Agile Software Project Management using JIRA

Avinash

Introduction:

What is Agile software development?

Agile software development methodologies are all about delivering small pieces of working software quickly to improve customer satisfaction (, while building software iteratively). These methodologies use adaptive approaches and teamwork to focus on continuous improvement. [1]

The four principles of Agile development [2] are:

1. Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation.
3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan.

Agile software development is a philosophy, whereas *scrum* is a framework that helps fulfill the philosophy or idea of Agile development. Jira is a software that helps implementing a scrum.

What is scrum project?

Scrum (or scrum project) is one of the most popular frameworks for implementing Agile. With scrum, the product is built in a series of fixed-length iterations called sprints that give teams a framework for shipping on a regular cadence (or time interval) [3]

What is an issue?

In Jira Software, we call work items like user stories (which define requirements), tasks, and bugs, "issues". Note that JIRA started out as an bug or issue tracking software, and therefore, to this day, user stories, tasks and bugs are all called issues.

But what are user stories, tasks and bugs?

User stories are used to **describe work items in a non-technical language** and from a user's perspective. As a {type of user}, I want {goal} so that I {receive benefit}. Once you've created a few user stories, you can start prioritizing them in the *backlog,* a certain section of JIRA software (Remember Product backlog and sprint backlog from class?) Over project lifetime, you can modify user stories and add new user stories [3].

Tasks are any work items that are to be performed by various team members of a project

Bugs are functional anomalies that are logged down so that they can be tracked to completion – whether the bug has been fixed and tested for non-existence.

Finally, what is a Sprint?

In Scrum, teams forecast to complete a set of user stories or other work items during a fixed time duration, known as a sprint. Generally speaking, sprints are one, two, or four weeks long. It is generally recommended that sprints be two weeks long, which gives enough time to get some work done, and yet quick enough to get feedback on where the project is headed and how well it is going.

The point of this lab is to introduce you to one of the most popular Software project management tools, that is, JIRA. JIRA is a standard used by many players in the industry, and is ideal for learning because it is free (haha?). Okay, so JIRA is primarily designed for Agile development, but can be tailored for waterfall based (or plan based) software project Management. Waterfall based approaches require simpler management tool support when compared to Agile (why?), and you will notice that many Project management tools are tailored for Agile. JIRA is no exception. However, you can use a little imagination and use JIRA for conventional project management (plan based, waterfall based). Other tools include MSProject which is popular (but is not free), OpenProject offers a 14 day trial and is similar to MSProject (link given in this assignment, you can check it out, worth 2 marks of the assignment, if you need more incentive). Other popular Software management tools include Trello and Asana, but it is beyond the scope of this lab to show multiple tools. Readers are urged to explore project management tools at their own time.

