DevOps-Agile-JIRA

DevOps Engineer: Daily Responsibilities & Activities

Collaboration & Communication

- Daily sync with Dev and Ops teams to align on tasks.
- Help developers troubleshoot deployment issues and support infrastructure needs.

Infrastructure Automation (IaC)

- Use tools like Terraform, Ansible for automating infrastructure setup.
- Ensure version-controlled infrastructure for consistency.

Version Control & Code Review

- Collaborate via GitHub/GitLab for code changes.
- Review infrastructure and automation pull requests.

CI/CD Pipeline Management

- Monitor & troubleshoot CI/CD processes (e.g., Jenkins, GitHub Actions).
- Automate build, test, and deploy workflows.

Security & Compliance (DevSecOps)

- Integrate security tools (e.g., Snyk) into CI/CD pipelines.
- Run vulnerability scans and ensure cloud security best practices.

Monitoring & Incident Response

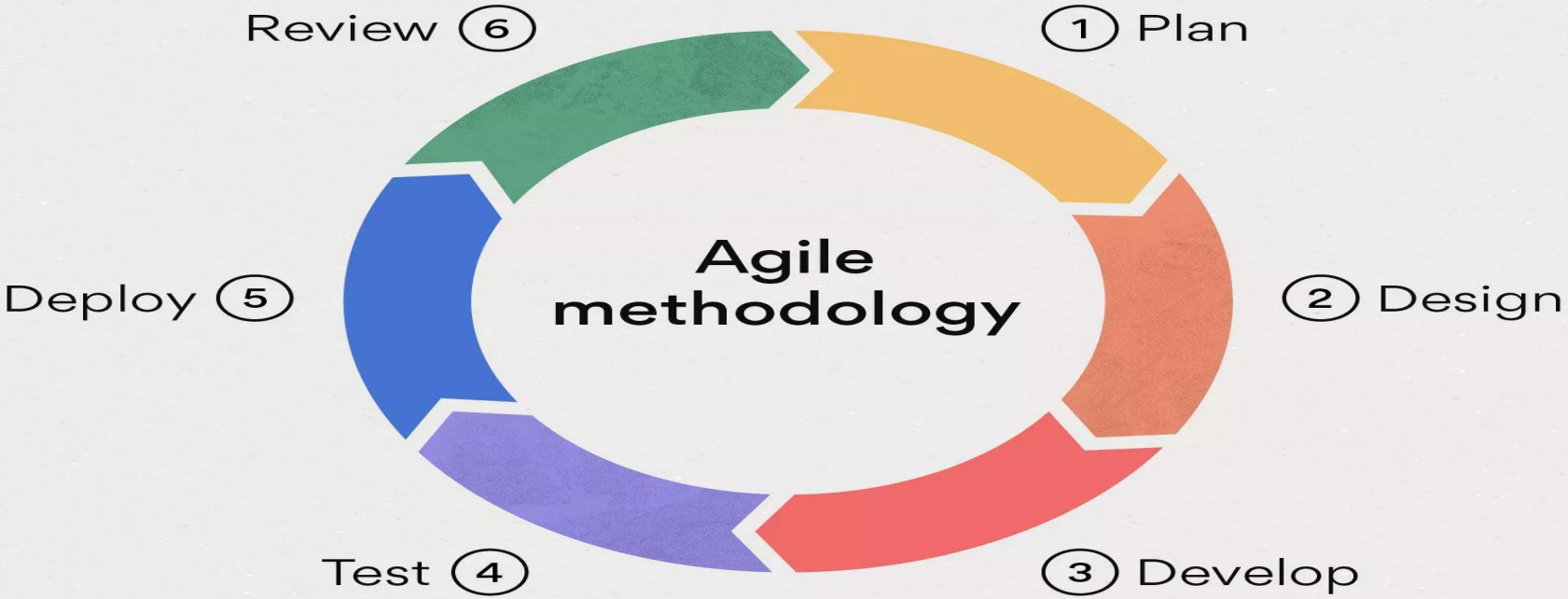
- Use tools like Prometheus, Grafana for system monitoring.
- Respond to alerts, resolve production issues quickly.

DevOps Lifecycle

What is the Agile methodology

Agile is an approach that divides work into phases, emphasizing continuous delivery and improvement. The Agile methodology benefits teams by enabling adaptive planning, rapid execution, and ongoing evaluation, leading to more responsive and successful outcomes.

