Kendall Taylor

DevOps

Module 1.3 Assignment

January 12th, 2024

History of DevOps

Before DevOps and Agile Methodology, software developers relied mainly on the waterfall method. The waterfall method broke down project processes and activities into different phases. Each phase within this method relied on the previous stage and allowed no overlap in phases. The issue with these designs was no flexibility and adaptability, little to no room for errors, and customer involvement was minimal.

In the 1990's, the Agile methodology was implemented to help speed up processes and create a smoother method to technology delivery. This methodology focused on smaller parts and features of an application rather than the whole. It allowed teams to work in shorter cycles known as sprints to make continuous improvements and deliveries. Overall, this method allowed for quicker development times, more flexibility, quality testing, and less overall risk during each process.

In 2007, DevOps came into play to aid in bridging the gaps between development and operation teams. In 2008, the idea of DevOps was discussed. During this discussion, they brainstormed and suggested solutions to get operation teams to become more agile. In 2010, a book about continuous delivery was released, setting out principles for practices allowing for quick delivery and functions to users.

The impact that DevOps had on software development is huge. It allows for quicker and more efficient delivery of software and updates, stronger collaboration between teams, better quality products, and overall higher customer satisfaction. Like most things technology, DevOps is constantly improving and evolving. This allows software development processes to improve and become more efficient.

The Lean Movement

The Lean Movement is an agile framework that is used to optimize and speed up the software development process. The goal here is to become more efficient and minimize waste during processes. The Lean Movement was brought up in the 1980's and has evolved to benefit modern-day software development.

Main Focuses:

- Eliminate Waste: identify unnecessary codes and processes.
- Quick Delivery: Focus on less complex systems, and create improvements based on customer feedback

- Learning Opportunities: Code reviews and meetings allow for developers to continuously learn and improve
- Improved Quality: Consistent feedback and reviews can improve quality of product
- Teamwork: Strong collaboration and communication help to create an efficient and strong team.
- Delaying Commitment: Waiting to make irreversible decisions until all tests are completed and improvements are made.
- Optimizing System as a Whole: Issues are broken down allowing for optimal and efficient workflows. This results in stronger performance and quality.

The Agile Manifesto

The Agile Manifesto was created in 2001 and includes 4 Agile Values and 12 Agile principles. This document was created to guide adaptive and customer focused software development. It emphasizes collaboration, customer feedback and satisfaction, and adaptability. It allows developers and processes to become more flexible and quickly react to changes.

Values of Agile Manifesto:

- Individuals and Interactions over Processes and Tools: Enhances effective communication and collaboration
- Working Software over Comprehensive Documentation: Focuses on functional software delivery as a progress measurement
- Customer Collaboration over Contract Negotiation: Allows customers and stakeholders to be involved during processes
- Responding to Change over Following a Plan: Embracing flexibility and adapting to change throughout all steps of the process.

Principles of Agile Manifesto:

- Customer Satisfaction through early and continuous delivery
- Allow changing requirements through all stages of development
- Delivering working software continuously
- Collaboration between stakeholders and developers
- Building projects around motivated individuals
- Face-to-face communication is more effective
- Working software as measurement of progress
- Maintaining sustainable work pace
- Attention to good technique and design
- Simplicity – Not doing unnecessary work
- Self-Organizing teams
- Regular reflections for effectiveness and improvements

Continuous Delivery Movement

The Continuous Delivery Movement is a development process that automatically prepares changes and applies continuous integration. In this process, developers can automate certain tests to verify software changes and updates before they are released. This process is beneficial because it allows for quick and simple releases, easier maintenance, better quality, and less downtime. The emphasis of continuous delivery is on automation and requires minimal manual effort to implement and deploy changes.

Resources

Odazie, D. (2024, August 5). *A brief history of DevOps and its impact on software development*. EverythingDevOps. https://everythingdevops.dev/a-brief-history-of-devops-and-its-impact-on-software-development/

GeeksforGeeks. (2024b, July 1). *Lean Software Development (LSD)*. https://www.geeksforgeeks.org/lean-software-development-lsd/

GeeksforGeeks. (2024c, August 2). *Manifesto for Agile Software Development*. https://www.geeksforgeeks.org/agile-manifesto-for-software-development/

Continuous delivery: Basic concepts and the Gitops Revolution. Codefresh. (2024, August 1). https://codefresh.io/learn/continuous-delivery/