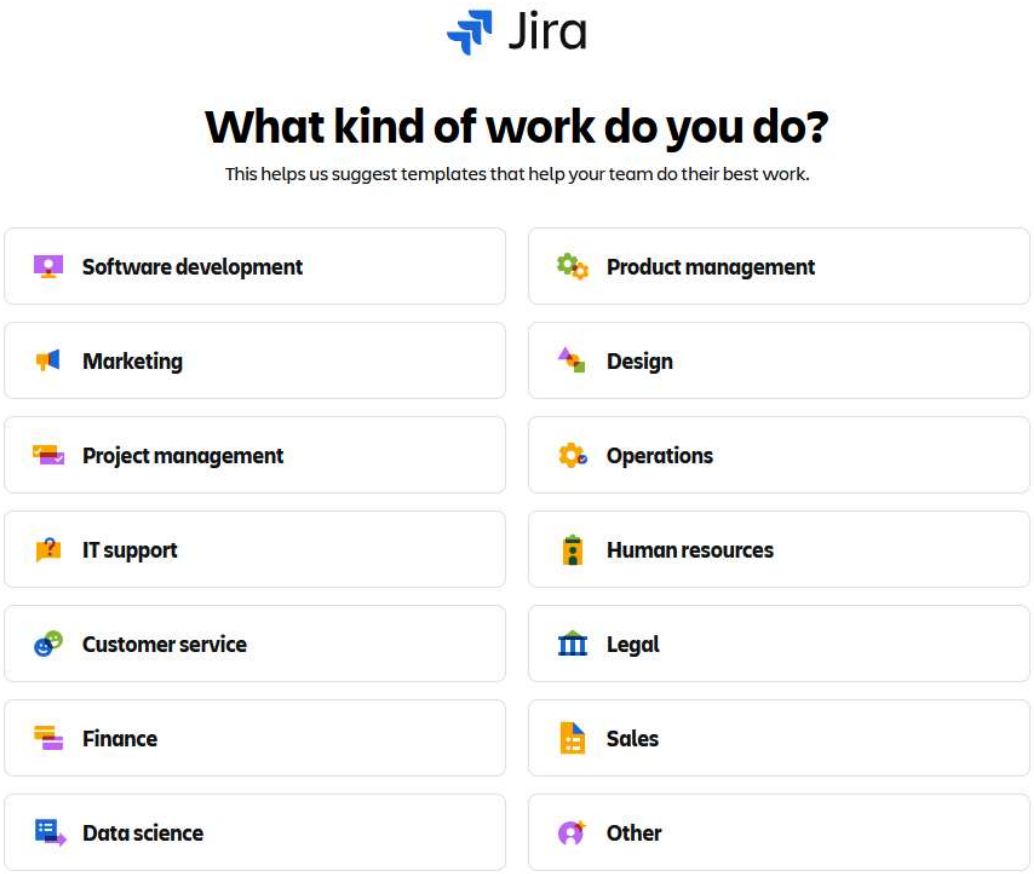
Coming back to JIRA, this tutorial might use Agile project development as its guide (because industry standard), but remember, you can use JIRA in your course projects – even if you follow a conventional waterfall like plan-based approach.

Step by step procedure to run Agile project development using a scrum project

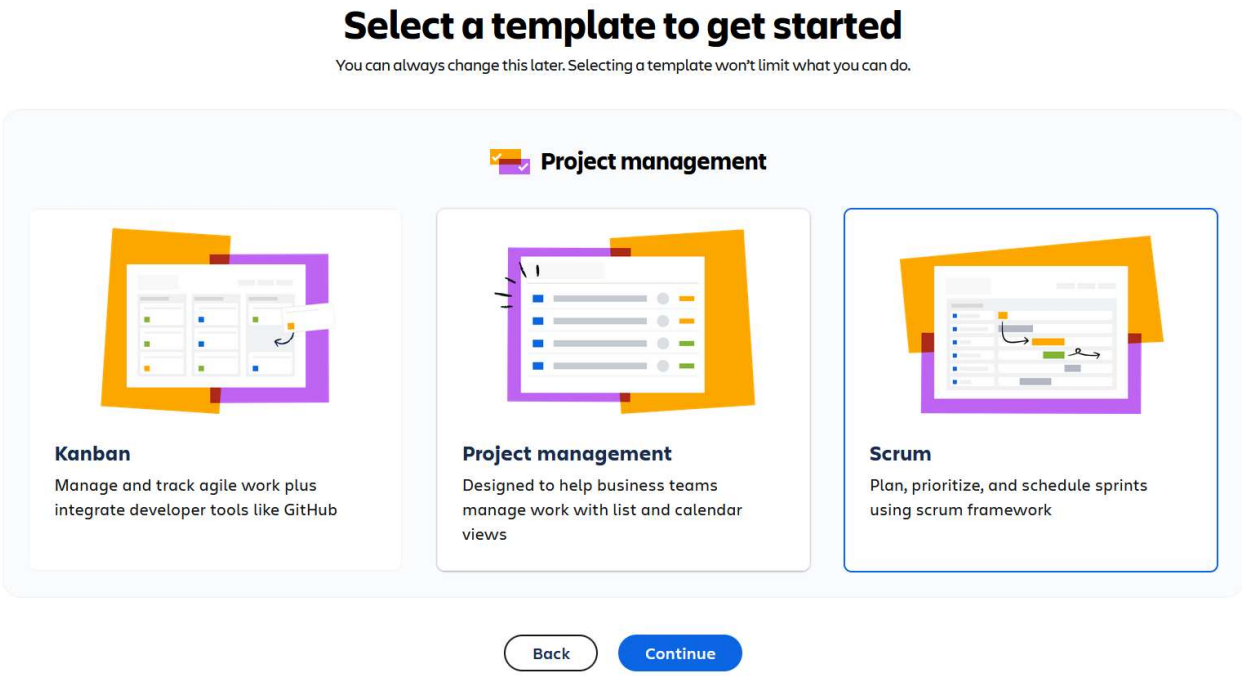
Prerequisite: Jira software account. Get a free Jira account here
Getting a Jira account:

