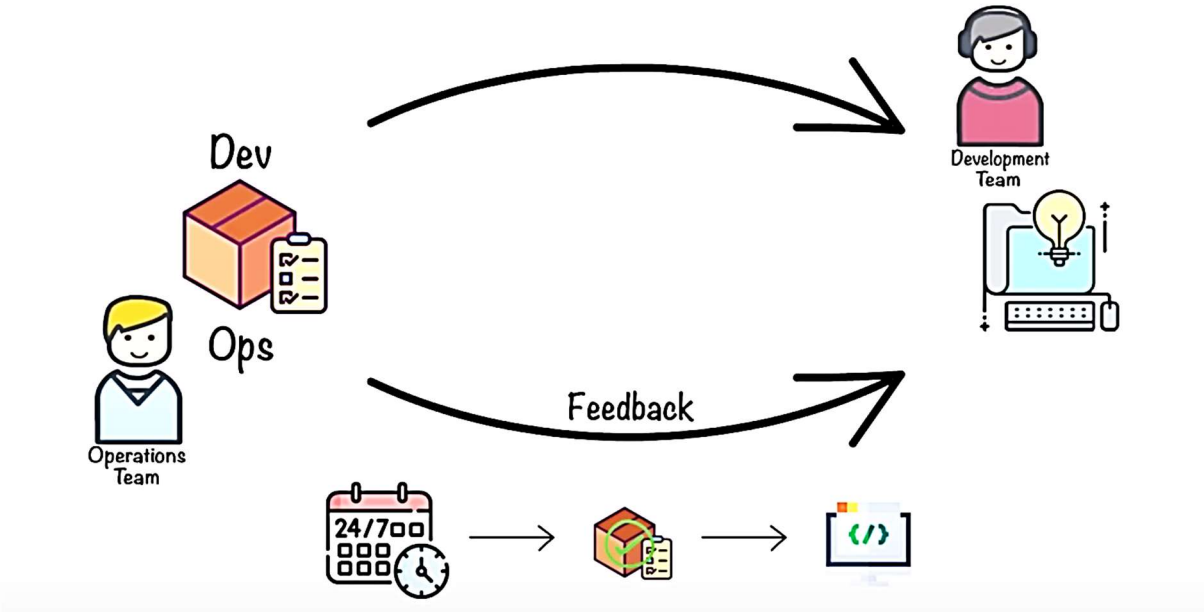
DevOps Life Cycle

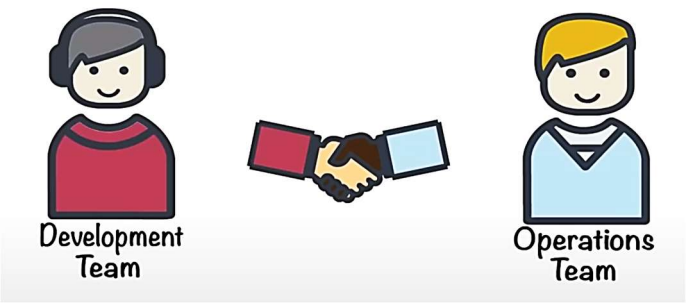
The software development comprised of two different departments.

- The **Development team** that *develops the plan, designs and builds* the system from scratch and
- The **Operation team** for testing the implementation of whatever is developed by the development team.

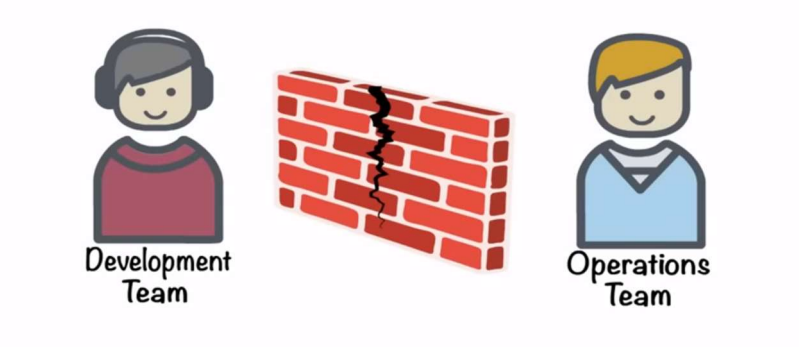
The operations team give the development team any feedback or about the bugs or any rework required. The development team would be idle awaiting feedback from the operations team, undoubtedly extends the timelines and may get delay.



