

Plan
Requirements and
objectives of
project defined.

→

Code
code written
by developer

Jira used to
define requirements,
create tasks, issues
to be tackled.

Code
check-in

build repository.
git used to
track changes,
collaborate, manage
source code



Build

Code compiled and
built into package
that can be deployed
Jenkins can be used

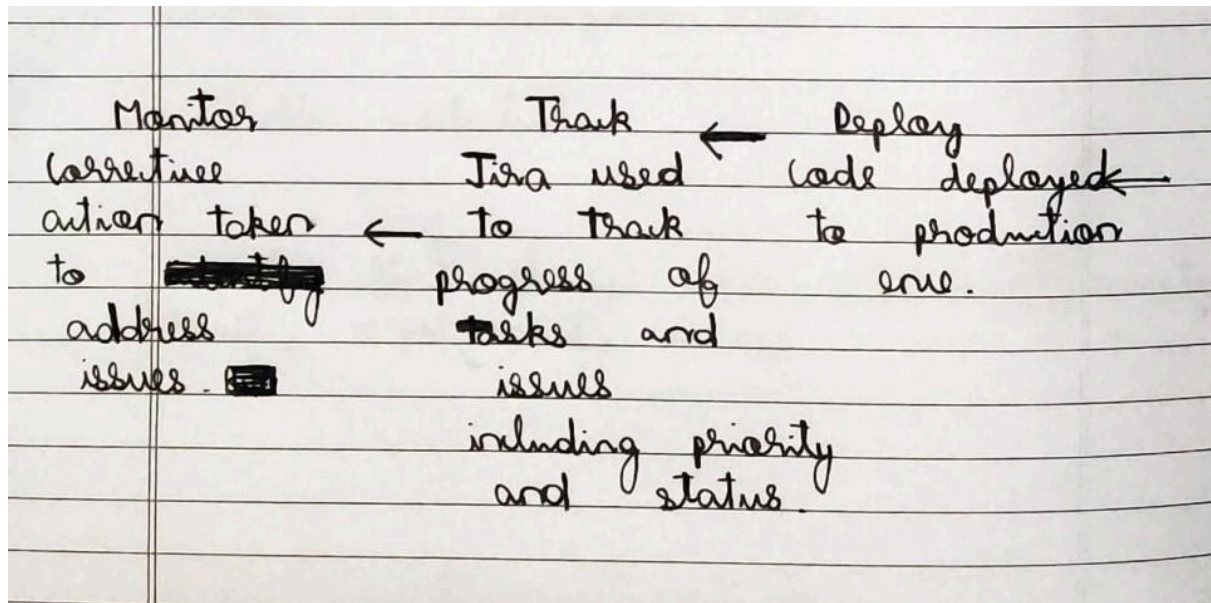


Test

Code tested to ensure
it meet specifications.

Automated testing tools
are used





Jira tickets are used in the Plan and Track stages of the DevOps process. In the Plan stage, Jira is used to define the requirements and objectives of the project, as well as to create a backlog of tasks and issues to be tackled.

In the Track stage, Jira is used to manage and track the progress of these tasks and issues, including their status, priority, and assignment.

Jira used to track work items, bugs, and issues in a software development project. Jira tickets also serve as a centralized place for communication.

Information in Jira ticket

1. **Summary:** A brief description of the issue or task.
2. **Description:** A more detailed explanation of the issue or task, including any relevant background information.
3. **Assignee:** The person responsible for completing the task or resolving the issue.
4. **Status:** The current state of the task or issue, such as Open, In Progress, Resolved, or Closed.
5. **Priority:** The level of importance assigned to the task or issue, such as High, Medium, or Low.
6. **Due Date:** The date by which the task or issue should be completed.
7. **Labels:** Keywords or tags that can be used to categorize and organize the tasks or issues.
8. **Components:** The parts of the application that are affected by the issue or task.
9. **Affects Version:** The version of the application that is affected by the issue or task.
10. **Fix Version:** The version of the application in which the issue or task will be resolved.
11. **Attachments:** Any supporting documents, screenshots, or other files that provide additional information about the issue or task.
12. **Comments:** A space for team members and stakeholders to discuss the issue or task and share updates.

Information in git commit

1. **Subject:** A brief and concise description of the changes made in the commit.
2. **Body:** A more detailed explanation of the changes made, including any context or background information. This section is optional but can be useful.
3. **Issue/Task Tracking IDs:** If issue tracking system like Jira is used, the corresponding ticket ID can be included in the commit message to link the commit to a particular task or issue.
4. **Author:** The name and email address of the person who made the commit.
5. **Date:** The date and time when the commit was made.

The information in a Git commit message helps to provide context and historical information about the changes made in a codebase.

Problems

1) Predicting failures before they occur: Machine learning algorithms can be trained on historical data from source control systems (e.g. GitHub), issue tracking systems (e.g. Jira) to predict when a failure is likely to occur.

2) Improving change management processes: Machine learning models can be used to analyze commit messages and Jira tickets to determine the impact of changes on the system. This information can then be used to prioritize changes, assess risk, and streamline the change management process.

3) Improving resource allocation: Machine learning can be used to analyze data from monitoring tools and resource utilization reports to determine the optimal allocation of resources for different stages of the DevOps process. This can help teams to optimize their processes and reduce waste.

4) Improving code quality: Machine learning algorithms can be used to analyze code and identify areas that are prone to failures or that need improvement. This information can then be used to prioritize code reviews and improve the overall quality of the codebase.