The steps to get an account on JIRA are straightforward. You may use your existing Google account to sign up. Once you signup, you will be suggested a name for your domain space (just a fancy name for your JIRA workspace – using another fancy name to describe a fancyname), keep it as it is, or change as per your need. (NOTE: Set password to something you are comfortable sharing / SE25AICSxxx@MU where xxx is your roll number in 3 digits)

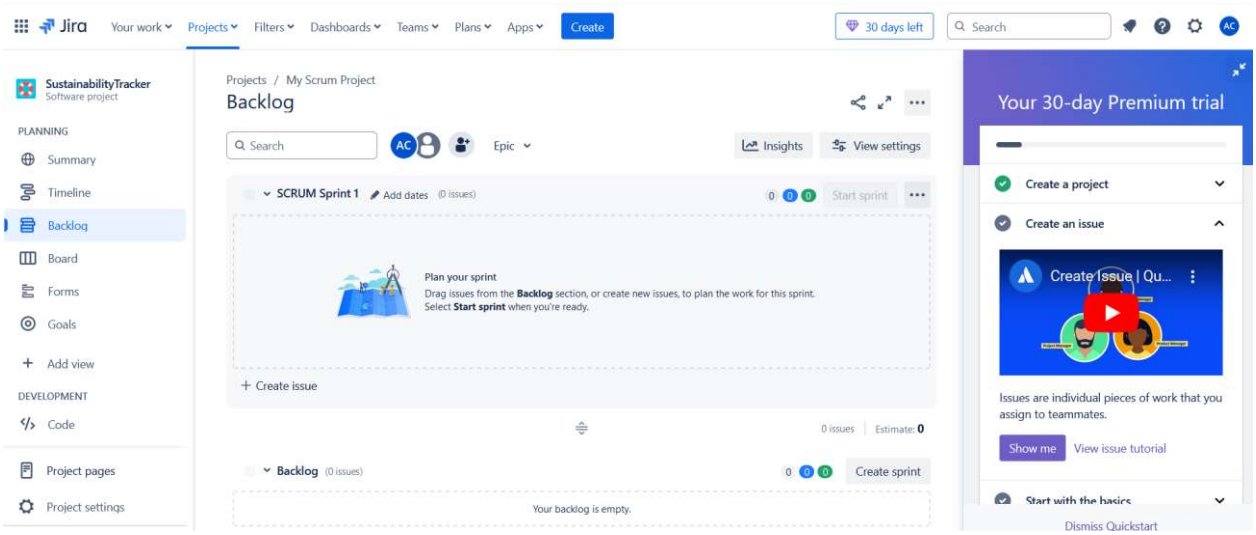
Upon signup, you will encounter the following screen:



You may choose either Software Development or Project Management. I suggest you choose 'Project Management' since it will provide additional features that aid in project management (eventual aim of this assignment). Choose 'Scrum' project (see screenshot below):



