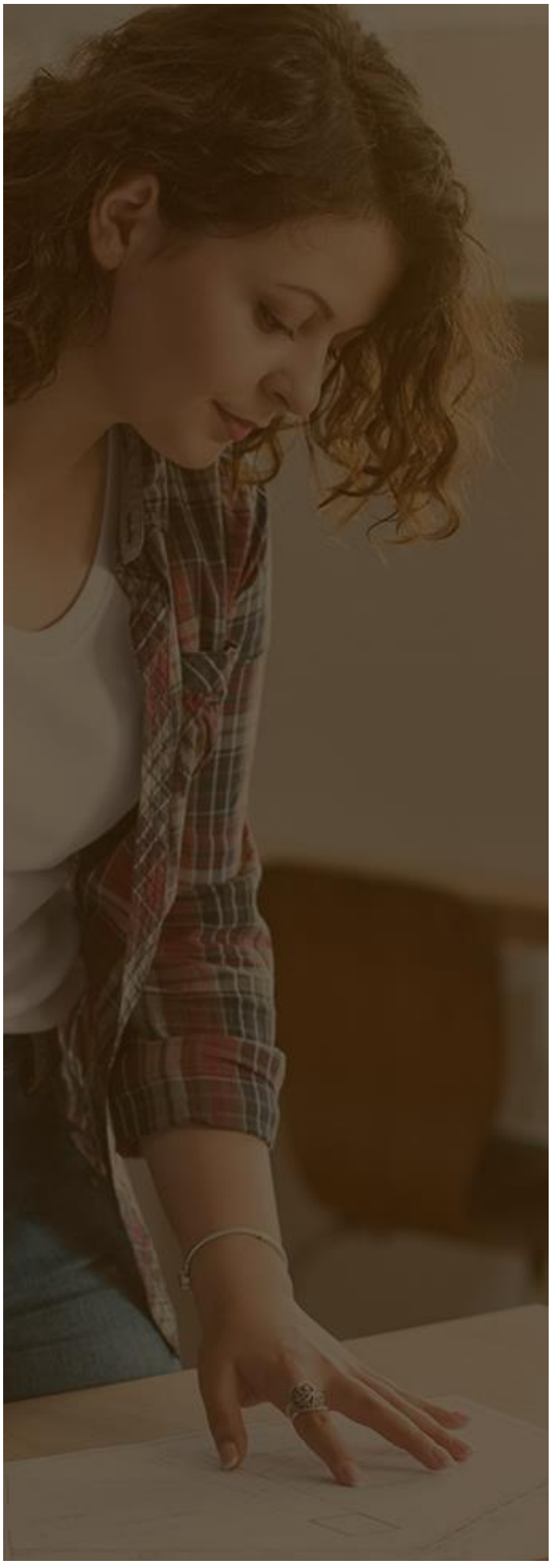
Introduction to DevOps

by

Hafsa Aleem

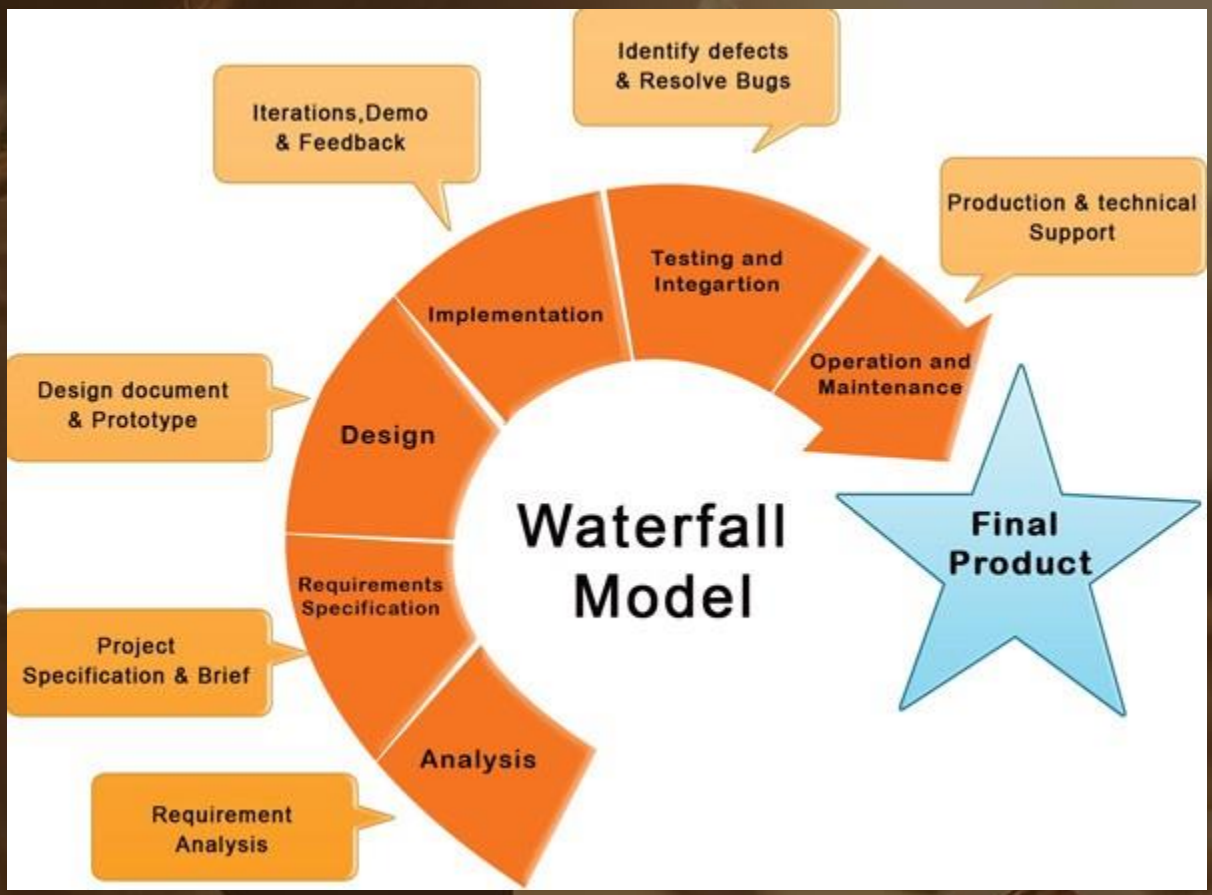


Out of keen interest in DevOps, I thought of coming up with a series of blogs that will educate you about the new culture being adopted in Software Development and help you understand what it is all about.

The DevOps Tutorial

