

Devops

PROJECT MANAGEMENT TOOLS FOR DEVOPS

WHAT DOES A DEVOPS ENGINEER DO WHEN JOINING AN ORGANIZATION?

When you join an organization as a DevOps engineer:

First priority: Understand the tools that the organization uses to track work and understand how things work.

This is why we call it project management.

AGILE METHODOLOGY

WHAT IS AGILE?

Agile is a methodology or process.

Fundamental idea of Agile: Instead of traditional waterfall method, break work into smaller iterations.

TRADITIONAL WATERFALL METHOD

How it works:

Step 1: Get feature Feature comes from Product Owner or Business Analyst once the product is finalized.

Step 2: Design phase There is a phase in SDLC where you define your idea and describe your idea as part of HLD and LLD.

What is HLD and LLD:

- HLD: High-Level Design
- LLD: Low-Level Design

Step 3: Prepare design specification Document the complete design.

Step 4: Development Design goes to the developer and developer will write the code.

Step 5: QA Code goes to QA for testing.

Step 6: Customer delivery Tested application goes to the customer.

Problem with waterfall:

If we are using traditional waterfall method: Everything happens in one single go.

Example:

- Develop the entire application (might take months)
- After this, it goes to QE
- Then to the customer

Delivery: We are delivering the entire application, not a feature.

This is what is followed in waterfall methodology.

Issues:

- Long development cycle
 - Late feedback
 - High risk
 - Difficult to change requirements
 - QA waits for months
-

AGILE METHODOLOGY

How it works:

In Agile: We break the application into chunks or features and we just work on these small chunks or features and deliver them to the customer in sprints.

What is a sprint: Sprint is a timeline that we define in which we deliver a part of a feature or a complete feature.

Typical sprint duration:

- 1 week
 - 2 weeks (most common)
 - 3 weeks
 - 4 weeks
-

Agile process:

Step 1: Break application into features Instead of building entire app, identify individual features.

Step 2: Work on small feature Developer works on one feature for 2 weeks (one sprint).

Step 3: Deliver to QA Pass this feature to QE environment or staging environment where QA can perform testing.

Step 4: Quick feedback QA tests and provides feedback quickly.

Step 5: Quick fixes If there is something wrong, developers have quick time to improvise this small code.

Step 6: Repeat Next sprint, work on next feature.

Why Agile is efficient:

Instead of QA waiting for longer duration to test the code: Developer can just work on a small feature, then pass it to QE environment where QA will test the code.

If issues found: Easier to fix small code than modifying entire application.

Comparison:

Waterfall:

- Develop entire app (6 months)
- Find major bug in QA
- Rework takes months
- Delayed delivery

Agile:

- Develop one feature (2 weeks)
- Find bug in QA
- Fix in days
- Continue to next feature

Agile is very important and when we join an organization which is using Agile, basic Agile knowledge is important.

JIRA - PROJECT MANAGEMENT TOOL

WHAT IS JIRA?

Jira is a tool used for project management where you can track your work.

USE CASE 1: EXECUTIVE LEVEL TRACKING

Scenario:

CEO wants to look at priority 1 issues in the organization.

Problem without Jira:

- Cannot go and ask each manager individually
- Cannot email each manager
- Time consuming
- No centralized view

Solution with Jira: There will be one centralized Jira board.

How it works:

1. CEO goes to Jira
2. Writes a Jira query to pull all:
 - Jira issues or
 - Jira epics or
 - Jira stories
 - With priority 1
3. Gets list of all priority 1 items
4. Can see status, assignee, progress

This is how Jira is useful.

USE CASE 2: PROJECT TRACKING

If we go one step down: Jira is also used to track your projects.

Scenario: We are working in a development team of 10 members.

Manager's needs: My manager wants to assign tasks.

In Agile terminology: We have a Scrum Master who does the sprint planning.

Sprint planning: We get the stories that are prioritized.

Once we enter a sprint: We have a list of activities assigned.

How Jira helps: We can go to our Jira board and see which stories are assigned to you.

Tracking work history:

If someone wants to track your work: See what you have done maybe 2 months back or so.

Solution: They can track all this as part of Jira.

What can be tracked:

- Team's velocity (how much work completed per sprint)
- Team's efforts in a sprint
- How much you are able to accomplish and achieve

This also can be tracked using Jira.

Conclusion: It is an efficient project management tool and once you get a good understanding of Jira, it helps in daily work.

JIRA KEY CONCEPTS

Epic: Large body of work that can be broken down into stories.

Example: Epic: User Authentication System

Story: Smaller unit of work that delivers value to user.