Agile Methodology-



1. **What is Agile?**
    - Flexible, iterative approach for delivering software.
2. **Core Values**
    - **People & Interaction** over processes.
    - **Working Software** over documents.
    - **Customer Collaboration** over contracts.
    - **Adapt to Change** over sticking to plans.
3. **Agile Frameworks**
    - **Scrum**: Sprints, daily meetings.
    - **Kanban**: Continuous workflow.
4. **Key Roles**
    - **Product Owner**: Prioritizes features.
    - **Scrum Master**: Guides process.
    - **Team**: Delivers product.
5. **Benefits**
    - Faster delivery, flexibility, continuous feedback, better quality.
6. **Example: Shopping App**
    - Sprint 1: "Add to Cart" feature.
    - Sprint 2: "Search Products" based on feedback.

How Scrum Works-

1-Organize the backlog: Sometimes known as backlog grooming, this event is the responsibility of the product owner. The product owner's main jobs are to drive the product towards its product vision and have a constant pulse on the market and the customer.

2-Sprint planning: The work to be performed (scope) during the current sprint is planned during this meeting by the entire development team. At the end of the planning meeting, every scrum member needs to be clear on what can be delivered in the sprint and how the increment can be delivered.

3-Sprint: A sprint is the actual time period when the scrum team works together to finish an increment. Two weeks is a pretty typical length for a sprint, though some teams find a week to be easier to scope or a month to be easier to deliver a valuable increment.

4-**Daily scrum or stand up:** This is a daily super-short meeting that happens at the same time (usually mornings) and a place to keep it simple. Many teams try to complete the meeting in 15 minutes, but that's just a guideline. **The stand up is the time to voice any concerns you have with meeting the sprint goal or any blockers.**
A common way to conduct a stand up is for every team member to answer three questions in the context of achieving the sprint goal:

- What did I do yesterday?
- What do I plan to do today?
- Are there any obstacles?

5-Sprint review: At the end of the sprint, the team gets together for an informal session to view a demo of, or inspect, the increment. The development team showcases the backlog items that are now 'Done' to stakeholders and teammates for feedback. The product owner can decide whether or not to release the increment, although in most cases the increment is released.
This review meeting is also when the product owner reworks the product backlog based on the current sprint, which can feed into the next sprint planning session. For a one-month sprint, consider time-boxing your sprint review to a maximum of four hours.

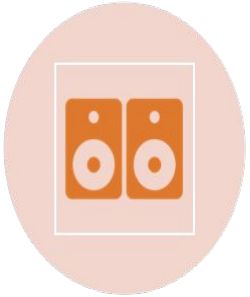
6-Sprint retrospective: The retrospective is where the team comes together to document and discuss what worked and what didn't work in a sprint, a project, people or relationships, tools, or even for certain ceremonies. The idea is to create a place where the team can focus on what went well and what needs to be improved for the next time, and less about what went wrong.

Introduction to JIRA and Scrum Planning

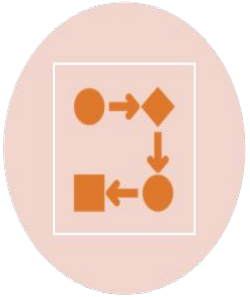
What is JIRA ?

Jira is a project-tracking tool used by agile and software development teams worldwide.

Jira lets you assign, track, audit, and effectively report your tasks. It's an all-in-one platform for project management.



Jira has over 200+ integrations support, seamlessly integrating with your favourite platforms.



Jira has the concept of Scrum, kanban boards with which project tracking is effectively done.

Jira Features and Capabilities

1-Issue Tracking

Jira's core functionality is its robust issue tracking system, allowing teams to log, prioritize, and manage tasks, bugs, and other project-related issues efficiently.

2-Reporting and Analytics

Jira's reporting capabilities provide teams with valuable insights into project progress, team productivity, and other key performance metrics, aiding in data-driven decision-making.

3- Agile Project Management

Jira integrates seamlessly with Agile methodologies, enabling teams to create and manage sprints, backlogs, and Kanban boards. This helps teams visualize and optimize their workflows.

Epic, Stories, and Tasks - The Hierarchy of Jira

In Jira, the project management hierarchy consists of Epics, Stories, and Tasks. Epics represent the largest units of work, which can be broken down into smaller, more manageable Stories. Each Story is then further divided into individual Tasks that need to be completed to realize the Story and, ultimately, the Epic.

Backlogs in Jira

- Jira's Backlogs provide a **centralized** view of all the work items that need to be completed for a project.
- Product Backlogs contain the high-level **Epics** and **Stories** that define the overall functionality and features to be delivered.
- Sprint Backlogs contain the specific **Tasks** that the team has committed to completing during the current Sprint.
- Backlogs can be prioritized and **groomed** to ensure the most important and valuable work is addressed first.
- Backlogs in Jira enable teams to **visualize**, **manage**, and **track** their work in a clear and organized manner.

