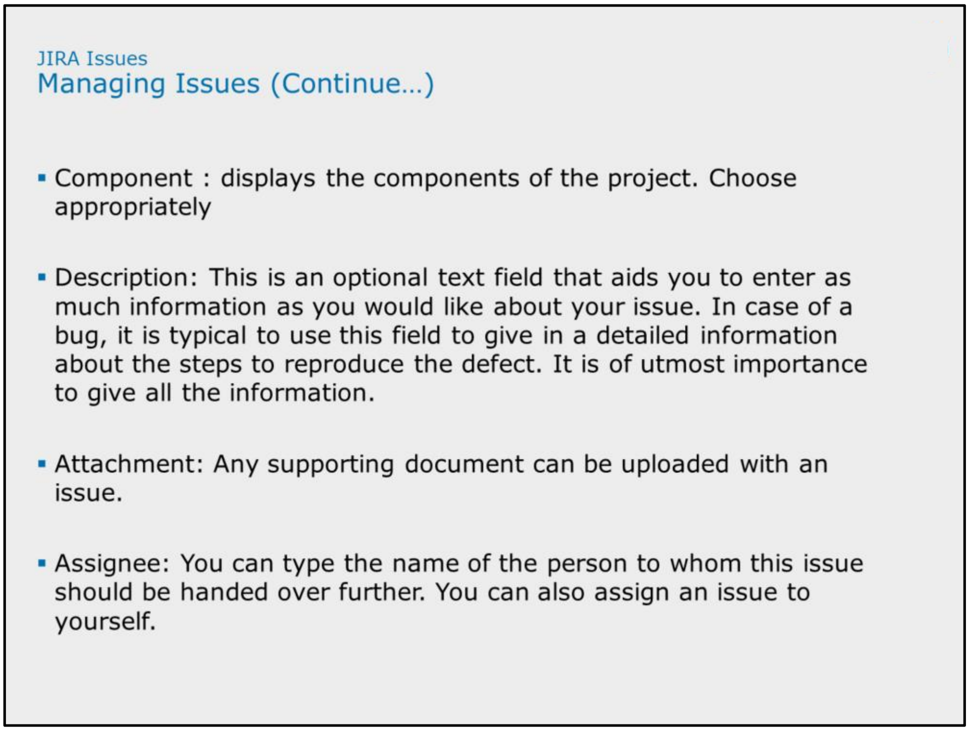


Examples of Epic & Stories (contd..)

Project Name: Product Order System

Issue Type	Summary
Epic 2	Develop this software any way you'd like as long as you achieve 100,000 concurrent users.
Story 1	As a customer, I want to access your application on all versions of Windows from Windows 95 on.
Story 2	As the CTO, I want the system to use our existing orders database rather than create a new one, so that we don't have one more database to maintain.
Story 3	As the Customer, I want the site to be available 99.999 percent of the time I try to access it, so that I don't get frustrated and find another site to use.

Instructor Notes:



JIRA Issues

Managing Issues (Continue…)

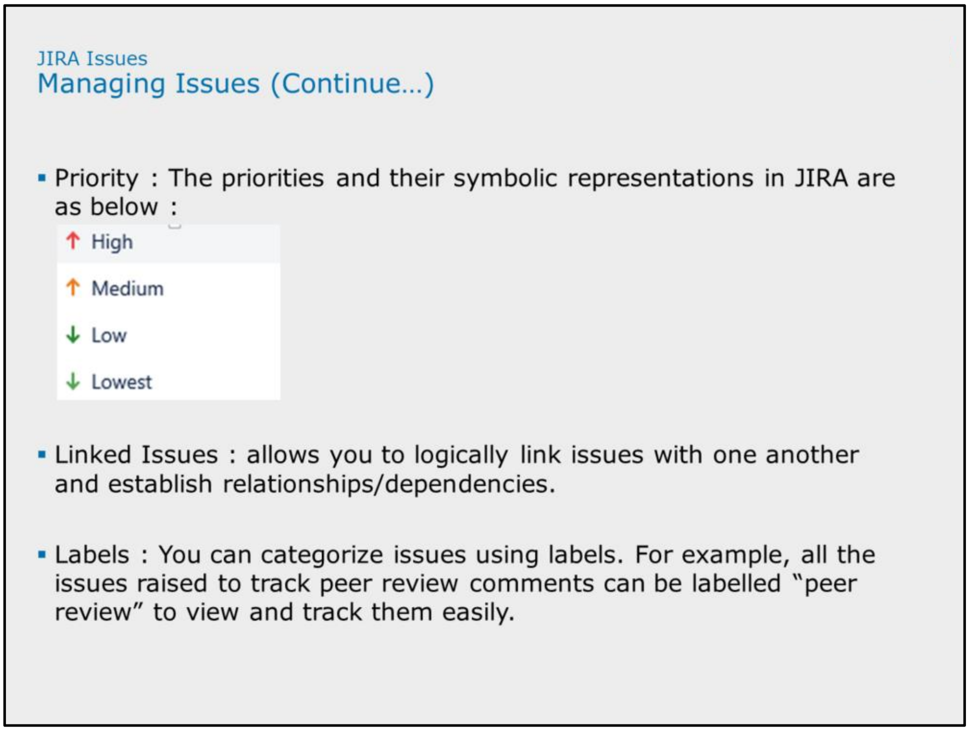
- Component : displays the components of the project. Choose appropriately
- Description: This is an optional text field that aids you to enter as much information as you would like about your issue. In case of a bug, it is typical to use this field to give in a detailed information about the steps to reproduce the defect. It is of utmost importance to give all the information.
- Attachment: Any supporting document can be uploaded with an issue.
- Assignee: You can type the name of the person to whom this issue should be handed over further. You can also assign an issue to yourself.

Description Example :

Say, there are two fields – dependent ones- State and City. When I choose State from the drop down, in the City field it should display the respective cities in the state I chose.

If I raised a bug as "The cities are empty for some states I selected". The description can be :
1) Enter the site
2) Click on the address page
3) Enter the other details like name, street address etc.
4) Choose the state as "MH" and click on the "City" drop down – all the cities will be displayed and the user can select a city as needed.
5) Choose the state as "AP" and click on the "City" drop down – the list will be empty.
6) The cities are empty for the states "TN" and "UP" also.