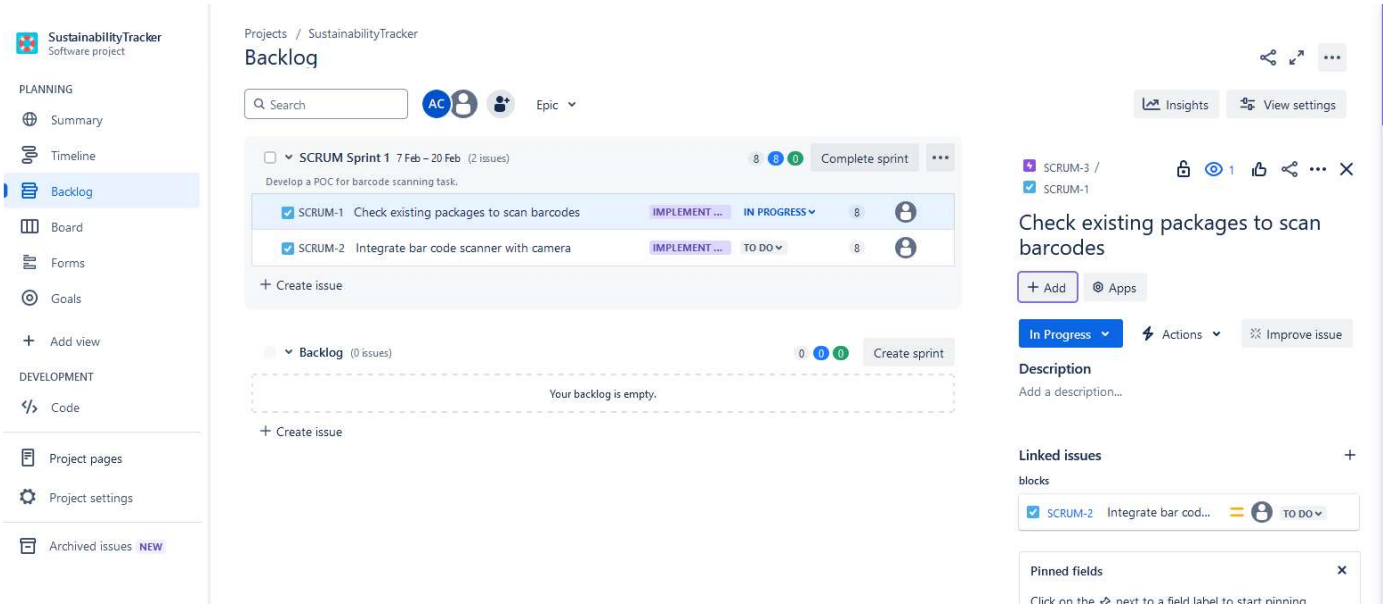
Give a name for your scrum project, and hit 'get Started'. This will take you to the landing page of the project (by landing page, I mean the first page you are redirected to post project creation). It is the backlog page (remember product and sprint backlog from class?)



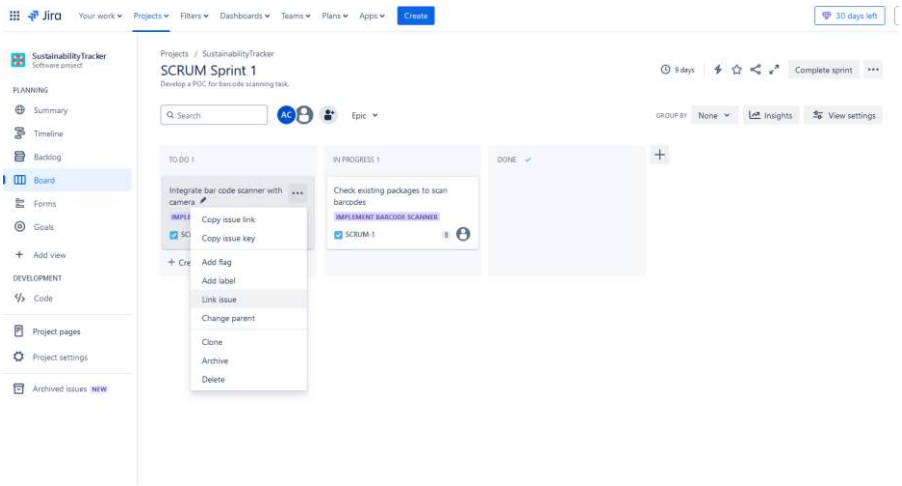
Notice the 'Quickstart' to the right of the screen (the YouTube thumbnail)? That is to help you get started with using JIRA (creating issues, creating epics, sprint dates, etc. - basically helps you, is a user manual to your JIRA account). You can refer the 'Quickstart' videos to complete your lab assignment.

