Software Engineering

Dr. Meisam Nazariani

Email: m_nazariani@aut.ac.ir

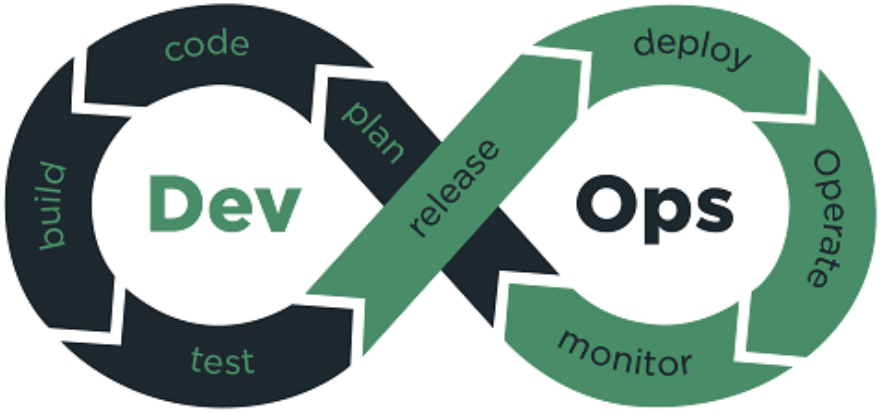
February 2022

Outline

1. Introduction
2. The Nature of Software
3. Software Engineering
4. The Software Process
5. Process Models
6. Agile Development
    1. XP
    2. Scrum
7. **DevOps**
8. Requirement Engineering
9. Software Modeling
10. Design Concepts
11. Umbrella Activities
12. Case Studies

DevOps

DevOps

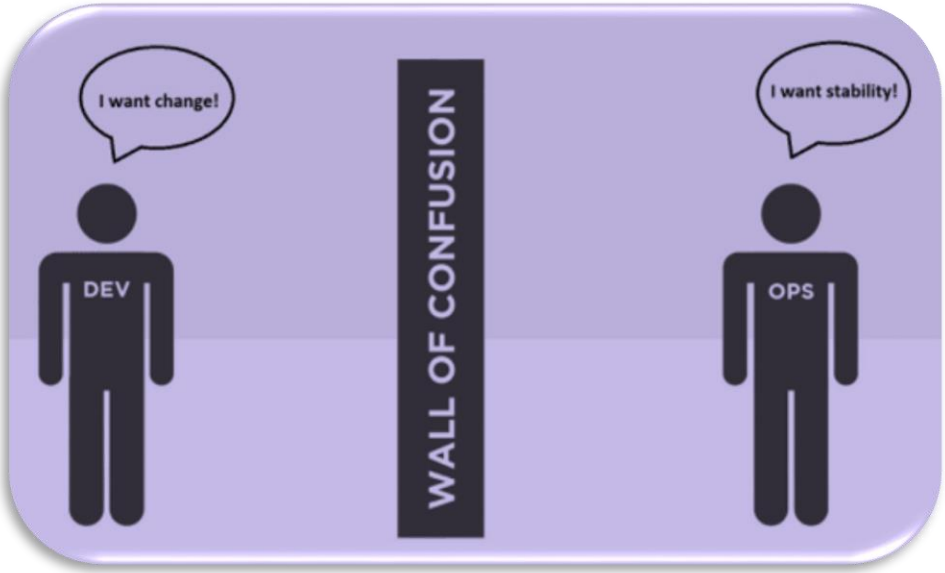


What is it?

DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams.

It emphasizes team empowerment, cross-team communication and collaboration, and technology automation.

DevOps



When did it start?

The DevOps movement began around 2007 when the software development and IT operations communities raised concerns about the traditional software development model, where developers who wrote code worked apart from operations who deployed and supported the code.