Instructor Notes:



JIRA Issues

Managing Issues (Continue…)

- Priority : The priorities and their symbolic representations in JIRA are as below :

  ↑ High

  ↑ Medium

  ↓ Low

  ↓ Lowest

- Linked Issues : allows you to logically link issues with one another and establish relationships/dependencies.

- Labels : You can categorize issues using labels. For example, all the issues raised to track peer review comments can be labelled “peer review” to view and track them easily.

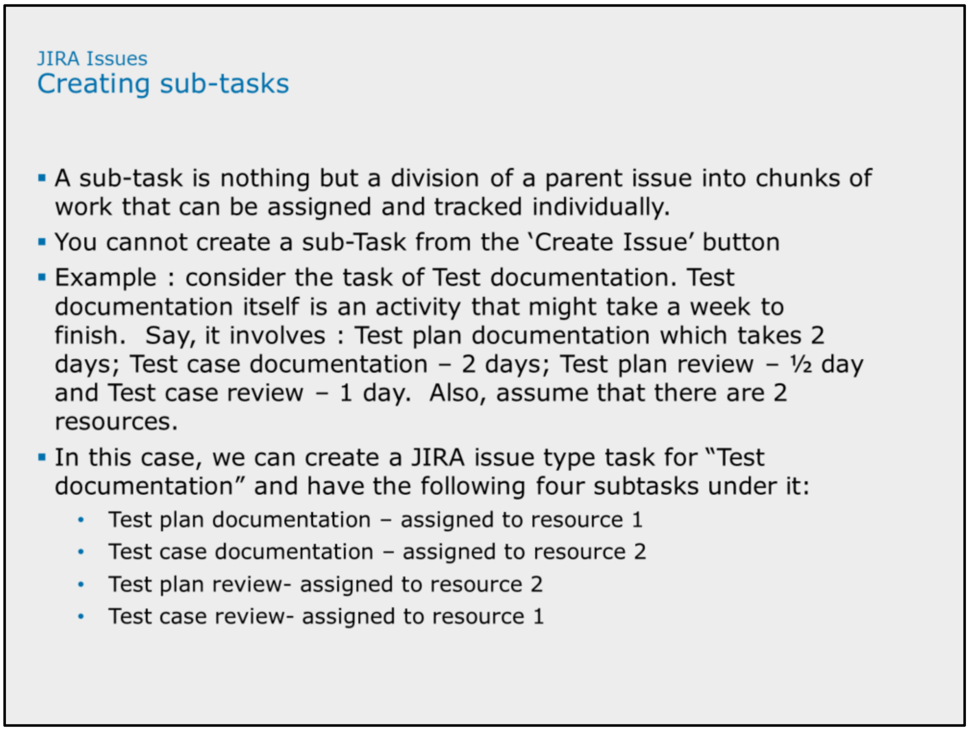
Move: JIRA issues can be moved across projects. However, the move from one project to another might mean a different target workflow, a different issue type, a new status etc. It is therefore advisable to analyze thoroughly how the move is going to affect the issue before going ahead with this.

Link: This is a very versatile feature of JIRA that allows you can logically link issues with one another and establish relationships/dependencies.

An example situation where this can be used in QA projects is when a certain defect prevents you from working a certain requirement. You can use this option to show the dependency.

Clone: As the name implies you can create a duplicate for an issue.

Instructor Notes:



JIRA Issues

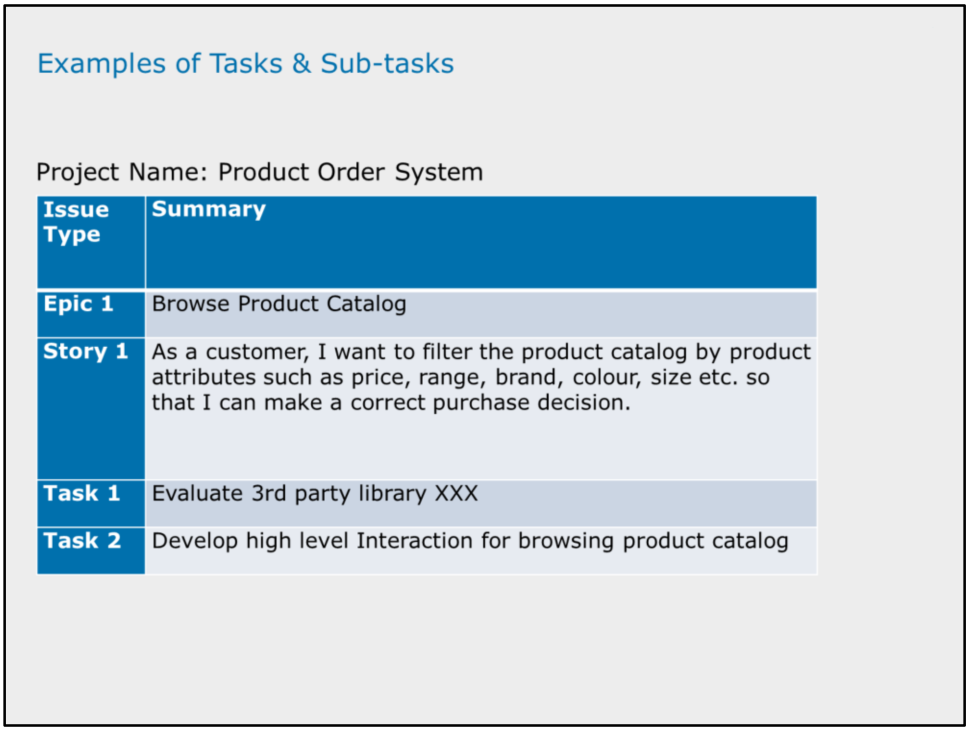
Creating sub-tasks

- A sub-task is nothing but a division of a parent issue into chunks of work that can be assigned and tracked individually.
- You cannot create a sub-Task from the 'Create Issue' button
- Example : consider the task of Test documentation. Test documentation itself is an activity that might take a week to finish. Say, it involves : Test plan documentation which takes 2 days; Test case documentation – 2 days; Test plan review – ½ day and Test case review – 1 day. Also, assume that there are 2 resources.
- In this case, we can create a JIRA issue type task for "Test documentation" and have the following four subtasks under it:
  - Test plan documentation – assigned to resource 1
  - Test case documentation – assigned to resource 2
  - Test plan review- assigned to resource 2
  - Test case review- assigned to resource 1

Advantage of sub-tasks : By doing so, it is easier to have a better insight into the progress task wise and resource wise by breaking a sizeable parent task into sub-tasks.