This DevOps tutorial blog series will familiarize you with DevOps methodology & industry-wide used tools, required for DevOps Certification. In this blog, I will take you through the following things, which will be the base of the upcoming blogs:

WATERFALL MODEL



By Hafsa Aleem

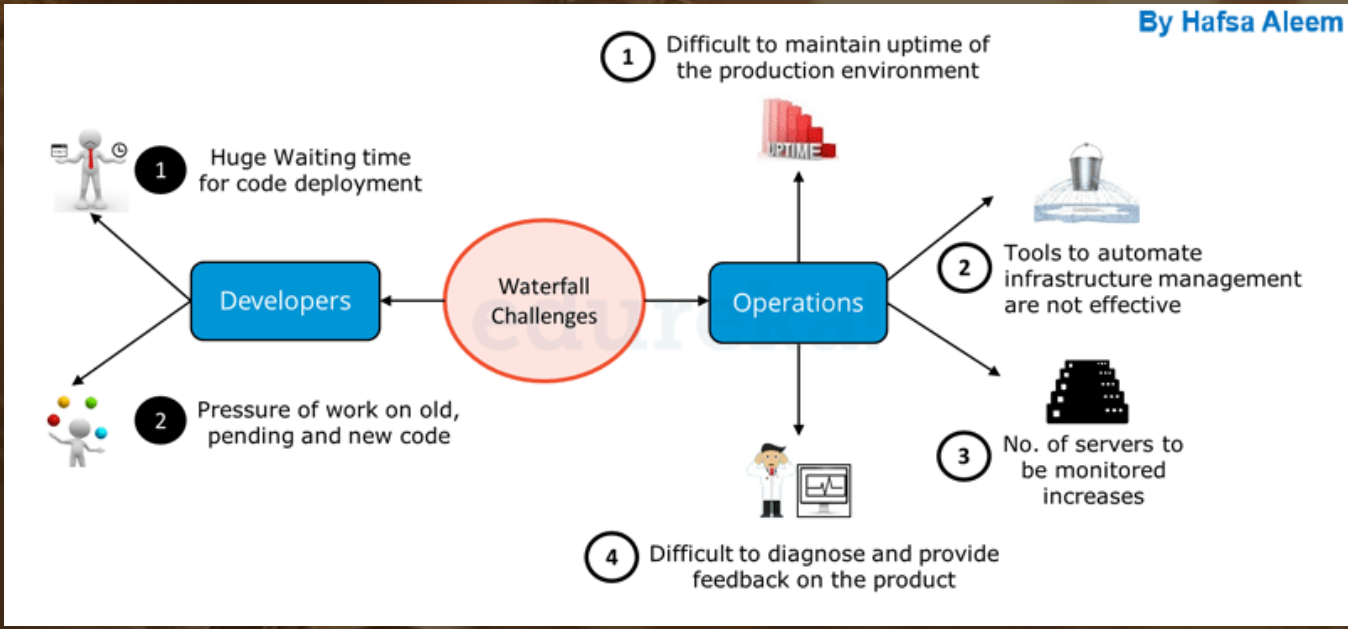
Let’s consider developing software in a traditional way using a Waterfall Model.