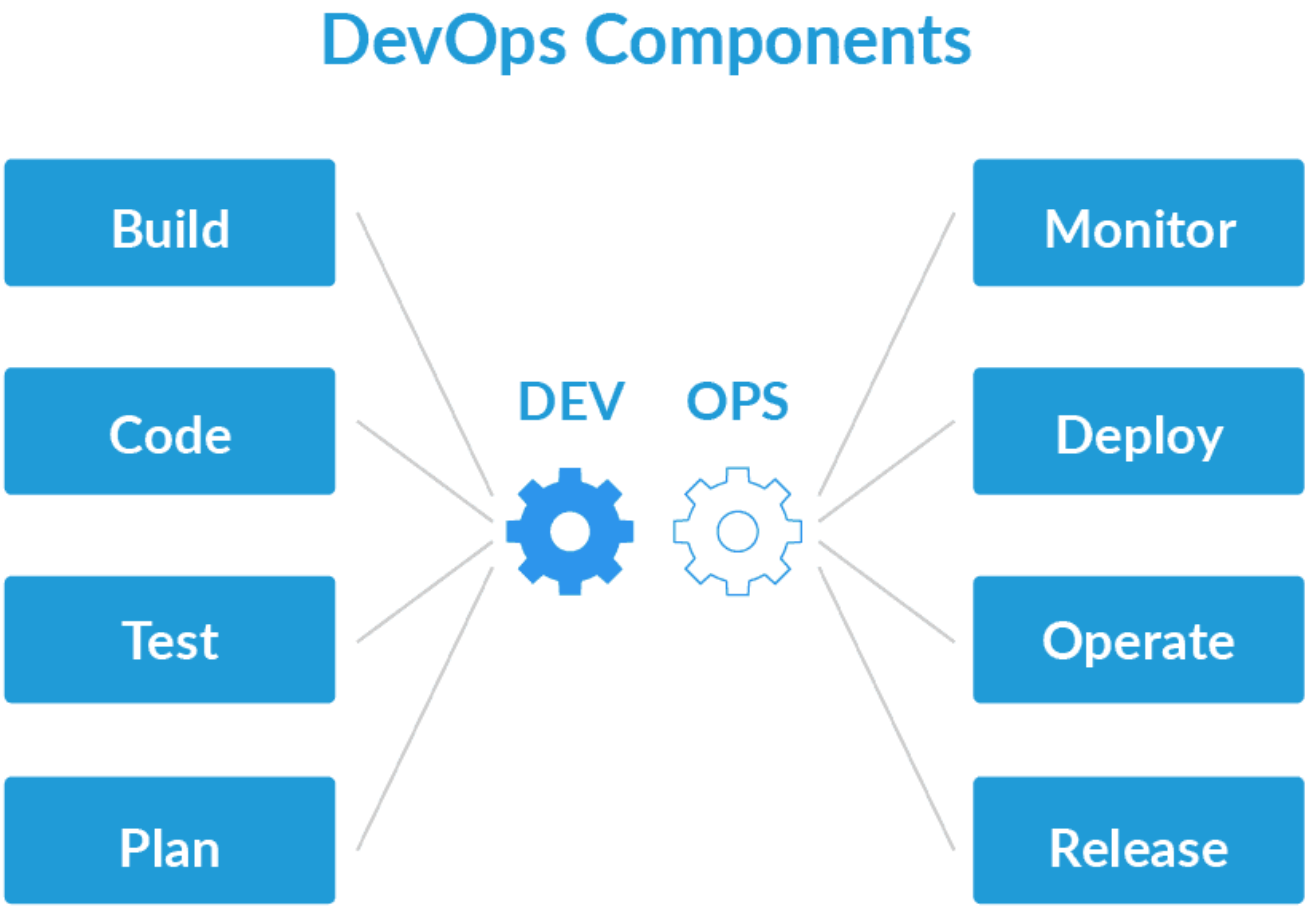
The term DevOps, a combination of the words development and operations, reflects the process of integrating these disciplines into one, continuous process.

DevOps

How does DevOps work?

- A DevOps team includes developers and IT operations working collaboratively throughout the product lifecycle, in order to increase the speed and quality of software deployment.
- Under a DevOps model, development and operations teams are no longer “siloed” Sometimes, these two teams merge into a single team where the engineers work across the entire application lifecycle from development and test to deployment and operations and have a range of multidisciplinary skills.
- DevOps teams use tools to automate and accelerate processes, which helps to increase reliability. A DevOps toolchain helps teams tackle important DevOps fundamentals including continuous integration, continuous delivery, automation, and collaboration.