Note : It should not be confused that an issue type of "Task" only contains "sub-tasks". An issue of any type can have sub-tasks.
<u>Example 2</u>: An example related to bug could be – if a bug is encountered and needs a code change to fix it, the developer can use a sub-task to track this code-fix that needs to take place. Here, the code-fix (of type sub-task) becomes a sub-task under the bug found (of type Bug).
You can create sub-tasks by following one of the two methods:
Create a sub-task to an issue, or Convert an issue to a sub-task and choose a parent

Instructor Notes:

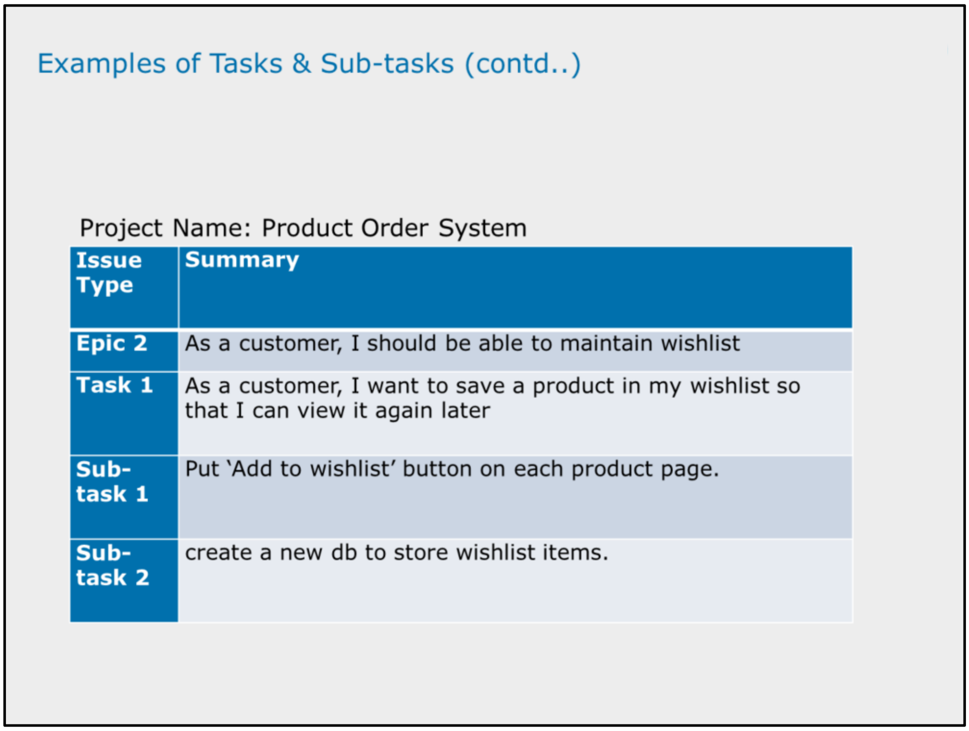


Examples of Tasks & Sub-tasks

Project Name: Product Order System

Issue Type	Summary
Epic 1	Browse Product Catalog
Story 1	As a customer, I want to filter the product catalog by product attributes such as price, range, brand, colour, size etc. so that I can make a correct purchase decision.
Task 1	Evaluate 3rd party library XXX
Task 2	Develop high level Interaction for browsing product catalog

Instructor Notes:

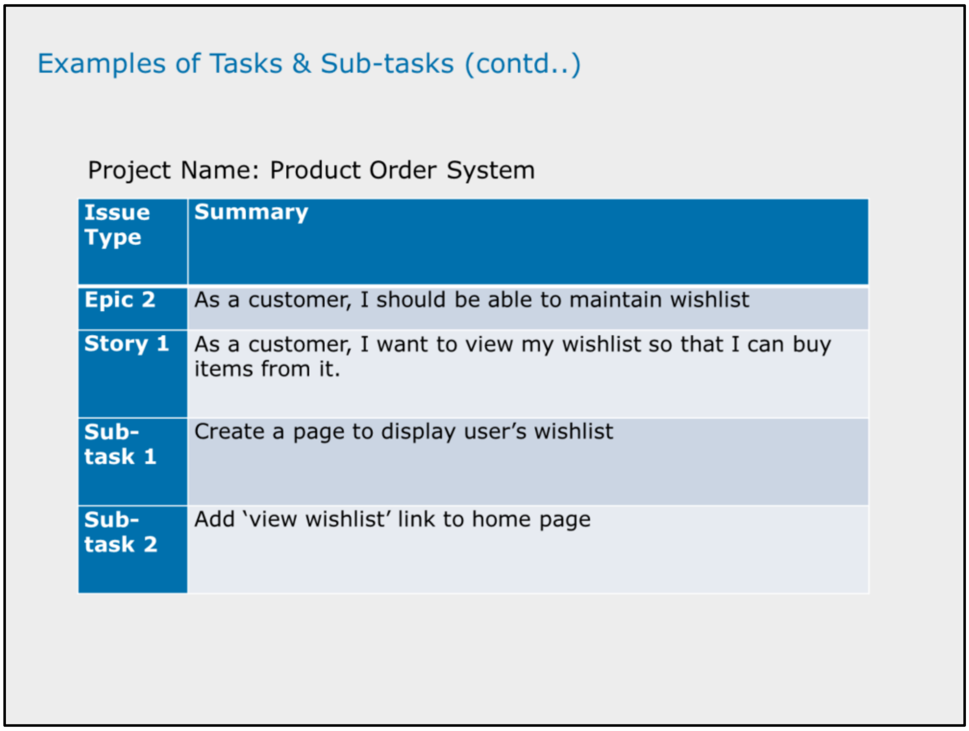


Examples of Tasks & Sub-tasks (contd..)

Project Name: Product Order System

Issue Type	Summary
Epic 2	As a customer, I should be able to maintain wishlist
Task 1	As a customer, I want to save a product in my wishlist so that I can view it again later
Sub-task 1	Put 'Add to wishlist' button on each product page.
Sub-task 2	create a new db to store wishlist items.