In the above diagram you will see the phases it will involve:

- In phase 1 – Complete Requirement is gathered, and SRS is developed
- In phase 2 – This System is Planned and Designed using the SRS
- In phase 3 – Implementation of the System takes place
- In phase 4 – System is tested, and its quality is assured
- In phase 5 – System is deployed to the end users
- In phase 6 – Regular Maintenance of the system is done

Waterfall Model Challenges

The Water-fall model worked fine and served well for many years however it had some challenges. In the following diagram the challenges of Waterfall Model are highlighted.



In the above diagram you can see that both Development and Operations had challenges in the Waterfall Model. From Developers point of view there were majorly two challenges:

1

After Development, the code deployment time was huge.

2

Pressure of work on old, pending and new code was high because development and deployment time was high.

On the other hand, Operations was also not completely satisfied. There were four major challenges they faced as per the above diagram:

1

It was difficult to maintain ~100% uptime of the production environment.

2

Infrastructure Automation tools were not very affective.

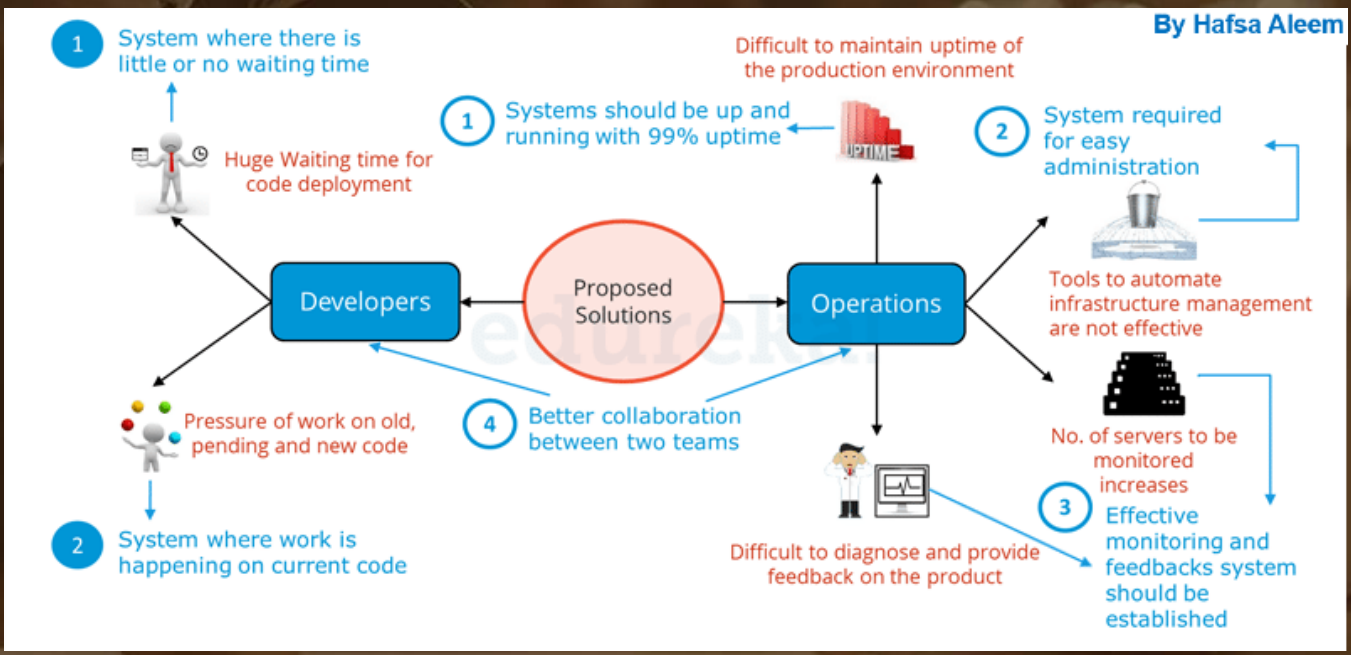
3

Number of severs to be monitored keeps on increasing with time and hence the complexity.

