Software Testing

Quality Assurance

Quality Assurance

- The process or ensuring the software meets requirements and is of high quality
- Overlooked in the early days
- Now considered a vital part of the development process

Agile Projects

- Traditional project management was driven by due dates.
- The project was laid out in terms of deliverables and dates assigned to each one.
- This was typically done using something like a Gantt chart
- There were two problems with this approach.
  - it was inflexible so that any unexpected event would require a complete rescheduling
  - if a deliverable turned out to be more complicated than initially thought, there was no way to easily break it down into smaller units of work.
- As a result of these shortcomings, this type of project management was usually doomed to failure.

Agile Projects

- Much more flexible
- Breaks projects into
  - Themes or initiatives,
  - Epics or projects
  - Tasks or issues

Themes

- one of the major goals of the project.
- usually a long term objective that is a major and significant component of the project.
- can also be viewed as a major strategic business objective.
- For example, if you decide you want to break into the project management software market then this would be the theme of your project.

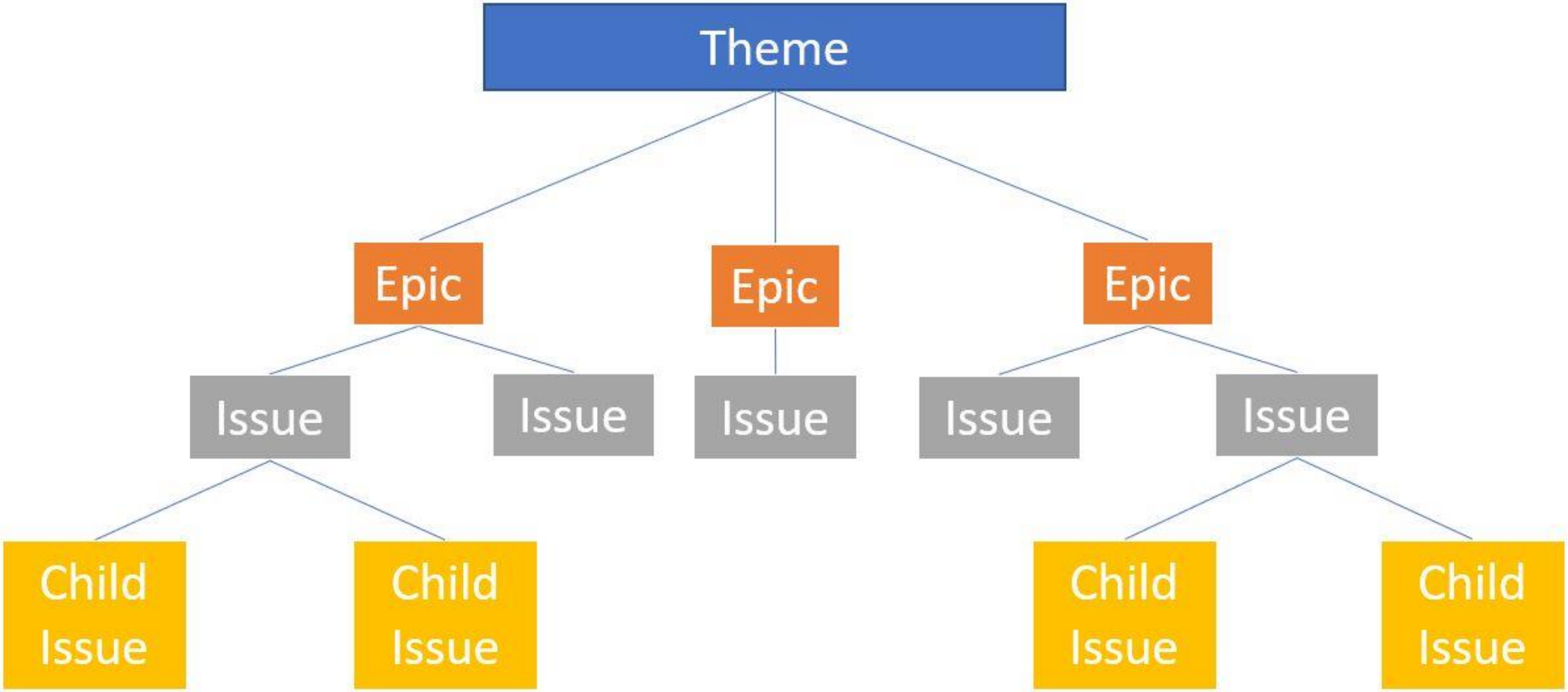
Epics

- a larger body of work that constitutes one of the main components of a theme
- Epics are measurable – you can determine when complete
- For our example all penetrating the project management software market, we might layout the following two epics:
  - New features -- an effort to develop new features for our project management software,
  - Enhancement -- and effort to enhance the current features of our software to make it suitable for use by project managers.