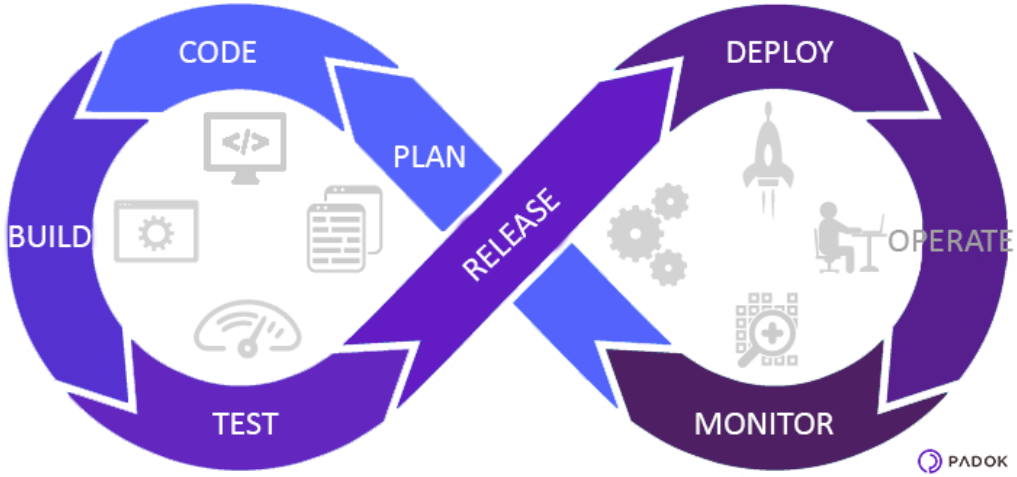
DevOps


DevOps Components
Build
Code
Test
Plan
DEV
OPS
Monitor
Deploy
Operate
Release

DevOps lifecycle

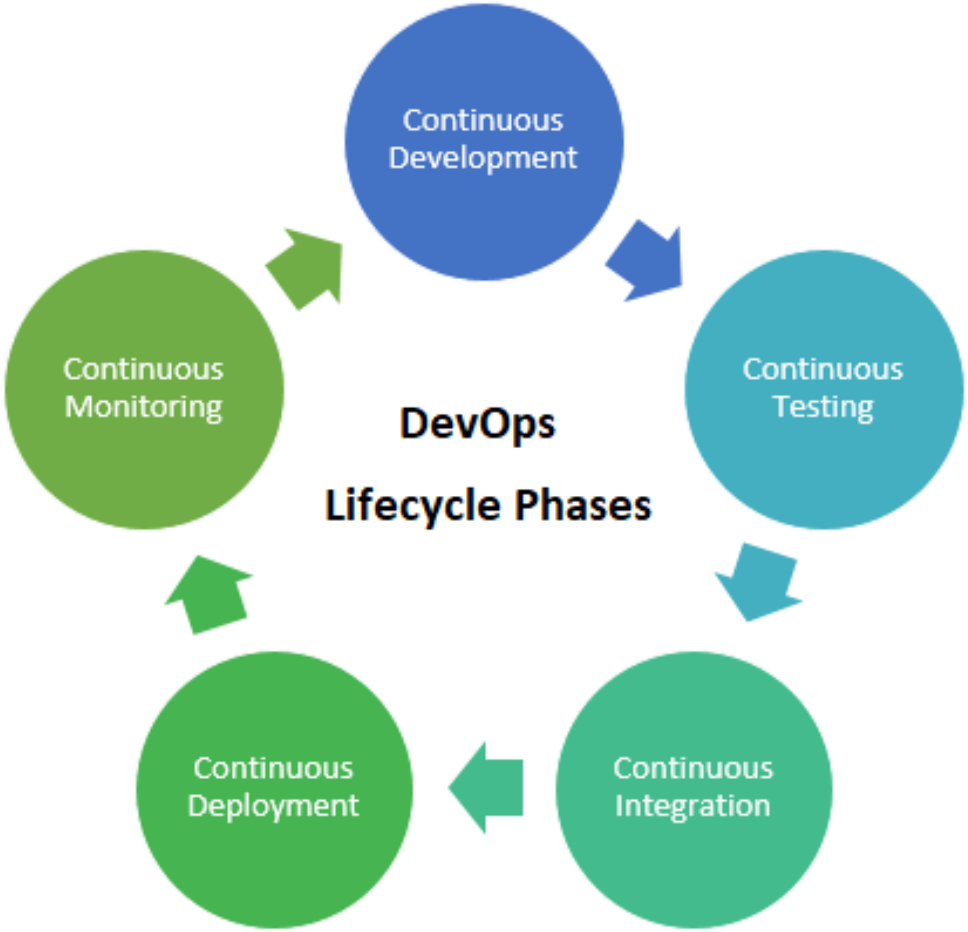
- Because of the continuous nature of DevOps, practitioners use the infinity loop to show how the phases of the DevOps lifecycle relate to each other. Despite appearing to flow sequentially, the loop symbolizes the need for constant collaboration and iterative improvement throughout the entire lifecycle.

- The DevOps lifecycle phases:
  - processes, capabilities, and tools needed for development (left side)
  - operations (right side)



DevOps practices

1. **Continuous Development**
2. **Continuous Testing**
3. **Continuous Integration**
4. **Continuous Deployment**
5. **Continuous Monitoring**



What are the benefits of DevOps?

