ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

From Dev to Ops – Introduction to Devops on understanding Continuous Integration and Continuous Delivery

A. Remi Jullian[1], M.Sangeetha[2]

PPSM Team Lead, Accenture Services Pvt Ltd., Shriram IT Gateway, Perungalathur, Chennai, TamilNadu, India

ABSTRACT: DevOps (short for development and operations), like most new methodologies, is only a catchword for many individuals working in IT industry. Everyone debates about it, but not everyone knows what it is. In wide terms, DevOps is an approach based on lean and agile principles in which business holders and the development, operations, and quality assurance sectors, work together to deliver software in a continuous manner that enables the business to more quickly seize market opportunities and reduce the time to include customer feedback. The proposed journal provides a clear understanding of Devops and the concepts revolving around it.

KEYWORDS: DevOps; Continuous Integration; Continuous Delivery; Continuous Deployment; Orchestration; Delivery Automation; Pipeline; Agile

I. UNDERSTANDING DEVOPS

Patrick Debois, who’s often called “the father of DevOps”, devised the name “DevOps” in 2009. As the word describes, it was formed by coalescing two words: “development” and “operations”. DevOps is a collaborative way of developing and deploying software. DevOps (a portmanteau of development and operations) is a software development method that stresses communication, association and amalgamation between software developers and information technology (IT) operation professionals.

It’s a movement of people who think it’s change in the IT Industry - time to stop wasting money, time to start delivering great software, and building systems that scale and last” – Patrick Debois

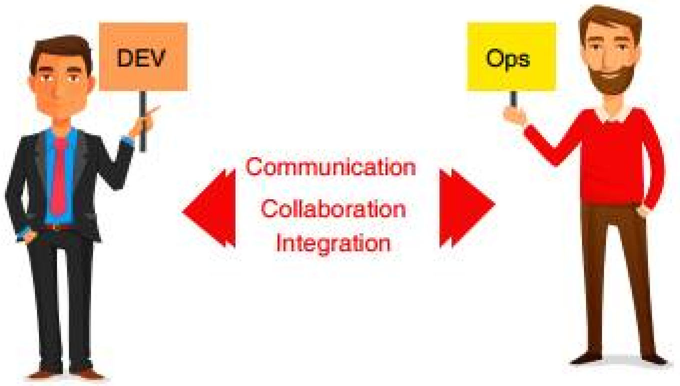

Fig.1.Understanding DevOps

A. *Devops Considerations:*

- DevOps is an approach based on agile and lean principles in which business owners, development, operations, and quality assurance team collaborate to deliver software in a nonstop stable manner.
- DevOps is an environment that endorses cross practicality, shared business jobs and trust
- DevOps is a movement that improves IT service delivery agility
- DevOps is a philosophy that promotes better working relationship within the organization

DOI: 10.15680/IJIRCCE.2016. 0406331

- DevOps is a set of practices that provides swift, trusty software delivery

II. Do we really need Devops?

Developers always want to deliver changes as soon as possible. Operations teams want reliability and stability. Lee Thomson describes this as a wall of confusion between development and operations. This wall of confusion not only survives amongst the mentalities of the two teams but also with the tools they practice. Development uses some tools and operation uses some other tools to perform the same work.

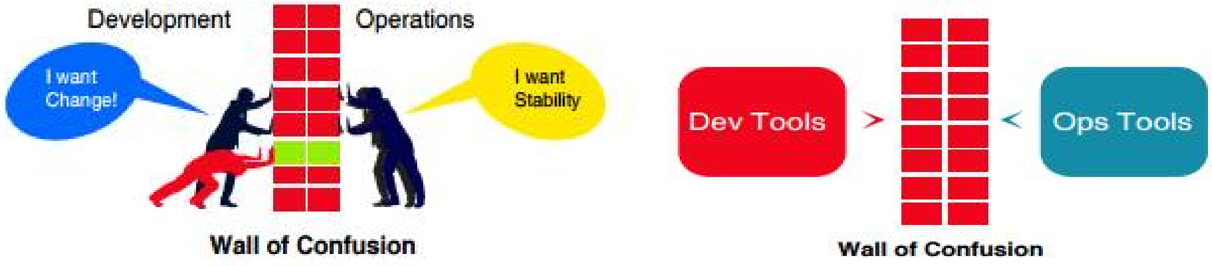

Fig.2.Wall of Confusion

DevOps break down the walls amongst development and operations team, merging development to operations for better, faster conclusions.