Issues

- one of the tasks that needs to be completed as part of an epic.
- a small enough task that it can be performed in a few days or a week.
- for the new features epic of our attempt to penetrate the project management software market, we could break it down into 3 issues:
  - Researching the project management tool market,
  - Designing new features that we want to implement,
  - Developing the new features that we are going to implement.

Agile Project Breakdown



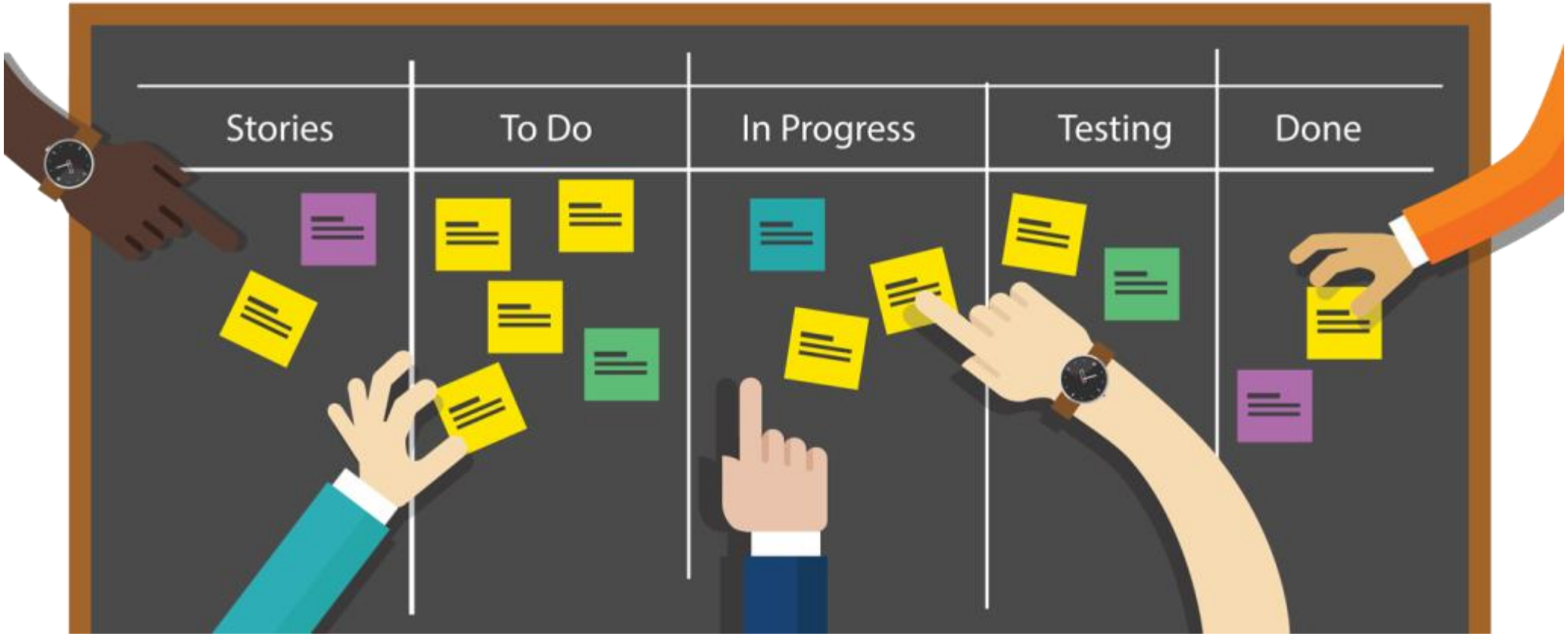
User Stories

- common to specify tasks as user stories
  - keep the developers informed of the importance of what they're working on
  - let them see how it fits into the big picture
- we now build units of functionality that are desired by the end user.
- they convey the users perspective to developer
- Eg
  - As a visitor to the project management website, I want to be able to filter the different projects based upon the project name as well as the personnel working on the project.

Every user story should be INVESTable

- **I**ndependent - the story should not depend on other stories and can be completed on its own,
- **N**egotiable - it should open a conversation with the customer to invite refinement and change,
- **V**aluable - it should provide significant value to the project,
- **E**stimable - it should be estimated to be the right size for an issue,
- **S**mall - it should be able to be completed in a few days,
- **T**estable - it should have clear acceptance criteria so tests can be writtten to verify it.