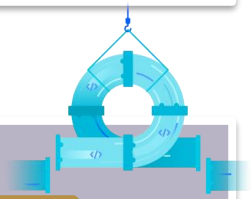
Speed

Teams that practice DevOps release deliverables more frequently, with higher quality and stability.

*208 times more frequently and 106 times faster than low-performing teams

Improved collaboration

DevOps makes teams more efficient and saves time related to work handoffs and creating code that is designed for the environment where it runs.



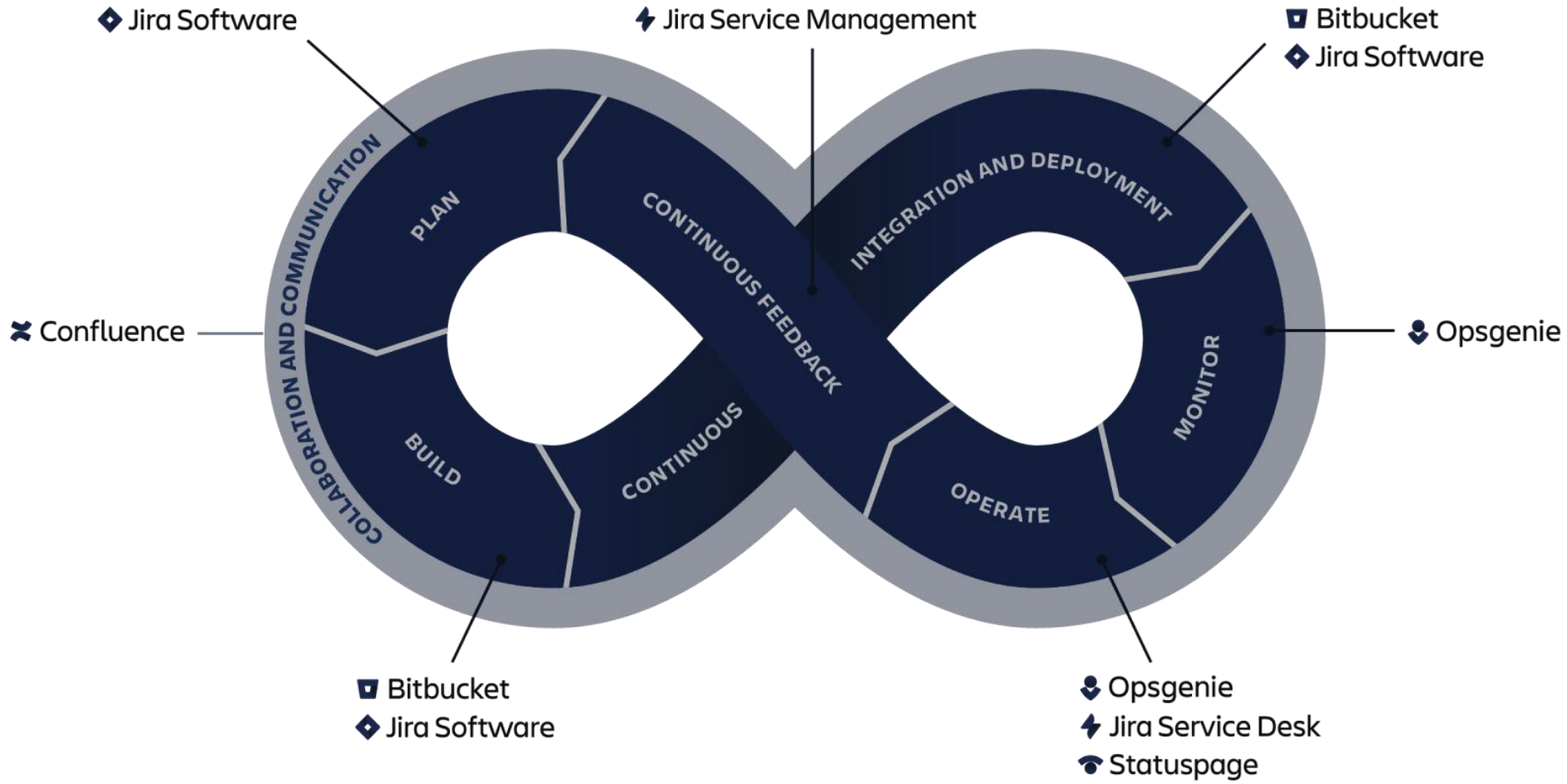
Quality and reliability

Practices like continuous integration and continuous delivery ensure changes are functional and safe, which improves the quality of a software product.

Security

By integrating security into a continuous integration, continuous delivery, and continuous deployment pipeline, DevSecOps is an active, integrated part of the development process.

DevOps tools



Agile Vs. DevOps: What's the difference?