Example: Story: Implement login functionality

Task: Specific work item to complete a story.

Example: Task: Create login API endpoint

Sprint: Fixed time period (usually 2 weeks) to complete selected stories.

Backlog: List of all pending work items.

Board: Visual representation of work items and their status.

Columns:

- To Do
 - In Progress
 - In Review
 - Done
-

CONFLUENCE - KNOWLEDGE SHARING PLATFORM

WHAT IS CONFLUENCE?

Confluence and SharePoint are knowledge sharing platforms.

What are knowledge sharing platforms:

Purpose: Place to store and share documentation within organization.

WHY KNOWLEDGE SHARING PLATFORMS ARE NEEDED

Scenario:

Public documentation: If you are working with AWS or GCP, you go to their official documentation.

Internal applications: Your organization has a product that is internally used by your team only.

Problem: There has to be a place where documents have to be placed. You cannot put all of the documents on public platforms.

Solution: Confluence and SharePoint are platforms where:

- Provide single sign-on
- Integrate with your organization

- Everybody gets to create an account
 - Every project can create their own pages
 - Start documenting and sharing knowledge
-

USE CASES FOR CONFLUENCE

Use case 1: Technical documentation

- API documentation
- Architecture diagrams
- Setup guides
- Troubleshooting guides

Use case 2: Process documentation

- Deployment procedures
- Incident response playbooks
- Change management process
- Onboarding guides

Use case 3: Team collaboration

- Meeting notes
- Decision records
- Project updates
- Knowledge base

Use case 4: Integration with Jira

- Link documentation to Jira issues
 - Embed Jira reports
 - Track requirements
-

SERVICENOW - INCIDENT AND CHANGE MANAGEMENT

WHO NEEDS SERVICENOW?

Not every DevOps engineer needs to have an account on this.

But in some DevOps engineer roles: There is change management and incident management.

When ServiceNow is used:

Scenario 1: You are working in a service-based organization.

Scenario 2: You are working extensively with clients.

Scenario 3: You have your application deployed on your platform and client is using the application from your platform (not any cloud platform).

In these cases: ServiceNow is very extensively used.

TWO KEY ASPECTS OF SERVICENOW

1. Change Management
 2. Incident Management
-

INCIDENT MANAGEMENT

WHAT IS INCIDENT MANAGEMENT?

Background: Monitoring is very key in DevOps roles.

Scenario:

You have deployed your application onto a production system. It can be on production servers.

Customer usage: Customer is using the application.

Problem occurs:

One day: Because of unexpected loads or load on the application, or due to some kind of leak in the server, or because of any reason, your app is not reliable or your app is not working as expected.

Monitoring system response:

In this situation: Our monitoring system will observe this and throw an alert.

Questions:

- Who is receiving these alerts?
- Who is acting upon these alerts?

Requirement: These alerts have to be immediately responded to and your system has to be immediately fixed.

Incident management process:

This entire process can be tracked as incident management.

Step 1: Alert thrown Once the alert is thrown by the monitoring system.

Step 2: Create incident This incident management system has to create an incident saying:

- There is an alert thrown by your monitoring system
- Specific team has to immediately look into it
- Fix it right away on the production system

Step 3: Assignment This entire thing has to be created as an incident and has to be sent to a specific person.

Step 4: Start work This specific person, once he starts working on it, will put a message saying:

- "I have started working on it"
- "This is the current status"

Step 5: Resolution Messages updated:

- "The resolution is done"
- "We made these changes"

Step 6: Deployment Finally:

- "We have deployed the application on production again"
- Incident resolved and closed

ServiceNow for incident management:

ServiceNow is such a system where: We can create such incidents.

Integration: Your monitoring system can be directly integrated with the ServiceNow API.

Automation:

- Monitoring system detects issue
- Creates incident automatically in ServiceNow
- No manual intervention needed

Assignment logic: Incident assigned depending on:

- Type of incident
- Priority level
- Team ownership

Example: If incident has:

- Priority 1
- Coming from team called "Payments"

Then: It will be assigned to a person from payment team with top priority.

CHANGE MANAGEMENT

WHAT IS CHANGE MANAGEMENT?

Background: Your incident has been identified and you created an incident and it is assigned to a specific person.

Next step: This person understands the root cause and wants to go ahead and make changes in the system.

Problem with production changes:

On production system: This is not that easy.

He cannot: Go and say "I identified a change and I need to make a change and I will ask customers to not use this system for next 15 minutes."

Reality: That's not possible. You cannot tell your customers to stop using the app.

Change management process:

This person has to go through a process: This process is called change management.

Steps:

Step 1: Create ServiceNow ticket Create change request in ServiceNow.

Step 2: Identify stakeholders

- Vendors involved
- List of teams who can work on this issue
- People who need to be notified
- Approvers required

Step 3: Document changes

- What will be changed
- When it will be changed
- How long it will take
- Rollback plan
- Impact analysis

Step 4: Get approvals

- Technical approval
- Business approval
- Security approval

Step 5: Schedule change window Plan maintenance window:

- Low-traffic period
- Communicate to customers
- Backup current state

Step 6: Implement change Execute changes during approved window.

Step 7: Verify Test that changes work correctly.

Step 8: Close ticket Document results and close change request.

All of these things have to be identified or tracked as part of change management.

Why change management is important:

Control: Prevents unauthorized changes to production.

Documentation: Track what changed, when, and why.

Approval: Ensures changes are reviewed.

Communication: Stakeholders know about changes.

Rollback: Plan exists if something goes wrong.

Audit: Compliance and regulatory requirements met.

READ THE DOCS

WHAT IS READ THE DOCS?

Read the Docs is an open-source tool similar to Confluence and SharePoint.

Purpose: Documentation hosting platform.

Features:

- Free and open source
- Hosts documentation
- Supports Markdown, reStructuredText
- Automatic builds from Git
- Versioning support
- Search functionality

Use case:

- Open source projects
- Public documentation
- Small teams
- Cost-conscious organizations

Difference from Confluence:

- Read the Docs: Open source, free, simpler
 - Confluence: Paid, enterprise features, more powerful
-

GIT AND GITHUB FOR PROJECT TRACKING

ALTERNATIVE TO JIRA

Some people use Git and GitHub instead of Jira to track the project work.

How GitHub can replace Jira:

GitHub Issues: Similar to Jira stories.

Features:

- Create issues
 - Assign to team members
 - Add labels
 - Track status
 - Add comments
-

GitHub Projects: Similar to Jira boards.

Features:

- Kanban boards
 - Track issue status
 - Organize work
 - Sprint planning
-

GitHub Milestones: Similar to Jira sprints.

Features:

- Group issues
- Set due dates

- Track progress
-

GitHub Wiki: Similar to Confluence.

Features:

- Documentation
 - Internal knowledge base
 - Markdown support
-

When to use GitHub for project management:

Scenario 1: Small team (fewer than 10 people).

Scenario 2: Already using GitHub for code.

Scenario 3: Simple project tracking needs.

Scenario 4: Want everything in one place.

Scenario 5: Cost-conscious (Jira costs money).

When to use Jira:

Scenario 1: Large organization.

Scenario 2: Complex project tracking needs.

Scenario 3: Need advanced reporting.

Scenario 4: Non-technical stakeholders.

Scenario 5: Enterprise features required.

COMPARISON TABLE

Aspect: Project tracking

Jira:

- Epics, stories, tasks
- Sprint planning

- Advanced boards
- Custom workflows

GitHub:

- Issues
 - Projects (Kanban)
 - Milestones
 - Simple workflows
-

Aspect: Documentation

Confluence:

- Rich text editor
- Templates
- Page hierarchy
- Powerful search

GitHub Wiki:

- Markdown
 - Version controlled
 - Simple
 - Integrated with code
-

Aspect: Incident management

ServiceNow:

- Full incident lifecycle
- SLA tracking
- Assignment rules
- Integration with monitoring

GitHub:

- Can create issue

- Manual process
 - No SLA tracking
 - Basic functionality
-

Aspect: Change management

ServiceNow:

- Formal approval process
- Change windows
- Risk assessment
- Audit trail

GitHub:

- Pull request workflow
 - Code review
 - No formal CAB
 - Developer-focused
-

Aspect: Cost

Jira: Paid (per user)

Confluence: Paid (per user)

ServiceNow: Expensive (enterprise)

GitHub: Free for public, paid for private

Read the Docs: Free (open source)

Aspect: Learning curve

Jira: Moderate

Confluence: Easy

ServiceNow: Steep

GitHub: Easy

Read the Docs: Easy

Aspect: Integration

Jira: Extensive (thousands of integrations)

Confluence: Good (Atlassian ecosystem)

ServiceNow: Extensive (enterprise focus)

GitHub: Good (developer tools)

Read the Docs: Limited

Aspect: Best for

Jira: Medium to large organizations

Confluence: All organization sizes

ServiceNow: Large enterprises with ITIL processes

GitHub: Development teams

Read the Docs: Open source projects, documentation