Kanban Boards



Software Test Plan

- one of the most fundamental documents for software testing.
- a project plan for the system testing.
- It is really the instruction manual for the testing process

Software Test Plan

- The test plan should include:-
  - A description of how the testing will be performed at a particular level,
  - The objectives of the testing which should describe what is being tested and what this will prove,
  - the scope of the testing which will determine what parts of the software are being tested as well as what parts are not being tested,
  - A schedule of when the tests will be conducted and the order of the tests,
  - A list of risks in the project and how they can be mitigated,
  - a list of the hardware and software resources necessary to conduct the testing,
  - A list of the roles and skills of the people required to do the testing.

Test Strategy

- a short, separate document which is often written before the test plan.
- define the major test objectives and to make sure that the test aligns with the organizational needs
- includes:
  - A description of what makes this project unique
  - A description of the SuccessFactors that are being tested,
  - Risks involved to the business the project the product, etc,
  - The roles and responsibilities of the people who will conduct the test,
  - A rough schedule of the test,
  - The level of the test for example component testing integration testing or system acceptance testing,
  - The type of testing to be performed whether it be functional testing, security testing, usability testing or some other type.

Test Plan Layout

- Introduction
  - a list of test objectives
  - the scope of the testing
  - a system overview
- Approach
  - a list of assumptions and constraints on the testing,
  - - how the test coverage will be determined,
  - - software components to be validated,
  - - business processes to be validated.
  - - Test Tools
  - - a list of the testing tools that will be required
  - - Test Type - list of the various types of testing to be done (functional, compliance, security, regression, etc.) and a high-level description of how the tests will be performed

Test Plan Layout

- Test Plan
  - the people on the test team and their responsibilities,
  - - schedule listing the major parts of the testing,
- Environment
  - a list and description of the hardware and software components needed for the testing,
  - - any special training needed for the testing team.
- Features to be tested
  - a list of the features to be tested.
- Features not tested.
  - a list of functionality that will not be tested

Test Plan Layout

- Testing procedure
  - a general overview of the testing process
  - the order in which the tests will be conducted,
  - - a general description of pass/fail criteria,
- Defect Management
  - how defects will be reported and managed
- Risks
  - list the risks to the project due to defects discovered during testing