Instructor Notes:

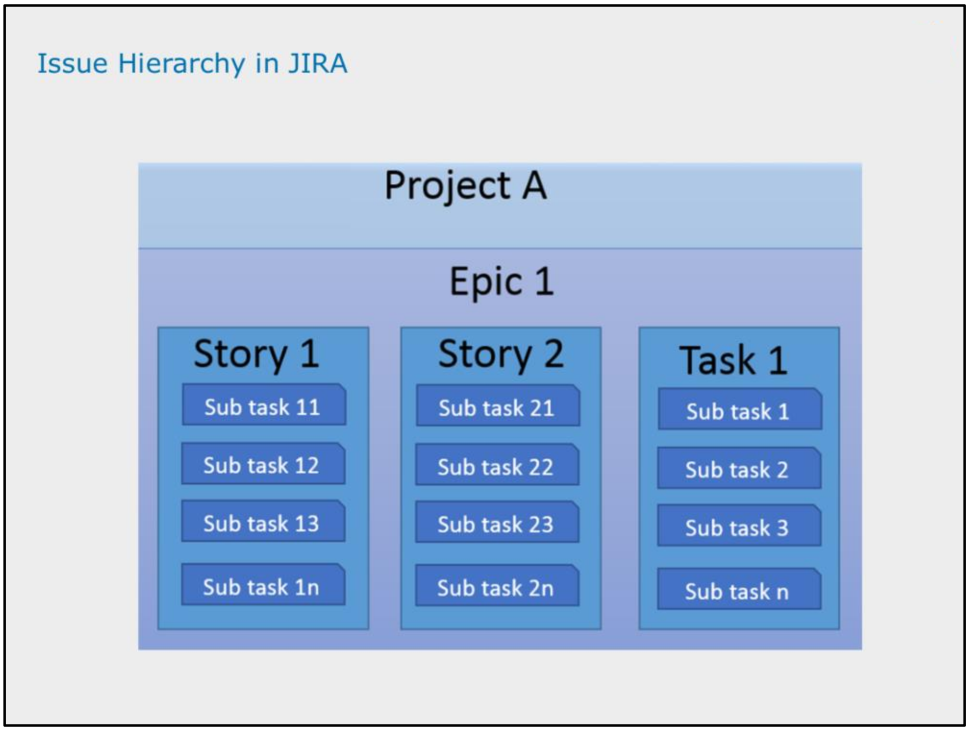


Examples of Tasks & Sub-tasks (contd..)

Project Name: Product Order System

Issue Type	Summary
Epic 2	As a customer, I should be able to maintain wishlist
Story 1	As a customer, I want to view my wishlist so that I can buy items from it.
Sub-task 1	Create a page to display user's wishlist
Sub-task 2	Add 'view wishlist' link to home page

Instructor Notes:



Issue Hierarchy in JIRA

Project A

Epic 1

Story 1

Sub task 11

Sub task 12

Sub task 13

Sub task 1n

Story 2

Sub task 21

Sub task 22

Sub task 23

Sub task 2n

Task 1

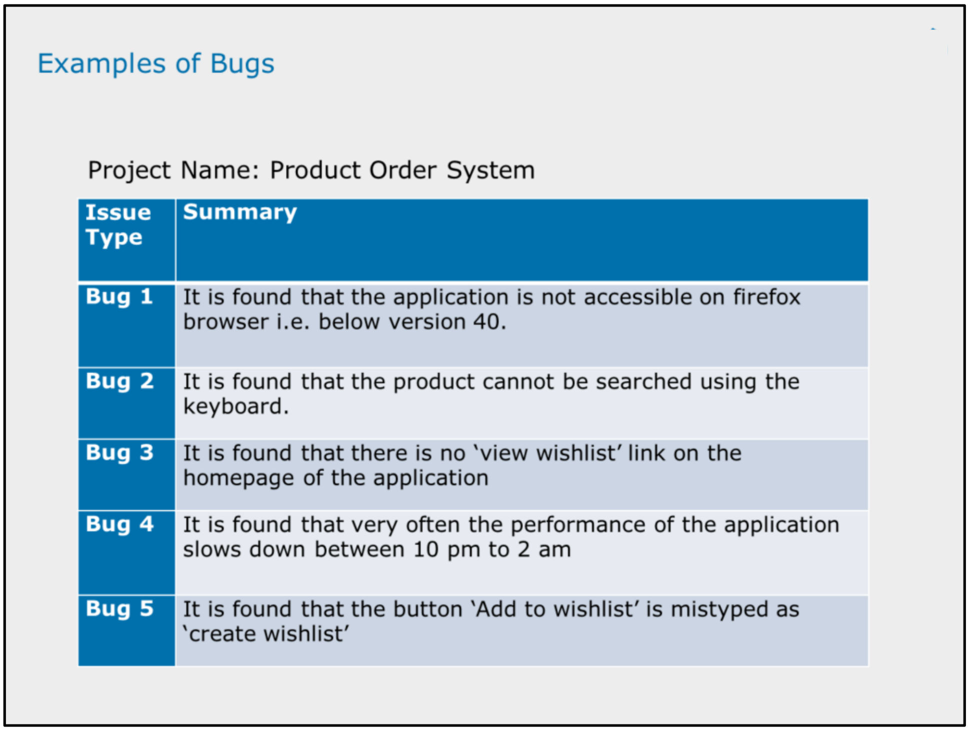
Sub task 1

Sub task 2

Sub task 3

Sub task n

Instructor Notes:



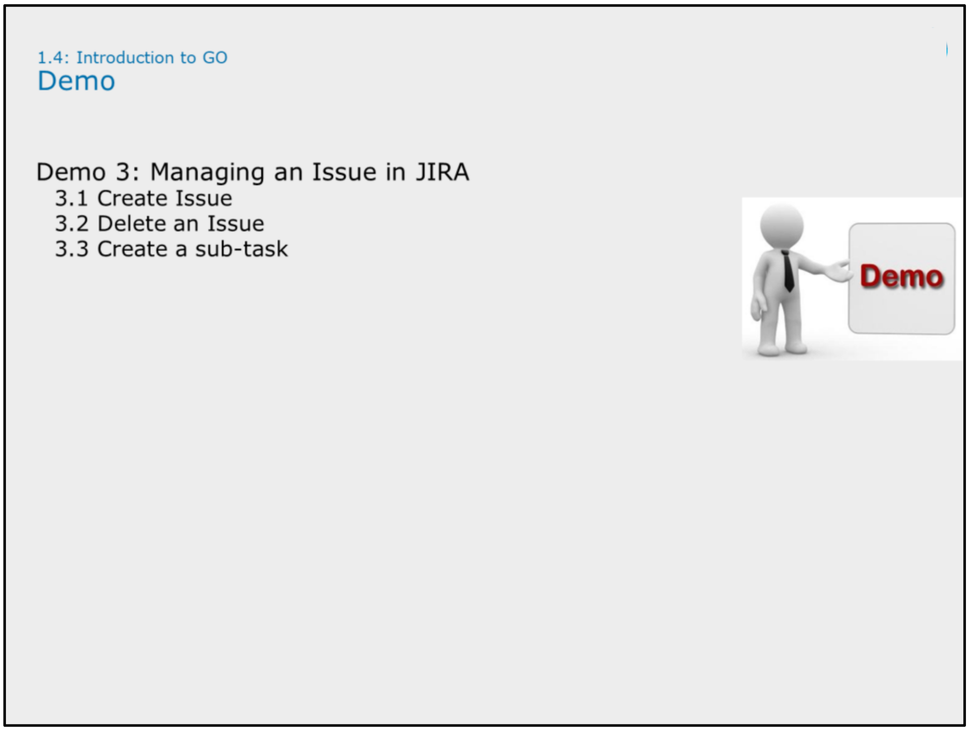
Examples of Bugs

Project Name: Product Order System

Issue Type	Summary
Bug 1	It is found that the application is not accessible on firefox browser i.e. below version 40.
Bug 2	It is found that the product cannot be searched using the keyboard.
Bug 3	It is found that there is no 'view wishlist' link on the homepage of the application
Bug 4	It is found that very often the performance of the application slows down between 10 pm to 2 am
Bug 5	It is found that the button 'Add to wishlist' is mistyped as 'create wishlist'

Instructor Notes:

Add instructor notes here.



1.4: Introduction to GO

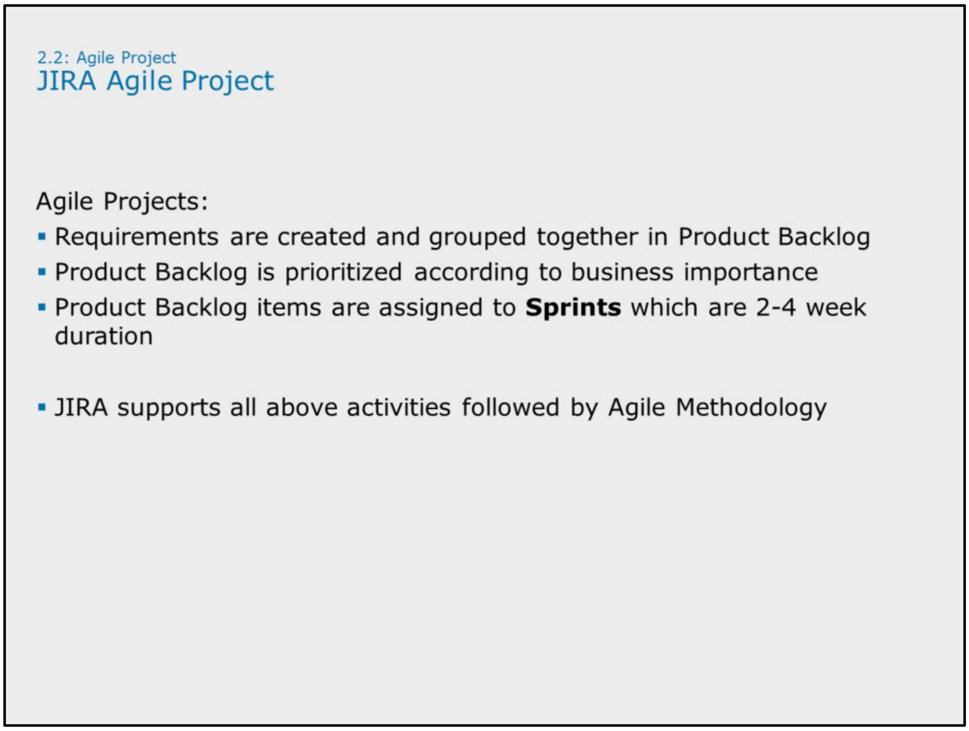
Demo

Demo 3: Managing an Issue in JIRA
  3.1 Create Issue
  3.2 Delete an Issue
  3.3 Create a sub-task

Add the notes here.

Instructor Notes:

Add instructor notes here.



2.2: Agile Project

JIRA Agile Project

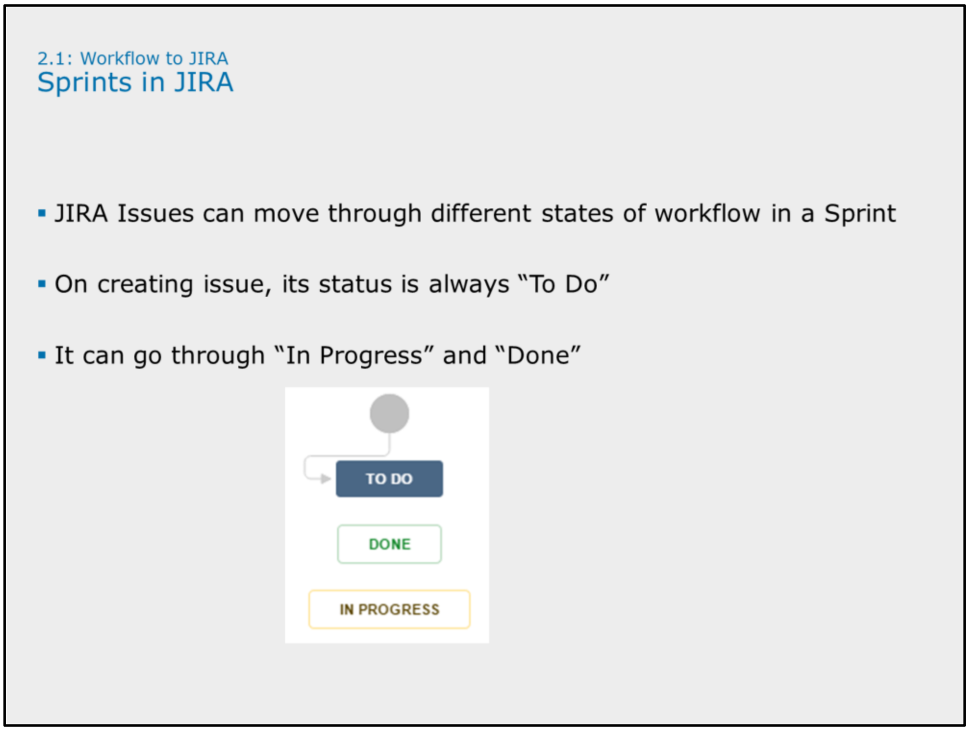
Agile Projects:

- Requirements are created and grouped together in Product Backlog
- Product Backlog is prioritized according to business importance
- Product Backlog items are assigned to **Sprints** which are 2-4 week duration

- JIRA supports all above activities followed by Agile Methodology

Instructor Notes:

Add instructor notes here.



2.1: Workflow to JIRA

Sprints in JIRA

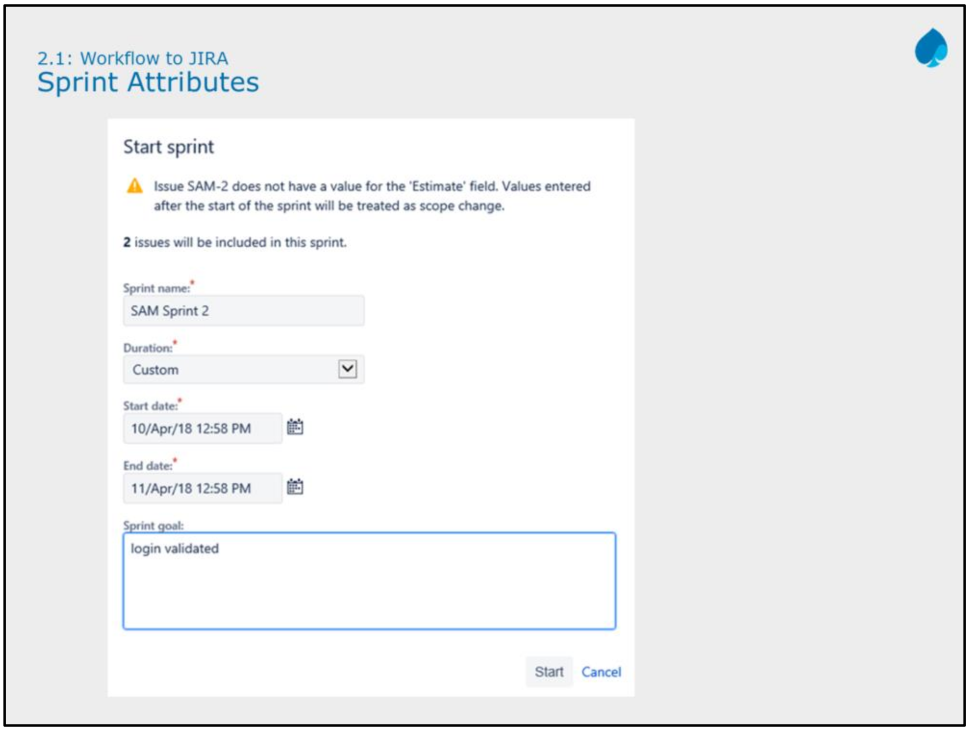
- JIRA Issues can move through different states of workflow in a Sprint
- On creating issue, its status is always "To Do"
- It can go through "In Progress" and "Done"

Workflow will depend on the project type and the workflow assigned to the project while creation.

Only Admin can choose this and once chosen, it cannot be changed and all the issues under the project will go through the same workflow.

Instructor Notes:

Add instructor notes here.



2.1: Workflow to JIRA

Sprint Attributes

Start sprint

⚠ Issue SAM-2 does not have a value for the 'Estimate' field. Values entered after the start of the sprint will be treated as scope change.

2 issues will be included in this sprint.

Sprint name:* SAM Sprint 2

Duration:* Custom

Start date:* 10/Apr/18 12:58 PM

End date:* 11/Apr/18 12:58 PM

Sprint goal: login validated

<u>To start a sprint :</u>

1. Go to the **Backlog** of your Scrum project.
2. Find the sprint that you want to start and click **Start sprint.**
3. Update the **Sprint name**, select the **Start date**, **End date** and **Duration** for the sprint. *Note, the default duration of a sprint is two weeks. These fields are mandatory to start a sprint.*
4. Add a **Sprint goal** if desired since it is optional.

Instructor Notes:

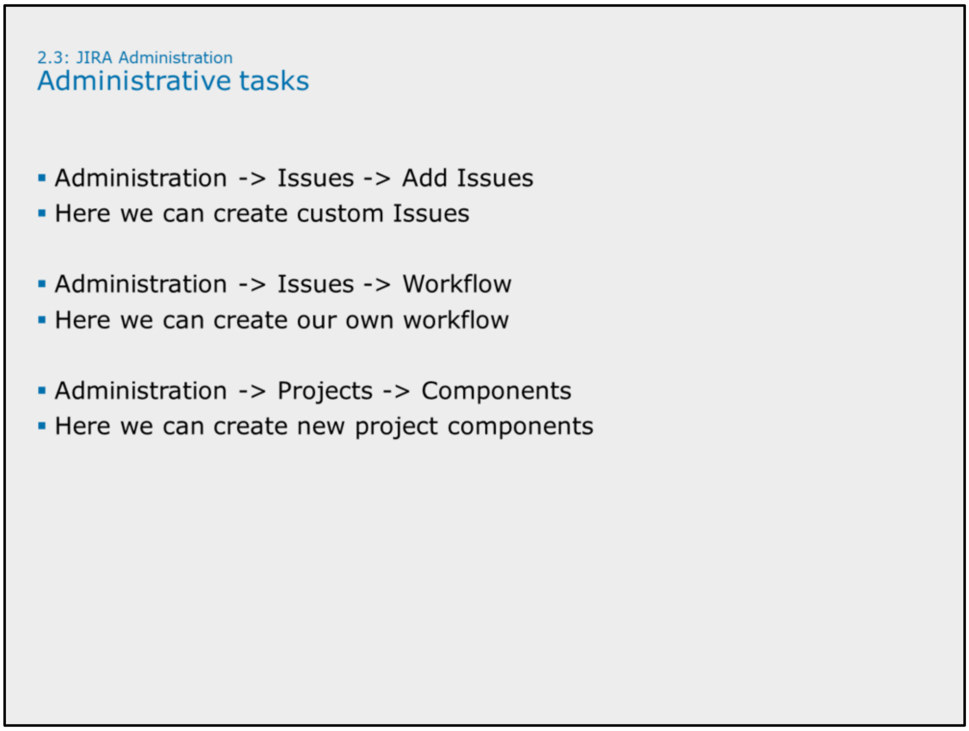
Add instructor notes here.

Demo

Demo 4: Managing Sprints in JIRA
- 4.1 Create Sprint
- 4.2 Start Sprint

Add the notes here.

Instructor Notes:



2.3: JIRA Administration

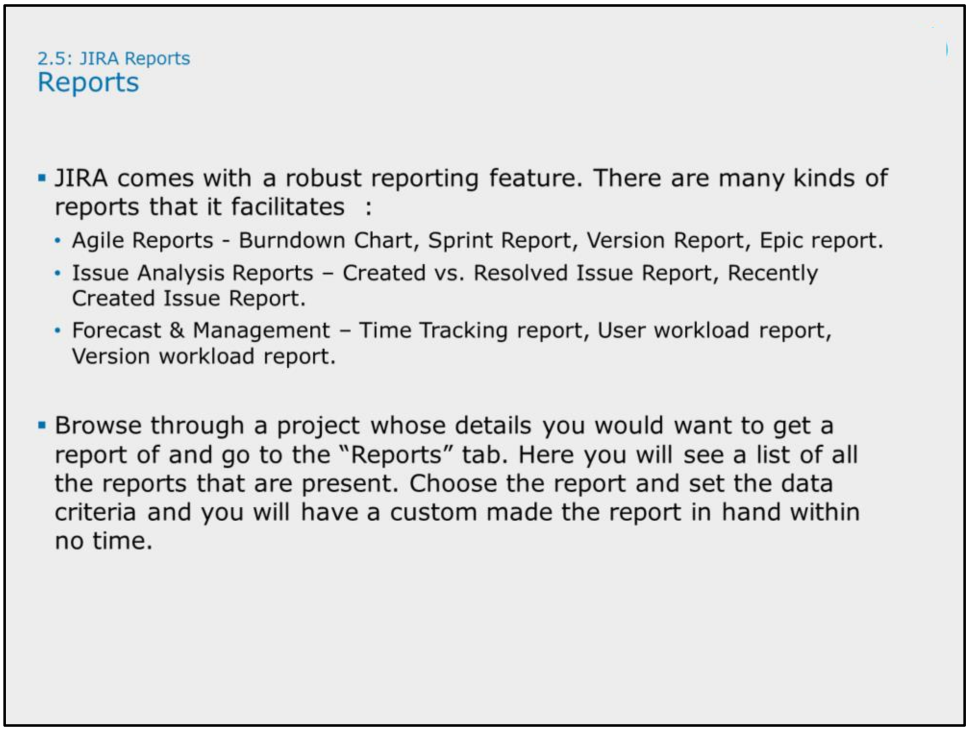
Administrative tasks

- Administration -> Issues -> Add Issues
- Here we can create custom Issues

- Administration -> Issues -> Workflow
- Here we can create our own workflow

- Administration -> Projects -> Components
- Here we can create new project components

Instructor Notes:



2.5: JIRA Reports

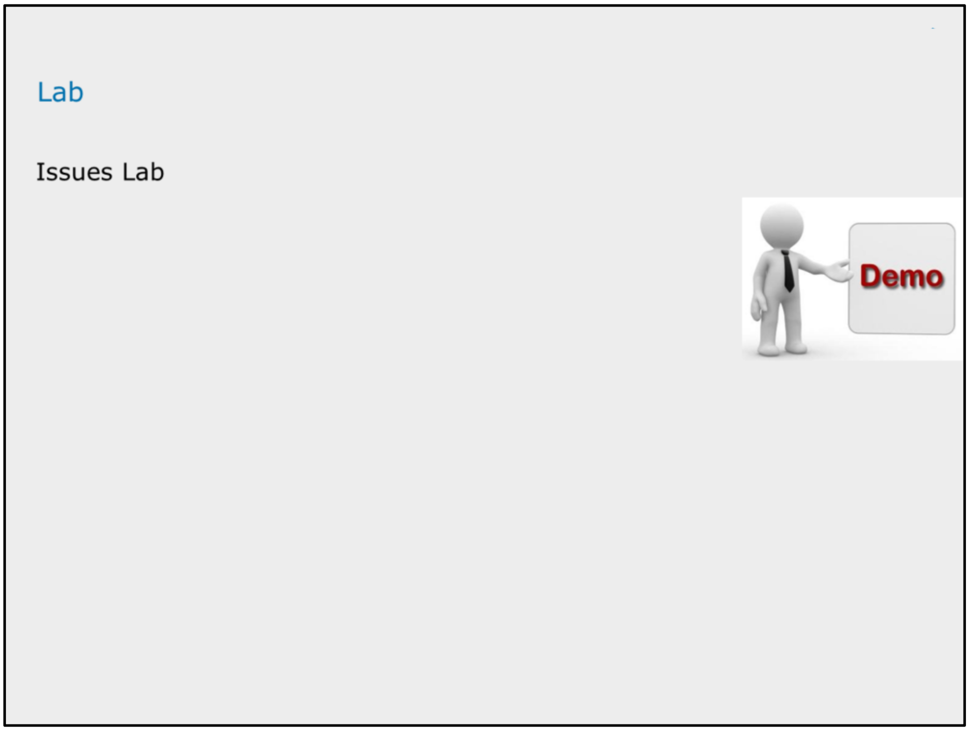
Reports

- JIRA comes with a robust reporting feature. There are many kinds of reports that it facilitates :
  - Agile Reports - Burndown Chart, Sprint Report, Version Report, Epic report.
  - Issue Analysis Reports – Created vs. Resolved Issue Report, Recently Created Issue Report.
  - Forecast & Management – Time Tracking report, User workload report, Version workload report.
- Browse through a project whose details you would want to get a report of and go to the "Reports" tab. Here you will see a list of all the reports that are present. Choose the report and set the data criteria and you will have a custom made the report in hand within no time.