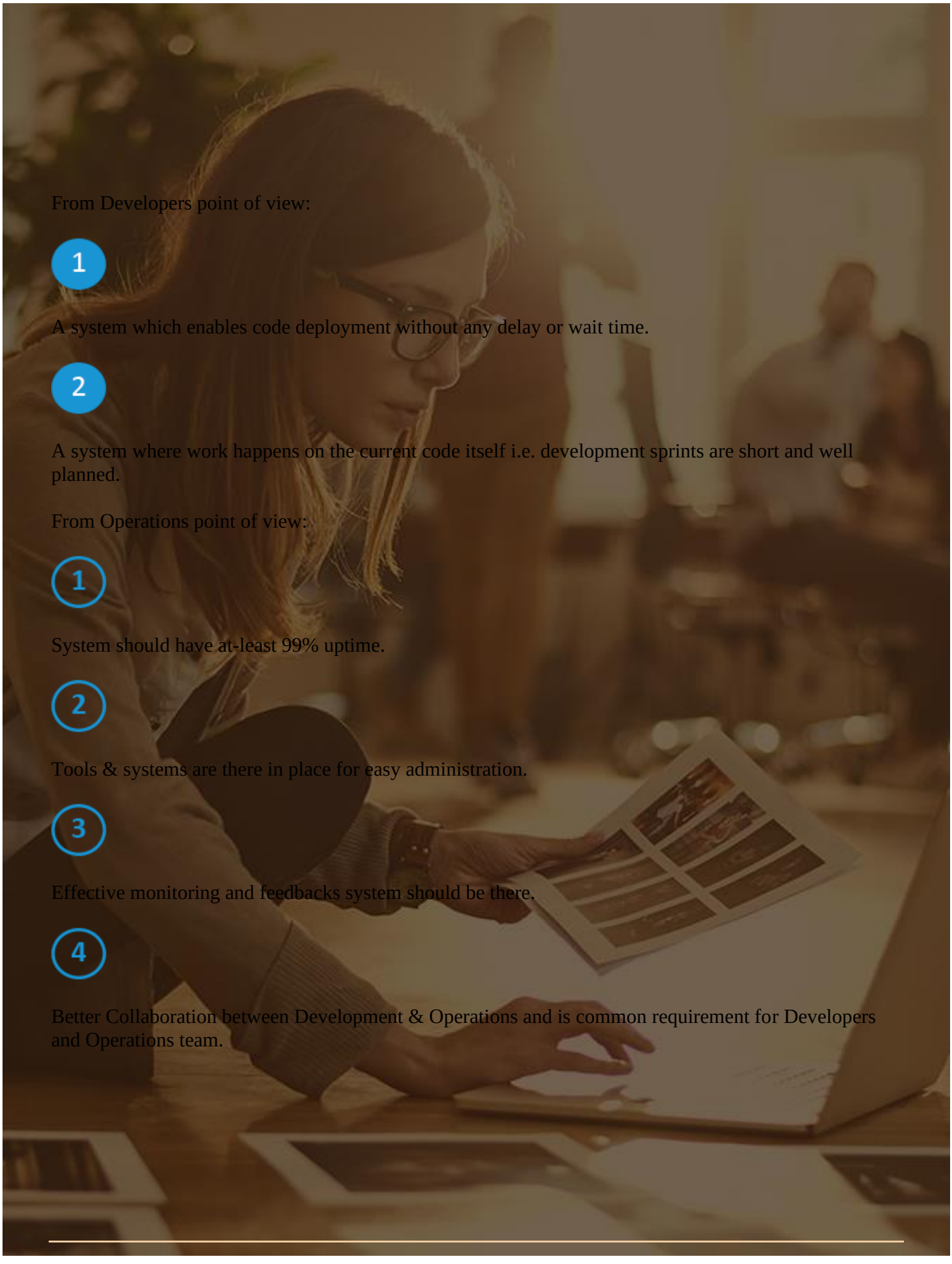
4

It was very difficult to provide feedback and diagnose issue in the product.

In the following diagram proposed solution to the challenges of Waterfall Model are highlighted.



In the above diagram, Probable Solutions for the issues faced by Developers and Operations are highlighted in blue. This sets the guidelines for an Ideal Software Development strateg



From Developers point of view:

1

A system which enables code deployment without any delay or wait time.

2

A system where work happens on the current code itself i.e. development sprints are short and well planned.

From Operations point of view:

1

System should have at-least 99% uptime.

2

Tools & systems are there in place for easy administration.

3

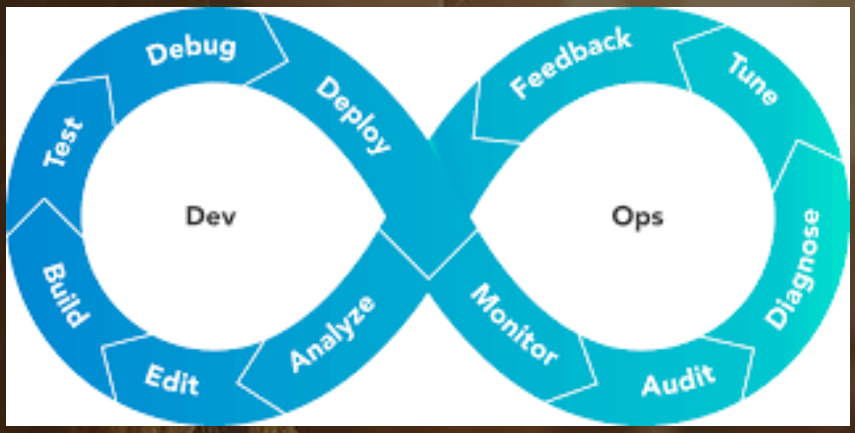
Effective monitoring and feedbacks system should be there.

4

Better Collaboration between Development & Operations and is common requirement for Developers and Operations team.

What is DevOps? | DevOps Introduction & Tools | DevOps Tutorial |

DevOps integrates developers and operations team to improve collaboration and productivity.



According to the DevOps culture, a single group of Engineers (developers, system admins, QA's. Testers etc turned into DevOps Engineers) has end to end responsibility of the Application (Software) right from gathering the requirement to development, to testing, to infrastructure deployment, to application deployment and finally monitoring & gathering feedback from the end users, then again implementing the changes.

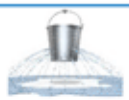
This is a never ending cycle and the logo of DevOps makes perfect sense to me. Just look at the above diagram – What could have been a better symbol than infinity to symbolize DevOps?

Now let us see how DevOps takes care of the challenges faced by Development and Operations. Below table describes how DevOps addresses Dev Challenges.

edureka!	Dev Challenges	DevOps Solution
	Waiting time for code deployment	• Continuous Integration ensures there is quick deployment of code, faster testing and speedy feedback mechanism • Thus there is no waiting time to deploy the code. Hence the developer focuses on building the current code
	Pressure of work on old, pending and new code	

