UnderStanding basics of Devops

# Heading

In this era of IT industry, a buzz word is Dev-Ops. Many people are talking about this. Every organization try to adopt the same.

***But the questions pop up in my mind are***

1. ***What is DevOps?***
2. ***Why I shall use this?***
3. ***What are the problems with legacy software management?***
4. ***Is Devops is necessity or Ornamental?***

To answer this questions, I am trying to look back the typical mode of Software management

In a typical Software development cycle, we do the following tasks

1. Gather requirements from client.
2. Doing some HLD, LLD blah blah.
3. Create a Version control system(VCS) to maintain the code base.
4. Developer implement code in their machine.
5. Run unit tests on that implementation.
6. Commit the code into Version control system.
7. Developer raise a ticket to Infrastructure team to build and deploy the code in Testing environment.
8. Infrastructure team deploy the code in to Test environment
9. Tester test the code according to test cases.
10. Raise a ticket to Infrastructure team for deploy in SIT environment
11. Infrastructure team deploy the code in to SIT environment
12. Tester do the Integration testing.
13. If previous steps are successful we are ready for deployment in to production.
14. We setup a meeting with client and fixed a date for deployment into production.
15. Infrastructure team ready to deploy the product on Production.
16. Infrastructure team complete the deployment and send status to team about the status of deployment.
17. In production if any post production bug raises we follow the same steps 1-15.

Oops!!! it’s a long step isn’t it?

What I understood from that steps are

1., ***many cross functional team is involved in the cycle***.

2. **Developers needs an environment to work seamlessly.**

3**. As per previous project model (waterfall, Iterative) a product will be delivered to Production after a long time from the gathering requirement steps.**

So according to the Pre- Devops days the pain points are

**Communication**: As I said earlier there many ***many cross functional team is involved in the cycle*** so hand shaking between the inevitable. And this is the point where we are blocked (2-3 days approx).

Let’s take an Example

Suppose a developer develops a feature and do the unit test. After successful unit test he/she commit the code to VCS. Then Developer wants to test that code by tester according to test scripts so he has to mail it or in a bug tracking system developer updates the status and assign to tester for test. But tester currently testing the Other feature so developer has to wait for conformation

*here the* ***Block state starts****.*

*Developer then picks up another feature from his plate. After 2-3 days Tester test his implementation and return to developer as there is a defect. So Now developer busy with new requirement that bug waits for developer account.*

*This is the typical scenario we work. If I search for actual problem I found that this is due to the process and the chaos is created as there is cross functional team involves and they highly dependent on each other.*

Another scenario you can consider suppose a bug is found developer inspect it and found it is due to

Memory space so to solve the problem we need to add a server in to the server pool. Then developer contract to IT-Operations team. They said raise a ticket and ticket is assign to operation. Once it is approved by Operations head operation team add the server in to the pool. Same thing here to solve a minimum problem we have to wait for 2-3 days not only that for a minimum build or deploy each time you need to raise a ticket***.***

***Developer, Operations or Testing team does not take order without ticket. Because their performance count on that ticket ☺.***

***So, Process hinder us to make smooth delivery.***

**Infrastructure**:

Infrastructure is another pain point; in a typical project I have seen developer are work in VDI. Not only that the biggest problem is that developers are work in Windows system but in production or SIT environment is on Linux. So developers machine is not a replica of production server so developer not confident about their code silly things can happen

Like suppose you have property file where you mention the path where you need to place you upload files as developer system is different from production developer set

This path to as per local home which is windows home. And forget to revert it when commit the files.

In SIT environment suddenly tester discovered upload functionality fails due to wrong home directory.

Not only that suppose developers VdI m/c has been changed so in new VDI m/c he has to setup his whole project which takes just 1-2 days to set up dependency and run in local server.

**Production Release**:

Another problem is waterfall or iterative approach takes a long time to release a project /product to the clients. So it can be happened when you think about the project that is a unique idea but delay to release the product another competitor thinks and release the Idea before you. So you are in a losing end just foe the process.

Due to these pain points **DevOps** rise and try to rescue us from this apocalypse.

***DevOps is culture which promotes a continuous delivery culture. Or I can say it promote a delivery pipe line concept where from developer commits a code to production release should be in a pipeline***

***And fully automated No human intervention needed.***

There is subtle difference between Continuous Integration, Continuous deployment and Continuous delivery. I will discuss them on an another article.

For now, We can consider Devops dissolve the problem of “***cross functional team involvement problem” .***

***Devops is like conveyer belt and consists of multiple tools it takes care of all the steps I mentioned earlier step 3-15.***

***Devops also take care of orchestration and make sure developers environment should be same as Production. By Puppet, Chef it can be archived,***

***Docker is also another container management tools which works very fine in context of Devops.***

***Agile methodology is an integral part of Devops so we can build a minimal viable product and show to client and based on client feedback we take the next steps.***

***It is like don’t go for big target brake down it to feasible small targets the achive it***

***The Best Divide and conquer rule.***

***Now the last question Devops is necessity or Ornamental***

It is totally based on your product if your product is simple an won’t change very frequently the cost to implement Devops will be high on Other hand If your product is complex and very costly I personally think go for Devops.