III. What drives the need of Devops?

A. *Considerations:*

- The need for greater collaboration between development and operations terms
- A greater need for synchronized deployment across dissimilar platforms
- Pressures from the business to release applications more hastily to meet customer demands or enter new markets
- Need to improve the end customer capability
- The increasing use of mobile devices (smartphones/tablets)
- The increasing necessity to develop or deploy cloud based applications
- An increasingly complex IT infrastructure that is part physical, part virtualized and part cloud
- Need to reduce IT costs

IV. Recognizing the business value of Devops

DevOps applies agile and lean principles across the entire software supply chain. It enables a business to maximize the speed of its delivery of a product or service, from initial idea to production release to customer feedback to enhancements based on that feedback.

DevOps improves the way that a business delivers value to its customers, suppliers, and partners, it’s an essential business process, not just an IT capability.

DevOps provides significant return on investment in three areas:

- Enhanced customer experience
- Increased capacity to innovate
- Faster time to value

A. *Enhanced Customer Experience:*

Delivering an enhanced (that is, distinguished and winning) customer experience builds customer loyalty and increases market share. To deliver this experience, a business must continuously obtain and respond to customer feedback, which requires mechanisms to get fast feedback from all the stakeholders in the software application that's being delivered: customers, lines of business, users, suppliers, partners, and so on

B. *Increased capacity to Innovate:*

Modern organizations use lean thinking approaches to increase their capacity to innovate. Their goals are to reduce waste and rework and to shift resources to higher-value activities.An example of a common practice in lean thinking is A-B testing, in which organizations ask a small group of users to test and rate two or more sets of software that have different capabilities. Then the better-capability set is rolled out to all users, and the unsuccessful version is rolled back. Such A-B testing is realistic only with efficient and automated mechanisms such as those that DevOps facilitates

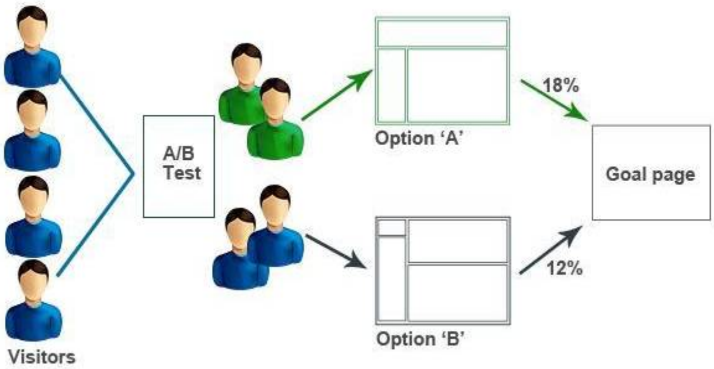

Fig.3. An example of A/B Testing

A-B testing (sometimes called split testing) is comparing two varieties of a web page to see which one performs better. We compare two web pages by showing the two variants (let's call them A and B) to similar visitors at the same time. The one that gives a better conversion rate, wins!