Sprints in Jira

1. **Time-boxed Iterations** - Jira's Sprints are time-boxed iterations, typically 1-4 weeks, where teams plan, execute, and review their work.
2. **Sprint Planning** - At the start of each Sprint, teams hold a planning session to select and commit to the work they'll complete during that Sprint.
3. **Sprint Boards** - Sprint boards in Jira provide a **visual** way for teams to manage their work, track progress, and identify blockers.
4. **Burndown Charts** - Jira's Burndown charts help teams monitor their Sprint progress and ensure they're on track to complete their commitments.
5. **Sprint Reviews** - At the end of each Sprint, teams hold a review session to demonstrate the work they've completed and gather feedback.

Scrum Boards in Jira

Visualization:-
Scrum boards in Jira provide a clear, visual representation of the team's workflow, allowing everyone to see the status of tasks at a glance.

Transparency:-
Scrum boards promote transparency, allowing the entire team to see what everyone is working on and identify any blockers or bottlenecks.

Workflow Management:-
These boards enable teams to easily manage their Scrum process, moving tasks through the various stages of development such as To Do, In Progress, and Done.

Collaboration:-
The boards facilitate collaboration, as team members can leave comments, assign tasks, and update the status of work items in real-time.

Scrum Boards in Jira:-

Visualization

Scrum boards in Jira provide a clear, visual representation of the team's workflow, allowing everyone to see the status of tasks at a glance.

Workflow Management

These boards enable teams to easily manage their Scrum process, moving tasks through the various stages of development such as To Do, In Progress, and Done.

Transparency

Scrum boards promote transparency, allowing the entire team to see what everyone is working on and identify any blockers or bottlenecks.

Collaboration

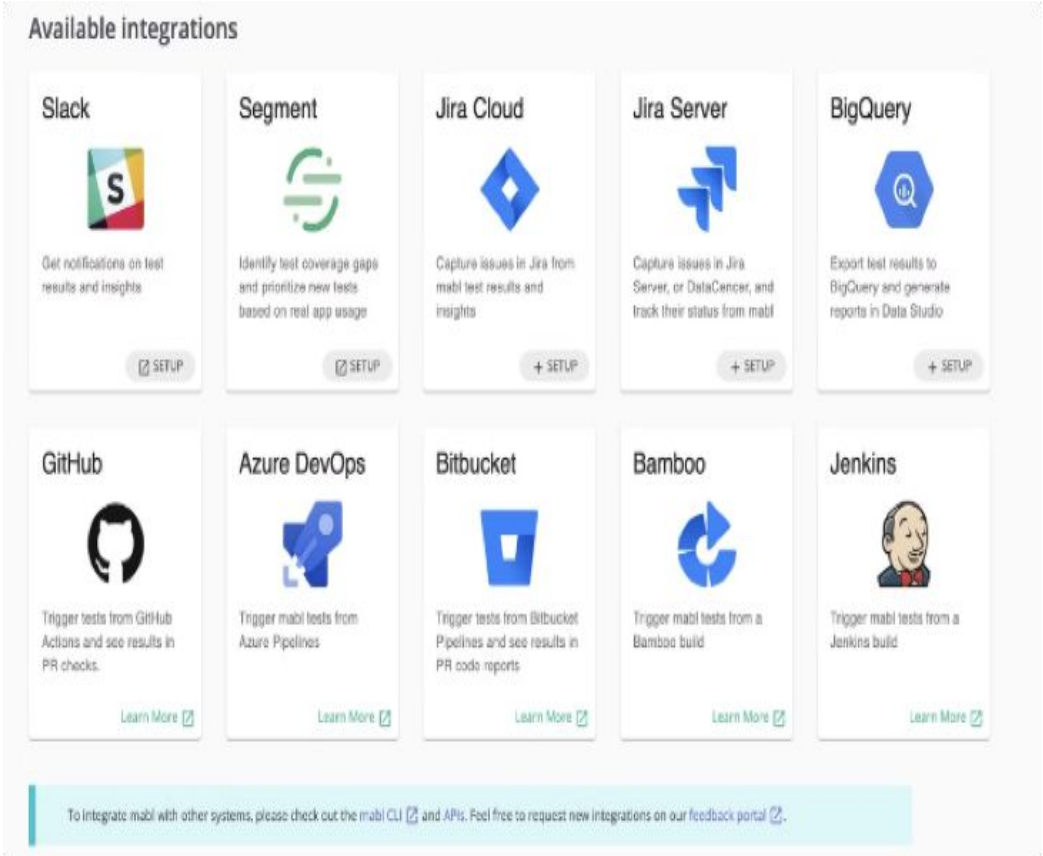
The boards facilitate collaboration, as team members can leave comments, assign tasks, and update the status of work items in real-time.