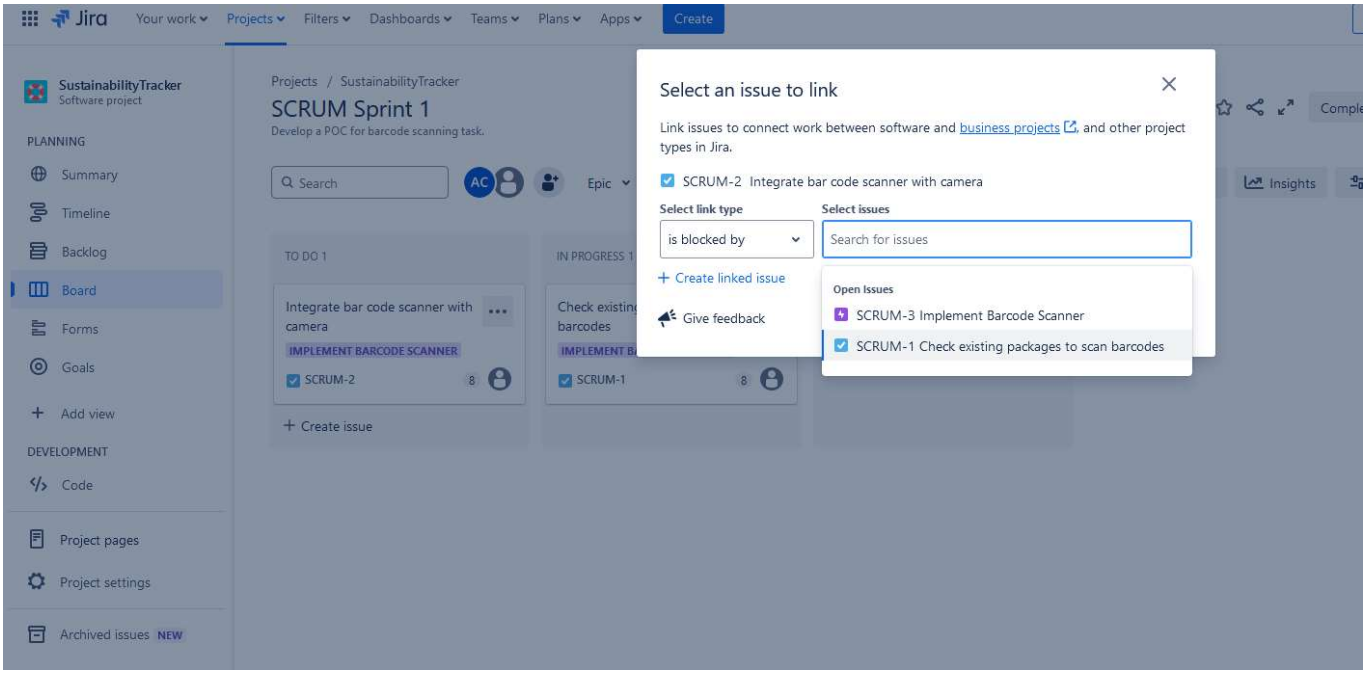
You can either continue reading this document to finish your assignment related tasks, or you can play around / explore the scrum project at the Atlassian website that runs Jira and complete the assignment related tasks. The following image shows your landing page once the project has been created.

This is what I did with my scrum project – not a lot but yeah (created issues/ epics, sprint, etc.). At this point, I suggest you explore around JIRA and create a Sprint of two weeks, which consists of an EPIC, which in turn consists of two issues (work items), and assign 'story points' to the two issues