DevOps Tutorial Table 1 – Above table states how DevOps solves Dev Challenges

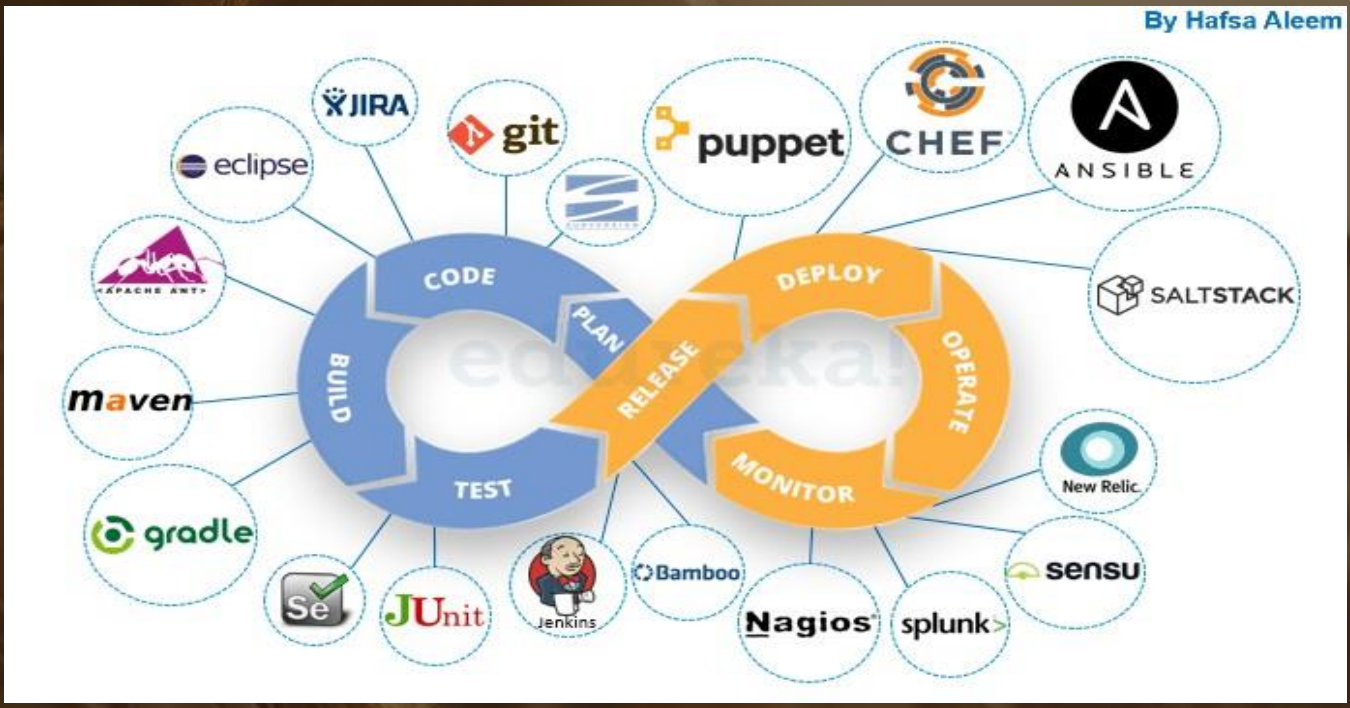
Going further, below table describes how DevOps addresses Ops Challenges.

	Ops Challenges	DevOps Solution
	Difficult to maintain uptime of the production environment	Containerization / Virtualization ensures there is a simulated environment created to run the software as containers offer great reliability for service uptime
	Tools to automate infrastructure management are not effective	Configuration Management helps you to organize and execute configuration plans, consistently provision the system, and proactively manage their infrastructure
	No. of servers to be monitored increases	Continuous Monitoring Effective monitoring and feedbacks system is established through Nagios Thus effective administration is assured By Hafsa Aleem
	Difficult to diagnose and provide feedback on the product	

DevOps Tutorial Table 2 – Above table states how DevOps solves Ops Challenges

However, you would still be wondering, how to implement DevOps. To expedite and actualize DevOps process apart from culturally accepting it, one also needs various DevOps tools like Puppet, Jenkins, GIT, Chef, Docker, Selenium, AWS etc to achieve automation at various stages which helps in achieving Continuous Development, Continuous Integration, Continuous Testing, Continuous Deployment, Continuous Monitoring to deliver a quality software to the customer at a very fast pace.

Now take a look at the below DevOps diagram with various DevOps Tools closely and try to decode it.



These tools have been categorized into various stages of DevOps. Hence it is important that I first tell you about DevOps stages and then talk more about DevOps Tools.

- Continuous Development
- Continuous Integration
- Continuous Testing
- Continuous Monitoring
- Virtualization and Containerization

These stages are the building blocks to achieve DevOps as a whole.

This is the end of the first blog of – The DevOps Tutorial Series