Jira Integration with Other Tools

Git and GitHub
Jira integrates with popular version control systems, allowing teams to link code commits and pull requests to issues.

Confluence
Jira's deep integration with Confluence, Atlassian's collaboration tool, enables teams to create and link documentation directly to their projects.

Third-Party Tools
Jira can be integrated with a wide range of third-party tools, such as Slack, Microsoft Teams, and Trello, to streamline workflows and data sharing.



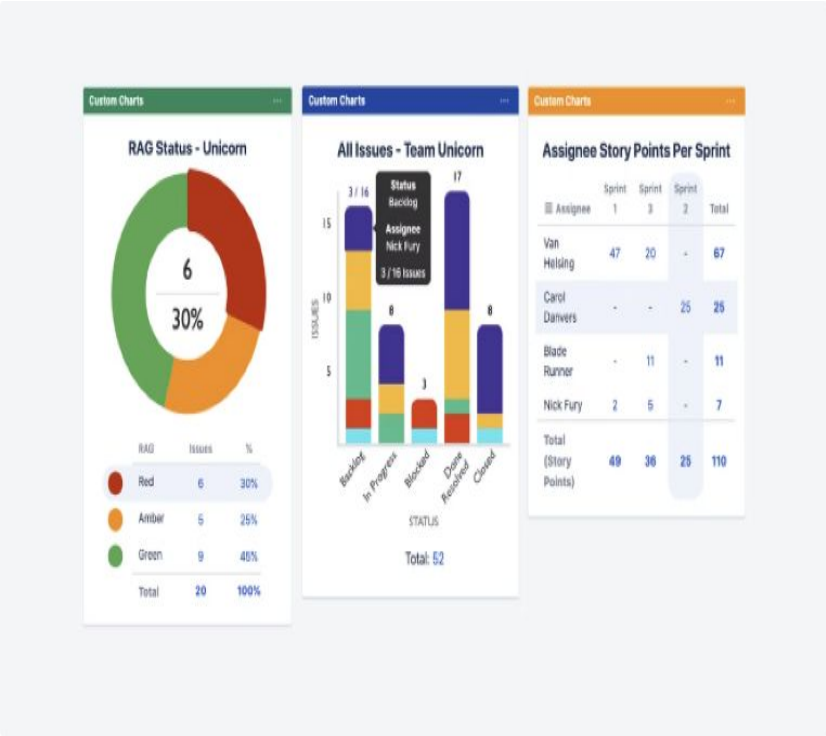
Jira Customization and Configuration

Workflows
Customize Jira's workflows to match your team's processes and requirements.

Project Templates
Create and customize project templates to streamline the setup process.

Custom Fields
Add custom fields to capture the unique data your team needs to track.

Permissions
Configure user permissions and access levels to control who can view and edit project data.



Github and JIRA Integration:-

Why Needed -

Problem: Developers work in GitHub, while project managers and stakeholders track tasks and issues in Jira. Without integration, there's often a disconnect between code changes and project progress.

Solution: The integration links GitHub commits, pull requests, and branches directly to Jira issues. This means that everyone can easily track the development progress of specific tasks or features.

Example: If a developer mentions a Jira issue (e.g., JIRA-123) in a commit or pull request, that information automatically updates in Jira. This gives full transparency on which code changes are related to which tasks.

Step by Step Method -

Install the GitHub for Jira app

1. In Jira, select **Apps**, then select **Explore more apps**.
2. Search for **GitHub for Jira**, then select it from the results.
3. Select **Get app**, then **Get it now**.

Connect a GitHub organization

1. After the app is installed, select **Get started**. If the app is already installed on your Jira site, you can find this section by selecting **Apps**, then **Manage your apps**, and then **GitHub for Jira**.
2. Select **Continue**.
3. Select **GitHub Cloud**, then **Next**.
4. Enter your GitHub username and password, then **Sign in**.
5. Find the organization you want to connect to Jira, then select **Connect**.

Add the Jira app to a new GitHub organization

If no organizations are available to connect to Jira, you'll need to install the Jira app on a new GitHub organization.

1. From step five in the instructions above, **Select an organization in GitHub.**
2. The GitHub site will open in a new tab and show a list of all the organizations available in your GitHub account. Select the organization you want to connect to Jira.
3. Choose the repositories you want to give Jira access to by selecting either **All repositories** or **Only select repositories**.
4. Select **Install.**