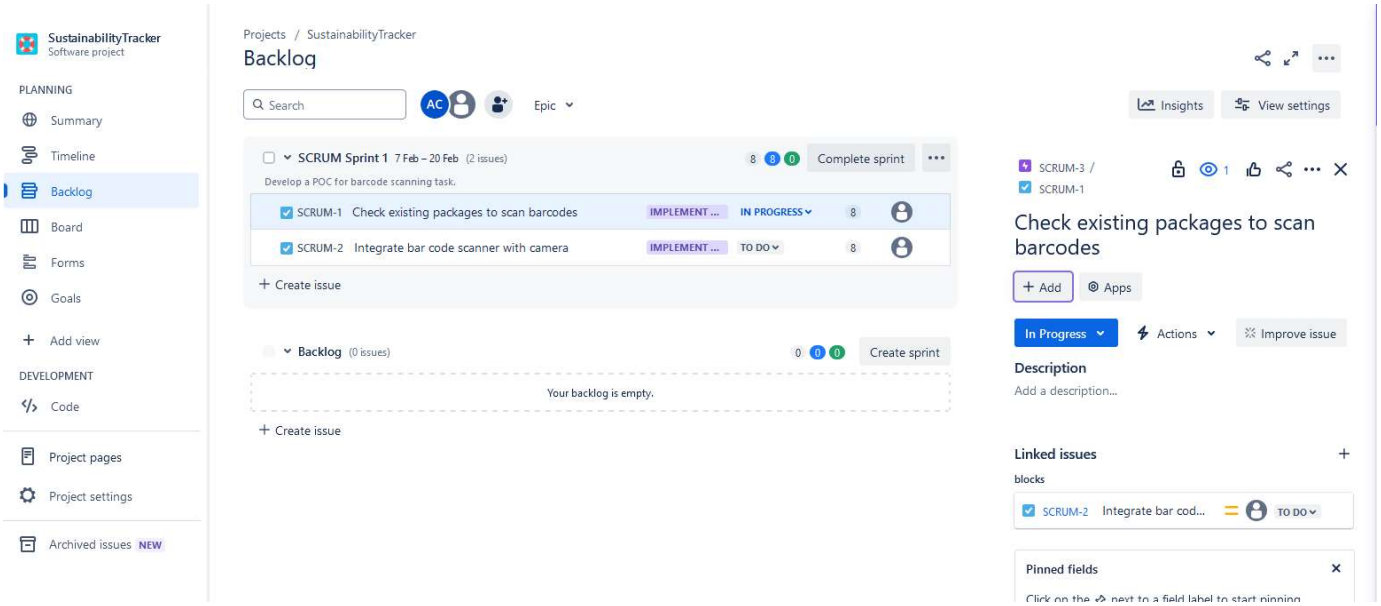


To link issues with one another (An example of linking is where one issue or task can commence only when some other issue/ task it is dependent on is complete), go to 'Board', select an 'issue', and select 'link' to link it to an other issue (see following screenshots)