C. *Faster time to value:*

Speeding time to value involves developing a culture, practices, and automation that allow for fast, efficient, and reliable software delivery through to production. DevOps, when adopted as a business capability, provides the tools and culture required to facilitate efficient release planning, predictability, and success. The definition of value varies from organization to organization and even from project to project, but the goal of DevOps is to deliver this value faster and more efficiently

V. The Devops Architecture

The Devops architecture provides a pattern of a proven solution by using a set of preferred methods and capabilities. This help practitioners access and use the guidelines and directives they need to architect or design a DevOps platform that accommodates people, processes, and technology. The capabilities that make up DevOps are a broad set that spans the software delivery life cycle. Where an organization starts with DevOps depends on its business objectives and goals — what challenges it's trying to address and what gaps in its software delivery capabilities need to be filled. It mainly targets the delivery pipeline structured as part of the software delivery life cycle.

The Devops architecture composes the following four sets of adoption paths:

- Steer
- Develop/Test
- Deploy

- Operate

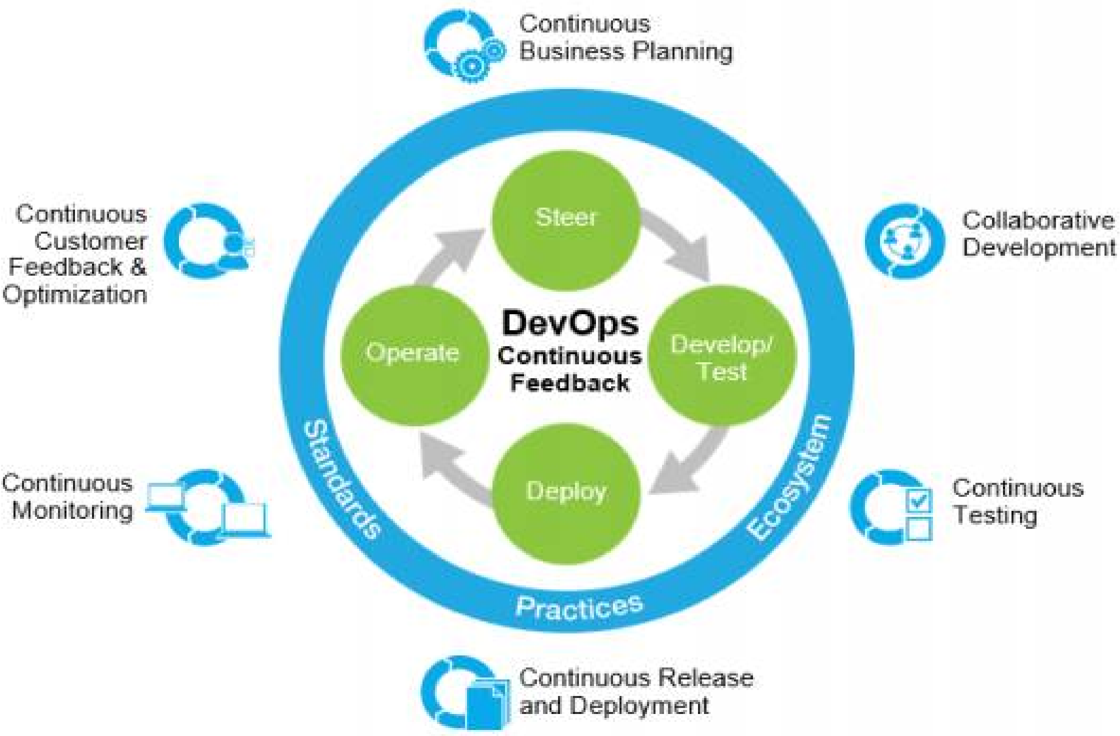
Continuous Business Planning
Continuous Customer Feedback & Optimization
Collaborative Development
Steer
Operate
DevOps
Continuous Feedback
Develop/ Test
Deploy
Continuous Monitoring
Continuous Testing
Standards
Ecosystem
Practices
Continuous Release and Deployment

Fig.4. DevOps Referential Architecture

A. *Steer:*

This adoption path consists of one practice that focuses on establishing business goals and adjusting them based on customer feedback i.e., continuous business planning