- **Burndown Chart :** Track the total work remaining, also whether sprint is achieving the project goal or not.
- **Sprint Report:** Track the work completed or pushed back to the backlog in each sprint.
- **Velocity Chart :** Track the amount of work completed from sprint to sprint.
- **Version Report :** Track the projected release date for a version.
- **Epic Report :** Shows the progress towards completing an epic over a given time.
- **Created vs. Resolved Issue Report :** Display the number of issues created vs the number of issues resolved in given period.
- **Recently Created Issue Report :** Shows the number of issues created over a time-period for a project and how many of those were resolved.

Instructor Notes:

Add instructor notes here.



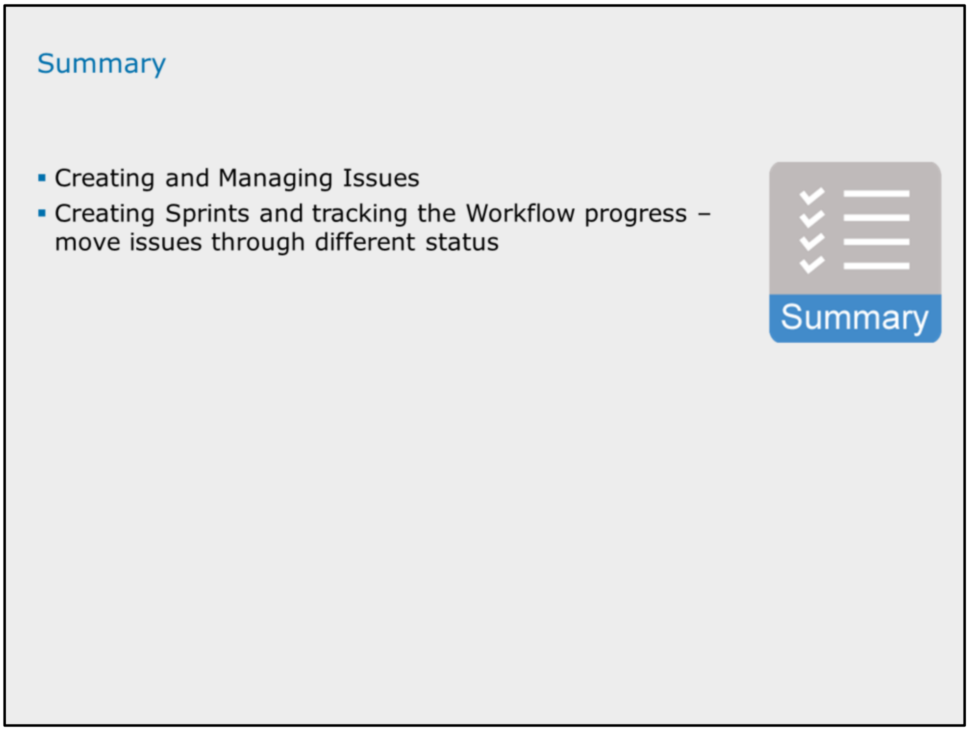
Lab

Issues Lab

Add the notes here.

Instructor Notes:

Add instructor notes here.



Summary

- Creating and Managing Issues
- Creating Sprints and tracking the Workflow progress – move issues through different status

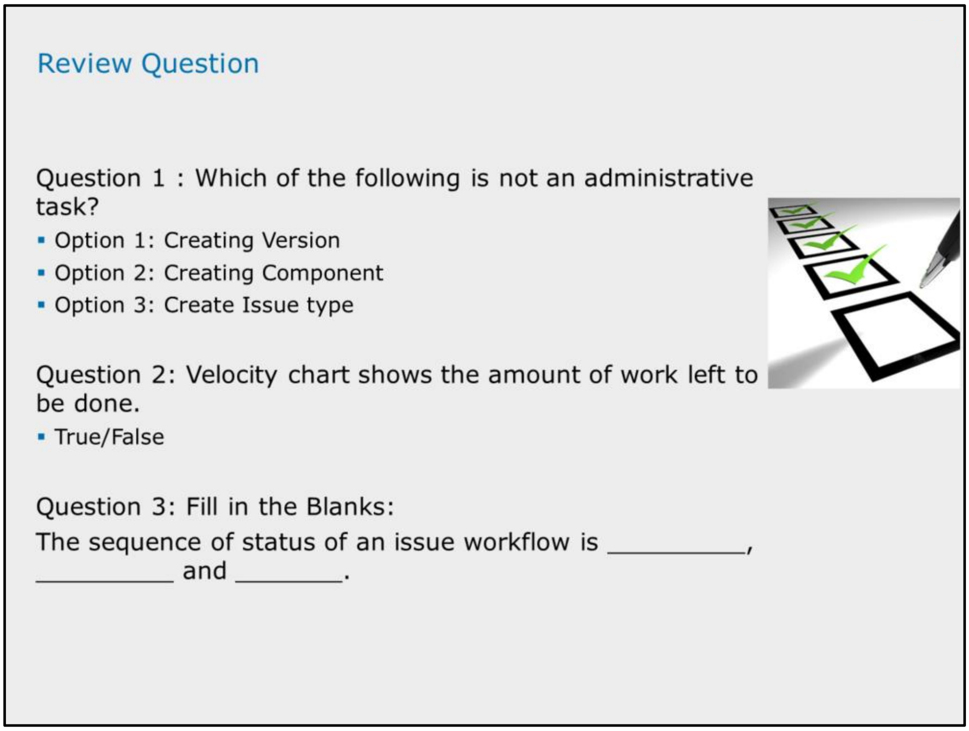
Add the notes here.

Instructor Notes:

Answers:

Question1. All of the above

Question 2: main()



Review Question

Question 1 : Which of the following is not an administrative task?

- Option 1: Creating Version
- Option 2: Creating Component
- Option 3: Create Issue type

Question 2: Velocity chart shows the amount of work left to be done.

- True/False

Question 3: Fill in the Blanks:

The sequence of status of an issue workflow is __________, __________ and ________.

Add the notes here.