The entire software development cycle, there would be instances where the development team moves on to the next project while the operations team continues to provide feedbacks for the previous code, and this meant weeks or even months for the project to be closed and final code to be developed.

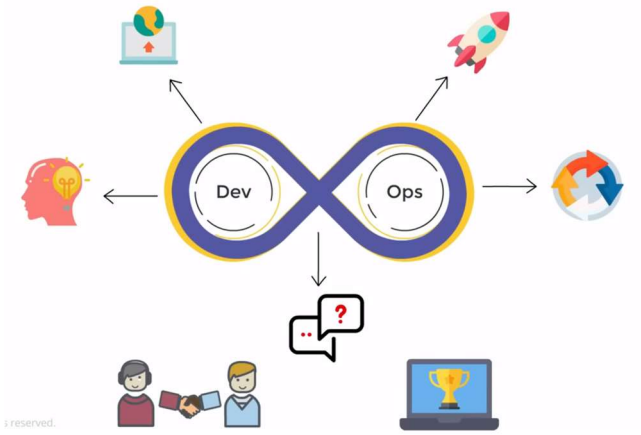


Now! What if both the teams come together and worked in collaboration with each other and the wall of confusion is broken.



This is called the DevOps approach. The DevOps symbol resembles an infinity sign. Suggesting that it is a continuous activity.

What does DevOps do?



- The DevOps approach makes companies *adapt faster to updates and development changes*.
- The teams can now *deliver more quickly*.
- The *deployments are more consistent*.
- The *deployments are smooth and continuous*.

- There may be communication challenges, and the DevOps managers can bring a stream flow between the teams and makes the software delivery process a success.

The DevOps culture is implemented in several phases using several automated tools.
The phases are:

1) **PLAN:** The first phase is the **Planning phase**, where the development team puts down a plan, keeping in mind the application objectives that are to be delivered to the customer.

2) **CODE:** After the planning is done, the Coding begins. The development team works on the same code and different versions of the code or stored into a *repository* with the help of tools like ***git*** and merge when required. This process is called *version control*.

3) **BUILD:** The code is then made executable with tools, *like Maven, and Gradle* in the build state.

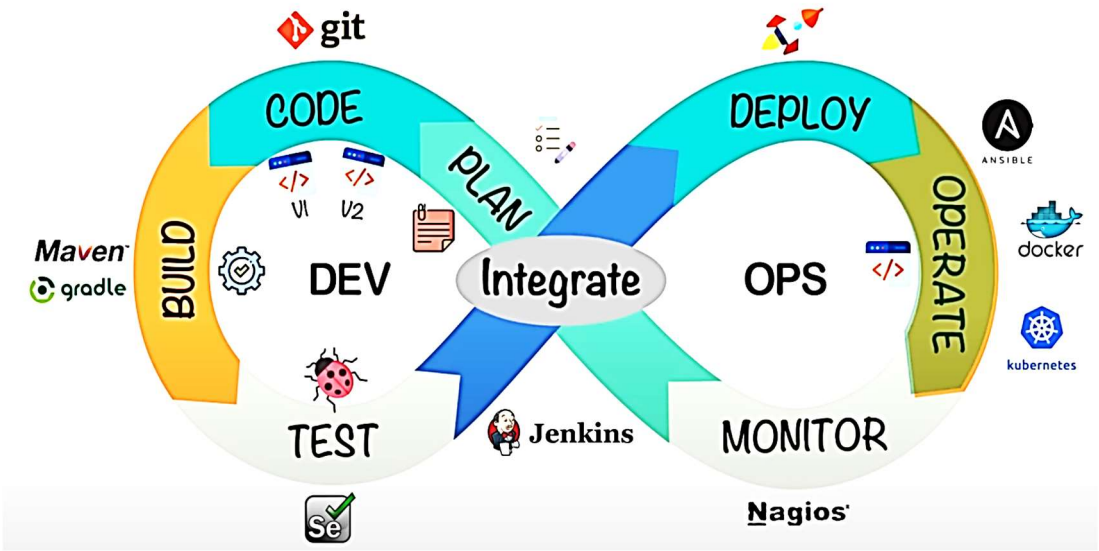
4) **TEST:** After the code is successfully built tested for any bugs or errors. The most popular tool for automation testing is *selenium*.