B. *Develop/Test:*

- This adoption path involves two practices: collaborative development and continuous testing. This forms the core of development and quality assurance (QA) capabilities
- **Collaborative Development**: This enables lines-of-business owners, business analysts, enterprise and software architects, developers, QA practitioners, operations personnel, security specialists, suppliers, and partners to work together by providing a common set of practices and a common platform they can use to create and deliver software. One core capability included within collaborative development is ***Continuous Integration***. It is a practice where software developers continuously or frequently integrate their work with that of other members of the development team

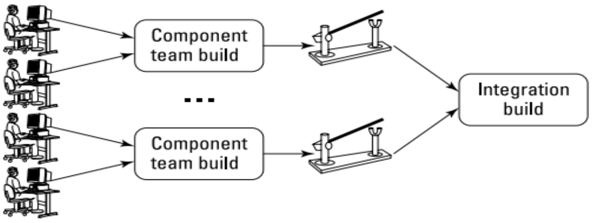

Fig.5. An example for Continuous Integration

DOI: 10.15680/IJIRCCE.2016. 0406331

- **Example for Continuous Integration:**
  - Consider a scenario – We are a developer working on an agile team. In fact, it is a large project, so there are multiple teams developing different components of the software application we are building.
  - We do our work, based on the user stories we and our team are working on. At the end of the day, we do a 'private build' of our work to verify it builds and 'deliver' it to a team build server. Other team members also deliver their work
  - We all 'integrate' our work in the common build area and do an 'Integration Build'
  - As we are working with other teams, we would then 'deliver' our teams work to a common cross team build server and do a system wide or application wide 'Integration Build'
  - Doing these integrations and builds to verify them on a regular, preferably daily basis, is what is known as **Continuous Integration**

- **Collaborative Testing:** Continuous/Collaborative testing means testing earlier and continuously across the life cycle, which results in reduced costs, shortened testing cycles, and achieved continuous feedback on quality. This process is also known as shift-left testing. Service virtualization is the new capability for simulation of production-like environments and makes continuous testing feasible

C. *Deploy:*

The Deploy adoption path is where most of the root capabilities of DevOps originated. Continuous Delivery is taking the concept of Continuous Integration to the next step. Continuous Delivery is otherwise called Delivery Automation. Deploying Software from one environment to another environment includes: Ex: a binary file deployment, configuration files movement, restart a app server, modify the database schema etc., All this needs to be done in a coordinated manner, steps happening at the right time (Orchestration). Once the application is built, at the end of every Continuous Integration build, deliver it to the next stages in the application delivery lifecycle. Deliver it to the QA team for testing and then to the operations team (the Ops in DevOps) for delivery to the production system. Continuous Delivery happens for applications as well as for environments. Automated deployment is the ability to get software deployed to any particular environment at any given time. Continuous delivery is the capability to deploy a software to any particular environment at a given time. This is called **Continuous Delivery**

D. *Operate:*

The Operate adoption path includes two practices that allow businesses to monitor how released applications are performing in production and to receive feedback from customers.

- **Continuous Monitoring -** provides data and metrics to operations, QA, development, lines-of-business personnel, and other stakeholders about applications at different stages of the delivery cycle
- **Continuous customer feedback & optimization -** The two most important types of information that a software delivery team can get are data about how customers use the application and feedback that those customers provide upon using the application. This feedback allows different stakeholders to take appropriate actions to improve the applications and enhance customer experience. This continuous feedback loop is an essential component of DevOps, allowing businesses to be more agile and responsive to customer needs.

VI. CONCLUSION

DevOps is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably.