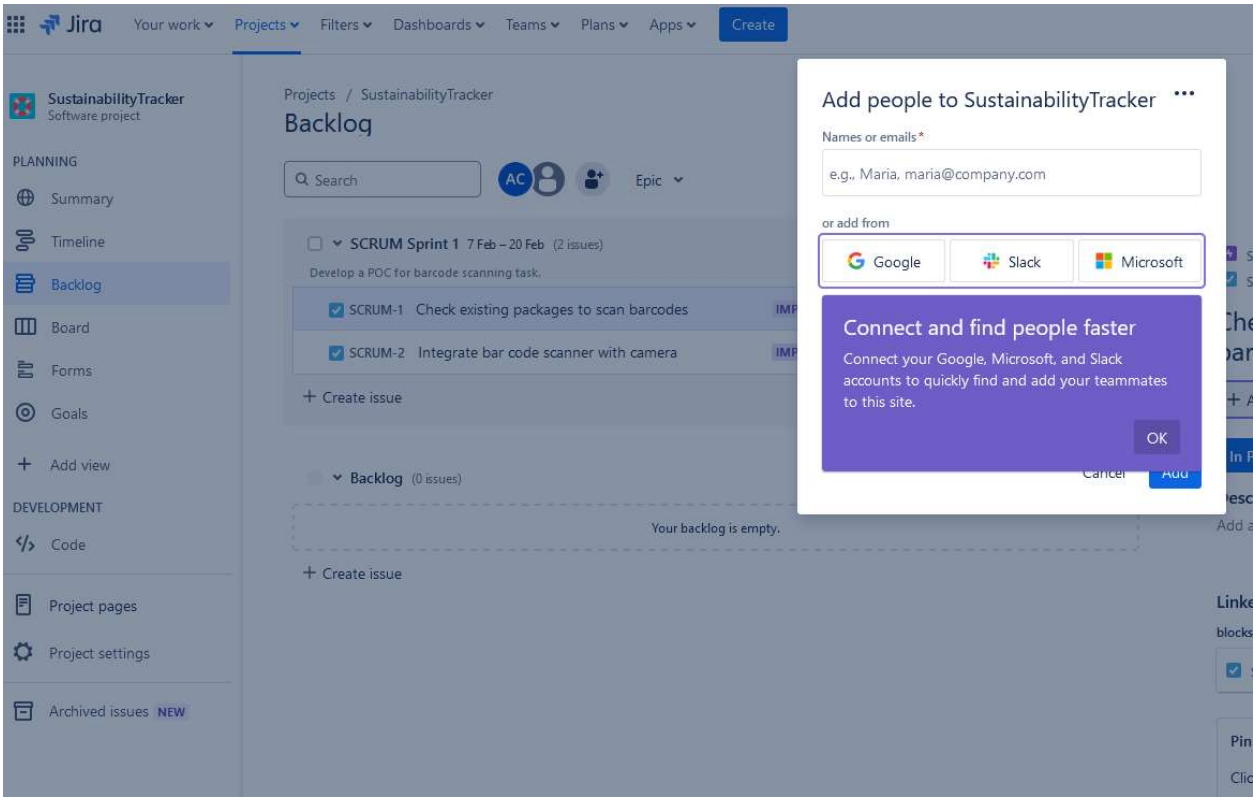




You can see the linked issues in your project timeline

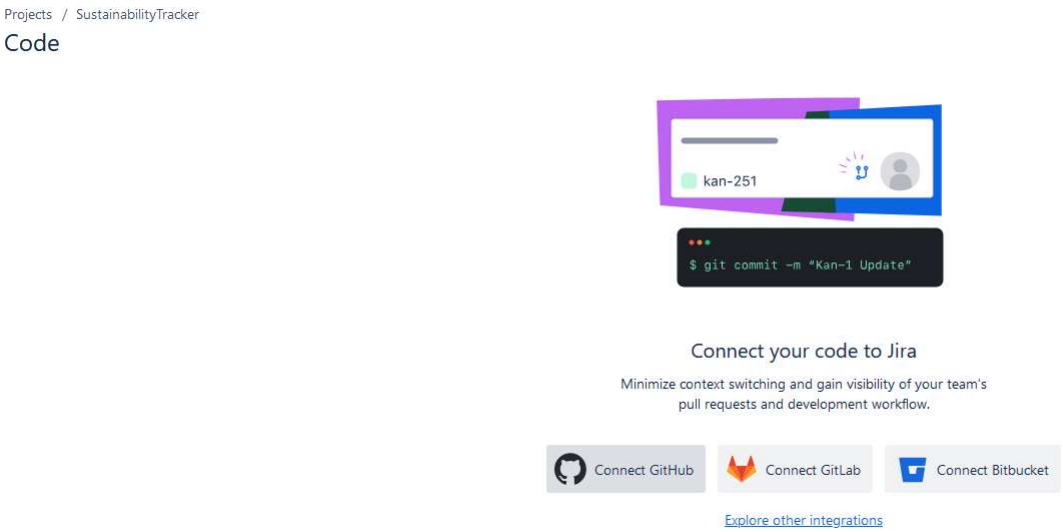


Add people to your project by clicking the gray person with a + sign icon, later you can assign issues to the people added



<u>Link GitHub to your scrum project:</u>

Goto ‘Code’ from the left panel, select ‘Connect GitHub’ at the Atlassian marketplace > Install > Configure App > Select ‘GitHub Cloud’ > Provide Github credentials (if not already logged in GitHub in your browser) > Select ‘Connect’



If everything works out well, you will be able to see something like this: