CSCI 4350-002 DevOps & Automated Deployment

Trent Williams

Sprint 1 – CHKN-34 [02/07/2024 – 02/20/2024] ChickenContenders

DevOps

What is DevOps?

DevOps is the process of combing **“development”** and **“operations”** under one roof. Traditionally, development and operation teams are **“siloed.”** Tearing down these silos enables teams to work from the development side to the deployment side. DevOps is a philosophy that emphasizes communication between teams and automation. The main focus of DevOps is to delivery the product faster while still maintaining its quality. There are four main components that make up DevOps.

Continuous Integration (CI)

Continuous Integration is a practice where code is merged into a central repository. The general process follows: Developers implement their code. Then test it via unit testing. Then the developer pushes their code to a centralized repository containing proper version control methods. Once the product is here, automated or manual test are run to determine the product’s health. Finally, the code is ready for deployment. Generally, this occurs several times a day.

Continuous Delivery (CD)

Continuous Delivery follows CI. After all the steps of the CI pipeline are completed, the product is pending deployment. CD picks up here. CD pushes the code into production. CD is automated thus resulting in less human errors. CD tests the new code, verifies it, then releases it into production. Not only does automation result in lower risk, it results in lower cost and quicker time to market. The latter results in quicker feedback from end-users. CD focusses more on The First Way.

Continuous Testing

Continuous Testing is a requirement for DevOps. Continuous Testing test the code at every step during CI. It is mandatory to ensure that the product works according to the requirements. The process follows: After automated test confirms there are no errors it is pushed to the QA environment where functional testing is done. After that it is pushed into production. It is important that Continuous Testing is automated to ensure it has no errors. Then it can be pushed to manual testing.

Continuous Monitoring

Continuous Monitoring is done by the operation team while it is in production. This ensures that the product is stable and available.

DevOps is defined into three sections:

The First Way

This First Way is described as “maximizing left-to-right flow of work through a process.” (Ramsey) The first way speeds up the left-to-right flow of work from Development to Operations to Customer. This **makes the work visible** using software like Kanban boards. Items from the backlog would start being added to the left, then moved into each phase on the right as it is being completed. Finally, you would move into done as it is being put into production. You also have to limit Work in Progress (WIP). We can also use a Kanban board to solidify what the limits of work are. Limiting WIP allows problems to be seen earlier that could hinder the completion of work. Reduce batch sizes. Smaller batch sizes can help you find and fix issues sooner rather than having a large chunk to fix. Reducing the number of handoffs. By limiting the number of handoffs between teams, we reduce the dependency of teams on other teams. Each team builds with quality to ensure that when it is passed off, the next team has fewer issues. This enables each team to be able to deliver value to the customer. Finally, we build systems that are safe to change. By building systems that are easily modifiable can allow for changes.

The Second Way

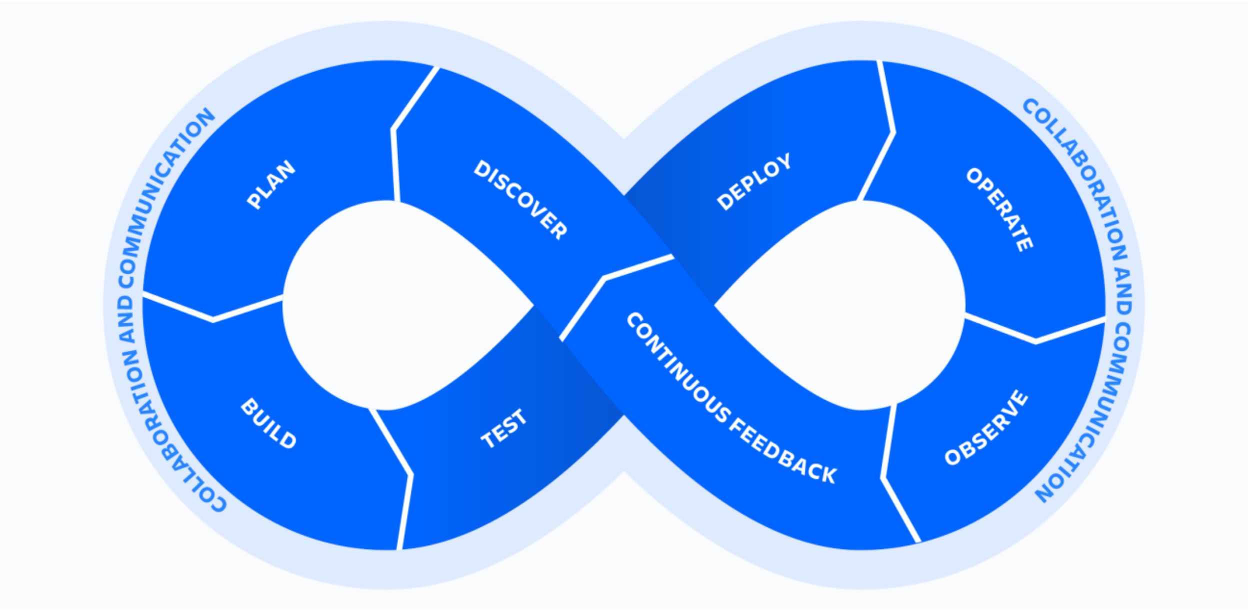
The Second Way: Amplify Feedback loops. The second way implements fast, constant feedback loops from left to right. This requires amplifying feedback to prevent problems from reoccurring and that we enable faster detection and recovery. By amplifying feedback, we can see what problems are occurring. We take data from the first way like boot time, usage patterns, performance, time to deploy, etc. and we can modify our product based upon these data points we have monitored thus stream lining the release. By enabling faster detection and recovery, we use feedback loops at all stages to be able to see where problems occur. By doing this, we can detect which stage a problem occurred in. These result in creating quality at the source, generating knowledge where it is needed, and creating safer systems where problems are found and fixed long before a catastrophic event.

The Third Way

The Third Way: Continual Experimentation and Learning. This enables the creation of a high-trust culture that supports a scientific approach to experimentation and risk taking. To do this, we need to facilitate organization learning from both success and failures, design a system that allows new knowledge to be multiplied, and we transform local discoveries into global improvements. Rather than blaming human error when an accident occurs, we would evaluate the system to see how it can be prevented from happening again. We can mitigate discoveries by using organizational wide source control. These result in no fear of change or fear or risk, faster innovations, and global knowledge.

DevOps Lifecycle (1)

According to Atlassian.com


COLLABORATION AND COMMUNICATION
PLAN
DISCOVER
DEPLOY
OPERATE
COLLABORATION AND COMMUNICATION
BUILD
TEST
CONTINUOUS FEEDBACK
OBSERVE

Provided by Atlassian.com

Discover

Building software is a team sport. In preparation for the upcoming sprint, teams must workshop to explore, organize, and prioritize ideas. Ideas must align to strategic goals and deliver customer impact. Agile can help guide DevOps teams.

Plan

DevOps teams should adopt agile practices to improve speed and quality. Agile is an iterative approach to project management and software development that helps teams break work into smaller pieces to deliver incremental value.

Build

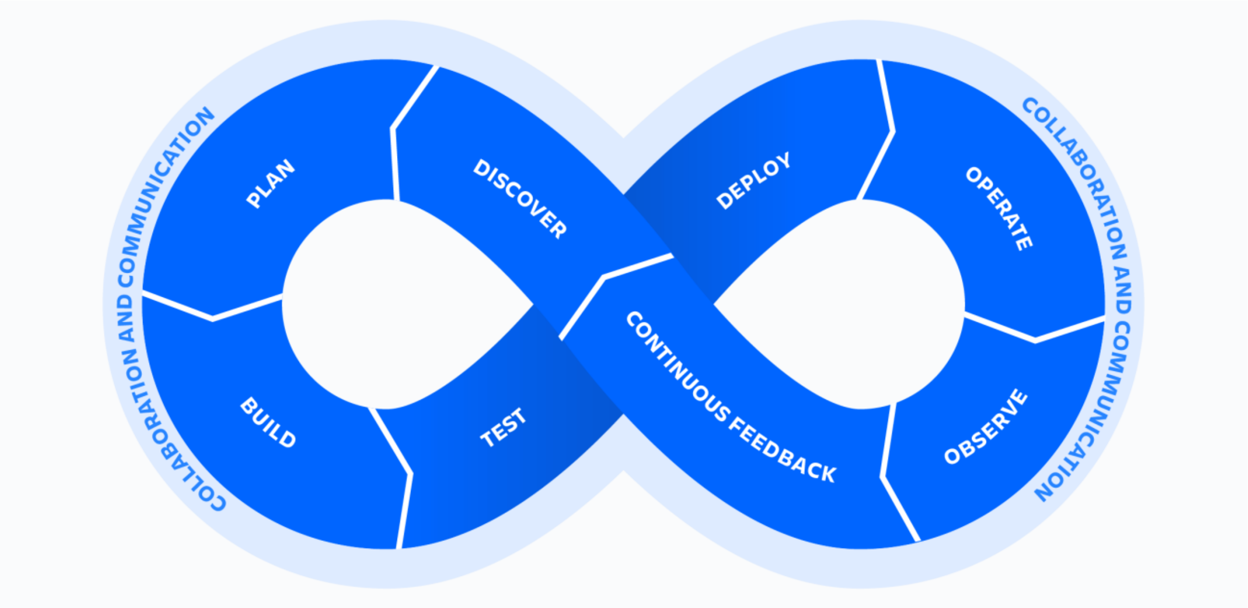
Git is a free and open-source version control system. It offers excellent support for branching, merging, and rewriting repository history, which has led to many innovative and powerful workflows and tools for the development build process.

Test

Continuous integration (CI) allows multiple developers to contribute to a single shared repository. When code changes are merged, automated tests are run to ensure correctness before integration. Merging and testing code often help development teams gain reassurance in the quality and predictability of code once deployed.

DevOps Lifecycle (2)

According to Atlassian.com


PLAN
DISCOVER
DEPLOY
OPERATE
BUILD
TEST
CONTINUOUS FEEDBACK
OBSERVE
COLLABORATION AND COMMUNICATION
COLLABORATION AND COMMUNICATION

Provided by Atlassian.com

Deploy

Continuous deployment (CD) allows teams to release features frequently into production in an automated fashion. Teams also have the option to deploy with feature flags, delivering new code to users steadily and methodically rather than all at once. This approach improves velocity, productivity, and sustainability of software development teams.

Operate

Manage the end-to-end delivery of IT services to customers. This includes the practices involved in design, implementation, configuration, deployment, and maintenance of all IT infrastructure that supports an organization’s services.

Observe

Quickly identify and resolve issues that impact product uptime, speed, and functionality. Automatically notify your team of changes, high-risk actions, or failures, so you can keep services on.

Continuous Feedback

DevOps teams should evaluate each release and generate reports to improve future releases. By gathering continuous feedback, teams can improve their processes and incorporate customer feedback to improve the next release.

Automated Deployment

Continuous Deployment incorporates CI, CD, Continuous Testing, and Continuous Feedback. Automation is key to all of these. By automating deployment, you can build. Package, test, and release the product into staging.

Deployment automation is a key component of DevOps practices, aiming to improve the efficiency and reliability of the software development lifecycle. In this context, deployment refers to the process of releasing and updating software applications. Traditionally, this process involved multiple manual steps, which could be time-consuming and prone to errors.

One of the significant advantages of deployment automation is its ability to reduce reliance on manual intervention. Instead of requiring human approval for each step, automated systems can be configured to perform these tasks based on predefined criteria. This not only accelerates the deployment process but also minimizes the risk of human error.

The deployment automation process can be scheduled to run at specific times or triggered automatically based on certain conditions, such as successful completion of tests or code commits. This flexibility allows for a more responsive and adaptive deployment pipeline.

There are many automated deployment tools. Jira and Jenkins are both great tools that we could incorporate into out project.

Jira

We already use Jira for our SCRUM board. With Jira, we can integrate applications to automate our deployment from our repository. With Jira we get the following benefits.

- Track the deployment status of work on your team's Jira issues and the board.
- Get visibility into what issues have been deployed at a given point in time via the deployments timeline.
- View deployment insights to track your team's performance over time and identify bottlenecks.
- Use automations to reduce admin tasks and "busywork" for your team (for example, updating an issue's status once it's deployed successfully).
- For teams following DevOps practices like CI/CD, integrating your deployments with Jira provides a consolidated timeline view of all your automated deployments. This makes it easier to fix release issues as you can quickly see which features have deployed to which environment.

With Jira we can integrate Jenkins.

Jenkins

Jenkins is an open-source automation server commonly used for building, testing, and deploying software. It facilitates continuous integration and continuous delivery (CI/CD) practices in software development. Jenkins automates the process of integrating code changes from multiple contributors and ensures that software is consistently built, tested, and deployed. Jenkins for Jira provides a free, secure, and reliable way to connect your Jenkins server, running behind the firewall, with either Jira Software Cloud or Jira Service Management Cloud.

Citations:

What is DevOps?", Atlassian, https://www.atlassian.com/devops.

Bose, Shreya. "What is DevOps? (A Beginner's Guide)." BrowserStack, BrowserStack Inc., 11 Jan. 2023, https://www.browserstack.com/guide/devops-for-beginners.

Vaidya, Neha. "Continuous Integration with Agile." BrowserStack, BrowserStack Inc., 17 Jun. 2023, https://www.browserstack.com/guide/continuous-integration-with-agile.

Kim, Gene. "The Three Ways: The Principles Underpinning DevOps." IT Revolution, IT Revolution Press, 22 Aug. 2012, https://itrevolution.com/articles/the-three-ways-principles-underpinning-devops/.

"Continuous Integration." Amazon Web Services, Amazon.com Inc., https://aws.amazon.com/devops/continuous-integration/.

"Continuous Delivery." Amazon Web Services, Amazon.com Inc. https://aws.amazon.com/devops/continuous-delivery/.

Kim, Gene, et al. The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations. IT Revolution Press, 2016.

"Deployment automation: What is it and how to start." Atlassian, https://www.atlassian.com/devops/frameworks/deployment-automation.

"Integrate Your Deployments with Jira." Atlassian Support, Atlassian Pty Ltd, Date of publication (if available), https://support.atlassian.com/jira-cloud-administration/docs/integrate-your-deployments-with-jira/.

"Integrate Jenkins with Jira." Atlassian Support, https://support.atlassian.com/jira-cloud-administration/docs/integrate-jenkins-with-jira/.