UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

Topic:

Introduccion a DevOps

By:

Argüelles Gálvez Antonio

Group:

10B

Matter:

Software Development Process Management

Teacher:

Ray Brunett Parra Galaviz

Date:

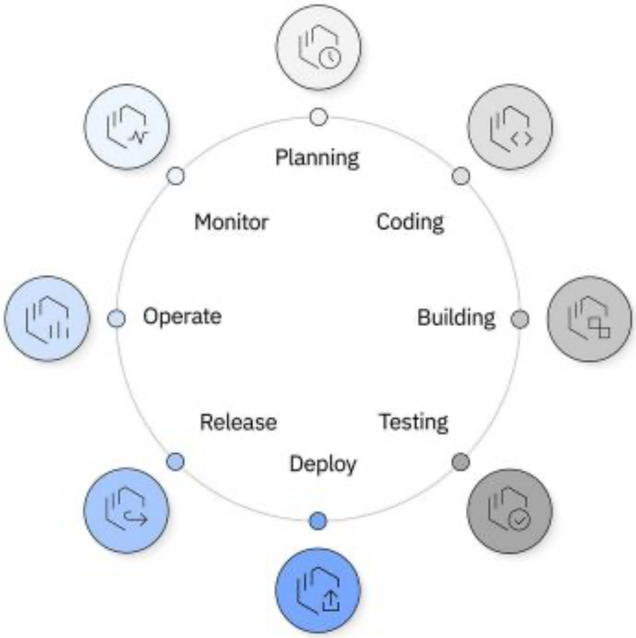
01/08/2025

What is?

DevOps is a software development methodology that accelerates the delivery of higher quality applications and services by combining and automating the work of software development and IT operations teams.

Therefore, this methodology is defined as describing a software development process and organizational culture change that fosters coordination and collaboration between development and IT operations teams.

DevOps Lifecycle



Planning

In this part of the process developers code and create new and enhanced features based on user stories and backlog work items. A combination of practices such as test-driven development (TDD), pair programming and peer-to-peer code reviews is common. Developers often use their local workstations to perform the internal cycle of writing and testing code before sending it down the Delivery Pipeline.

Programming

In this part of the process developers code and create new and enhanced features based on user stories and backlog work items. A combination of practices such as test-driven development (TDD), pair programming and peer-to-peer code reviews is common. Developers often use their local workstations to perform the internal cycle of writing and testing code before sending it down the Delivery Pipeline.

Creating continuous integration and delivery

In this workflow, new code is integrated into the existing code base and then tested and packaged for release and deployment. Typical automation activities include merging code changes into a master copy, checking in that code from a source code repository, and automating compilation, unit testing and packaging into an executable file. Best practice is to store the result of the CI phase in a binary repository for the next phase.

Testing

Teams use testing, often automated, to ensure that the application meets standards and requirements. The traditional DevOps approach includes a discrete testing phase that occurs between build and release.

However, DevOps has evolved to the point where certain testing elements can take place during planning (behavior-driven development), development (unit tests, contract tests), integration (static code analysis, CVE analysis), deployment (smoke tests, penetration tests, configuration tests), operations (chaos testing, compliance testing), and learning (A/B testing).

Deployment

Deployment is when the project is moved to a production environment where users can access changes in the application. The infrastructure is set up and configured (often using infrastructure as code), and the application code is deployed. A best practice for production deployment is to first release to a subset of end users and eventually roll it out to all users once stability is confirmed.

Release

The first stage of operations, the release stage, is the last step before users gain access to the application. In this workflow, the runtime build output (from integration) is deployed to a runtime environment, typically a development environment where runtime tests are conducted to determine quality, compliance, and security.

Operation

If feature delivery in a production environment is considered “Day 1,” then once the features are running in production, “Day 2” operations begin. Monitoring performance, behavior, and feature availability ensures that the features provide value to users.

In this phase, teams verify that the features work correctly and that there are no service disruptions, ensuring that network, storage, platform, computing, and security conditions remain stable. If issues arise, operations teams identify the incident, alert the appropriate personnel, troubleshoot the problems, and implement fixes.

Monitoring

This involves collecting user and customer feedback on features, functionality, performance, and business value to plan improvements and new features for the next release. It also includes any learning and outstanding elements from operational activities that can help developers proactively prevent known incidents from recurring. This is where the "wrap-around" of the planning phase occurs, driving continuous improvement.

Advantages

- **Faster problem resolution and delivery:** Allows teams to iterate, test, troubleshoot, and deliver quality builds consistently and more efficiently.

- **Encourages collaboration and trust:** Primarily through its focus on teamwork and a shared sense of responsibility among teams. The process fosters a positive, open, and transparent environment, the establishment of organizational goals, and the implementation of unified tools and processes.

- **Significant cost reduction:** Helps drive continuous improvement through early and frequent implementation, followed by the integration of feedback loops at each step of the DevOps lifecycle. This helps optimize performance, reduce costs, and increase delivery speed.

- **Prioritizes customer satisfaction:** Provides the flexibility to incorporate customer feedback into plans more easily and ensure that the product aligns with their desires and needs.

Conclusion

DevOps focuses on optimizing software development and operations through automation and continuous integration, fostering team collaboration to accelerate the delivery of high-quality products. Throughout its lifecycle, from planning to monitoring, it enhances efficiency, reduces costs, and responds quickly to market needs. Its implementation not only streamlines processes but also strengthens an organizational culture based on transparency, trust, and continuous improvement, making DevOps an essential approach for competitiveness in the technology sector.

References

- ¿Qué es DevOps? Explicación de DevOps | Microsoft Azure. (s. f.). https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops#:~:text=Definici%C3%B3n%20de%20DevOps,los%20clientes%20de%20forma%20constante.

- Ibm. (2024, 4 noviembre). DevOps. *¿Qué es DevOps?* https://www.ibm.com/mx-es/topics/devops

- Beneficios y ventajas de implementar DevOps | Unidad. (s. f.). Unity. https://unity.com/es/topics/benefits-of-devops