5) **DEPLOY:** Once the code has completed several automated tests, its ready for deployment and is sent to the operation team. The operations team now Deploys, the code to the working environment. The most prominent tools used to automate include *Ansible, docker, and Kubernetes*.

6) **OPERATIONS:** After the deployment, the product is continuously monitored and Nagios is one of the top tools Used to automate this phase.

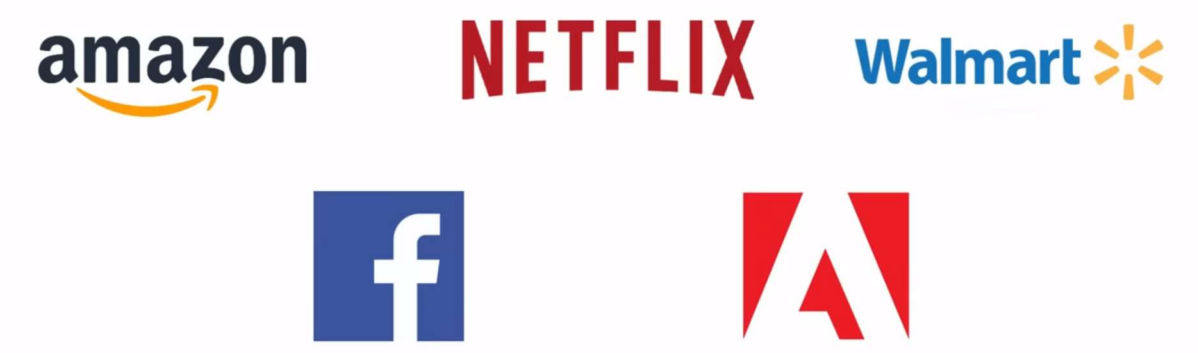
7) **MONITOR:** The feedback received at this phase is send to the development team, which forms the core of the *DevOps life cycle*, that is the ***integration phase***. *Jenkins* is the tool that sends the code for building and testing. If the code passes the test, it is sent for deployment. And this is referred to as *continuous integration*.

The DevOps culture is implemented in several phases with the help of several tools

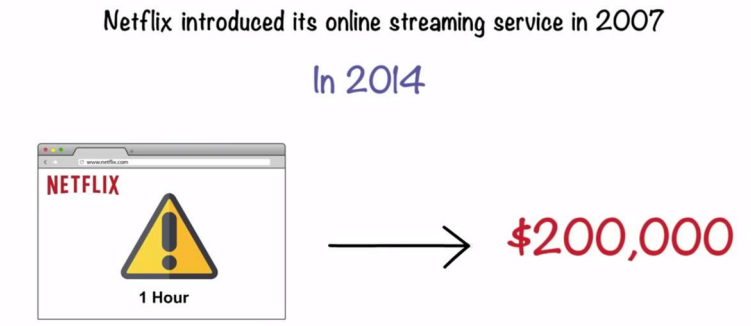


There are many tech giants and organizations that have opted for the DevOps approach.

For example, Amazon Netflix Walmart Facebook and Adobe.



Netflix introduced in 2007, and in 2014 it was estimated that a downtime for would cost Netflix $200,000.



However, now Netflix can cope with such issues. They opted for DevOps in the Netflix and developed a tool called the *Simian Army* that mind without affecting. The is chaos motivated the developers to build a system that does not fall apart when any such thing happens.