REFERENCES

1. Carmine Giardino, Nicolò Paternoster, Michael Unterkalmsteiner, Tony Gorschek and PekkaAbrahamsson, "Sofware Development in Startup Companies: The Greenfield Startup Model", IEEE Transactions on Software Engineering, 2016
2. David P. Harvie, Arvin Agah, "Targeted Scrum: Appling mission command to Agile Software Development", IEEE Transactions on Software Engineering, 2016
3. Dan Hao, Lu Zhang, Lei Zang, Yanbo Wang, Xingxia Wu, Tao Xie, "To be optimal or not in Test-case prioritization", IEEE Transactions on Software Engineering, 2016
4. Earl T. Barr, Mark Harman, Phil McMinn, MuzammilShahbaz, Shin Yoo, "The Oracle Problem in Software Testing: A Survey", IEEE Transactions on Software Engineering, 2015
5. Lucy Ellen Lwakatare, PasiKuvaja and Markku Oivo, "Dimensions of Devops", Springer International Publishing Switzerland 2015
6. Manish Virmani, "Understanding DevOps & bridging the gap from continuous integration to continuous delivery", Innovative Computing Technology (INTECH), Fifth International IEEE Conference, 2015
7. S. R. Chidamber ; C. F. Kemerer, "A metric suite for Object Oriented Design", IEEE Transactions on Software Engineering, 2015
8. Maximilien de Bayser, Leonardo G. Azevedo, Renato Cerqueira, "The case for Devops in scientific applications", IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015
9. Michael Huttermann, "DevOps for Developers", Springer International Publishing Switzerland, vol. 1, 2012
10. Sungjoo Kang, Okjoo Choi and JongmoonBaik, "Model-based Dynamic Cost Estimation and Tracking Method for Agile Software Development, IEEE/ACIS Transactions on Software Engineering, 2010
11. M. WasifNisar, Yong-Ji WANG, ManzoorElahi, "Software Development Effort Estimation Using Fuzzy Login - A Survey", IEEE Transactions on Software Engineering, 2008
12. L. Dobrica, E. Niemela, "A survey on software architecture analysis methods", IEEE Transactions on Software Engineering, 2002

BIOGRAPHY

A.Remi Jullianwas born in Karaikal, Pondicherry state, India. He received MCA degree from SASTRA, Thanjavur affiliated to Bharathidasan University, TamilNadu in 2002 and MTech in Information Technology from Sathyabama University, Chennai in 2011.

He joined the Department of Computer Science, EGSPillay College affiliated toBharathidasan University as a Lecturer in 2002. He pursued Diploma in Software Testing from AmitySoft Technologies Pvt Ltd, Chennai and moved into IT services in 2004. He was a Software Tester cum corporate trainer expertise in handling Manual testing, automation tool trainings in functional, non-functional, test management & security test categories both in-house as well as giant corporates like Accenture, Wipro, IBM, HCL, SAP Labs, Ford Infotech, Capgemini etc., across India.

He has completed massive International testing certifications which includes ISTQB, CSTE, ICST, HP Unified Functional Tester, HP ALM, HP LoadRunner, IBM RFT, IBM RPT, IBM RQM, Tricentis Tosca, Tricentis Web Services Test Automation, Tricentis Cross Browser Test Automation etc.,

Heis a Project Program and Service Management (PPSM) Team Lead working for Accenture Services Private Limited, Shriram IT Gateway, Chennai, India from 2010. He is working for Learning & Knowledge Management (LKM) business function and leading a 10 member team.His current research interests includes Software Engineering (Software Testing), Agile Methodologies and Devops practices. He has received lots of awards, monetary & non-monetary recognitions for his good work.

M.Sangeethawas born in Tiruvannamalai, TamilNadu. She received her M.Sc degree in Information Technology from P.R College, Bharathidasan University, Trichy, India in 2002 and M.Tech degree in Computer Science and Engineering from M.G.R Engineering College, Dr.M.G.R. University, Chennai, India in 2007. Currently she is working as Assistant Professor in Panimalar Engineering College, Chennai, India. She has 10 years of teaching experience. Her interested area is in the field of Computer networks, Software engineering, Data structures and cloud computing to the distributed parallel architecture based Systems. She has received lots of recognitions for her good work.

DOI: 10.15680/IJIRCCE.2016. 0406331