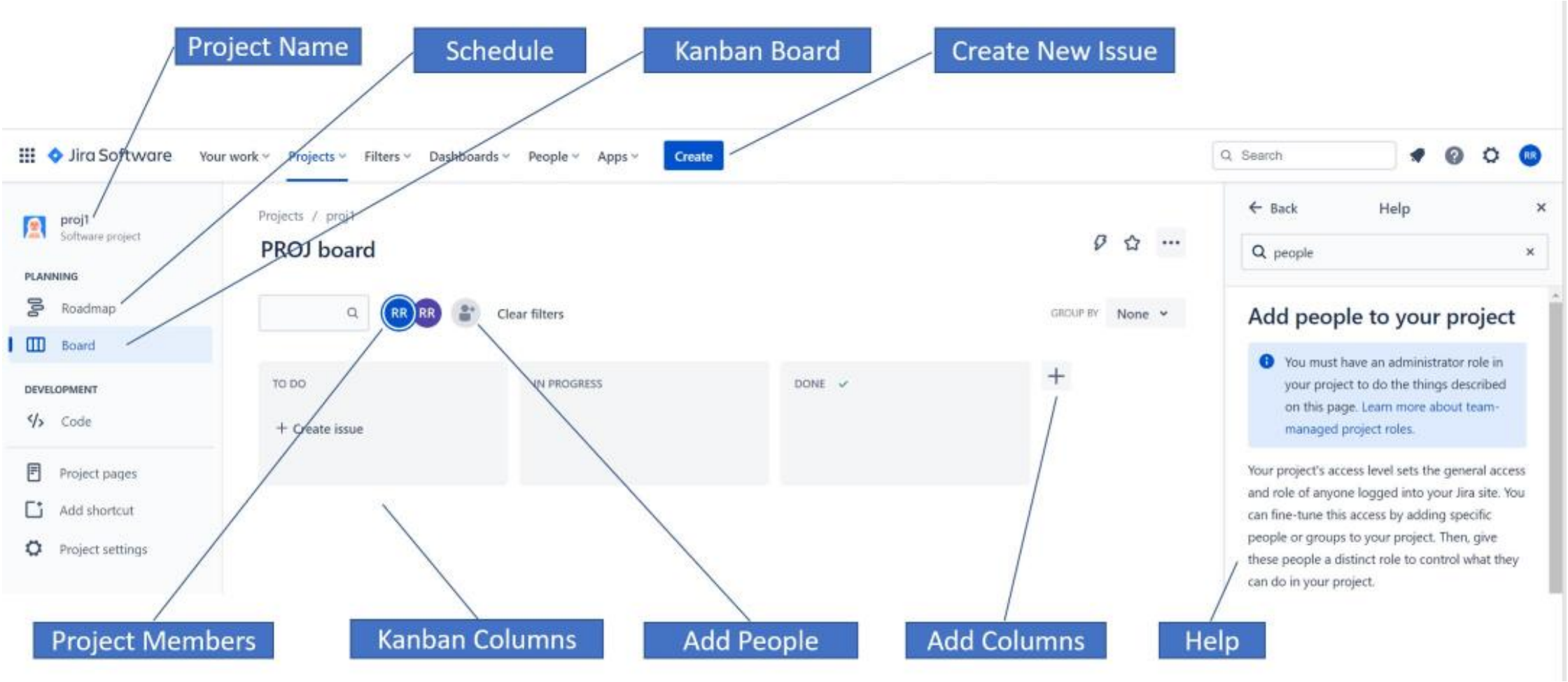
Jira

- a popular tool for managing teams working on software projects.
- It tracks Issues might include:
  - - Features which need to be built,
  - - Design documents which need to be written,
  - - Software tests which need to be written,
  - - Software tests which need to be executed,

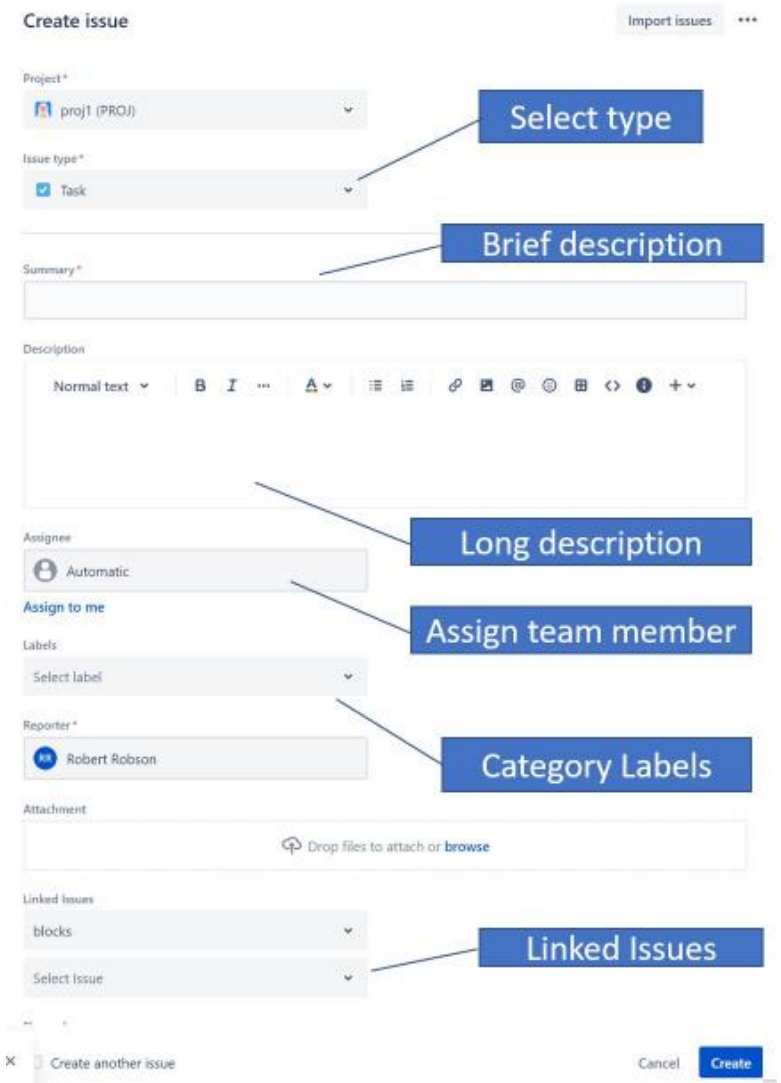
Jira Capabilities

- Add users to your project,
- - Creating issues and placing them in one of the Kanban columns,
- - Assigning issues two one of the team members,
- - Moving issues through the Kanban columns,
- - Notifying team members when the status of an issue in which they are interested changes.

Jira Interface



Creating Issues



Reporting Bugs