Case Study Report
on

"Discreet Analysis of DevOps Implementation in SlideShare"

Submitted By:
Vishakh Lakshmikanth

DEPAUL UNIVERSITY
COLLEGE OF COMPUTING AND DIGITAL MEDIA

Under the Guidance of:
Prof., Christopher Jones

Abstract:

According to 2017 State of DevOps report “DevOps is an understood practices and cultural values that has been proven to help organizations of all sizes improve their software release cycle, software quality and ability to get rapid feedback on product development.” We have witnessed a drastic technology led transformation over the last few years. Companies have started leveraging DevOps methodologies to bridge the siloed structure thus improving the overall productivity with reduced product release cycles. One of the main goals of DevOps is to improve the overall workflow in the software development life cycle and create a more cohesive team and achieve maximum efficiency. DevOps mainly comprises of five activities including: Cloud Infrastructure, Continuous Integration, Continuous Delivery, Configuration Management and Test Automation. Continuous Delivery involves continuously creating releasable artifacts. Continuous Integration is a process in which developers and testers work collaboratively to validate new code. This Continuous Delivery and Continuous Integration (CICD) is a prominent process in some companies such as Facebook and Netflix to complete 10 or more releases per week.

History:

SlideShare is a non-profit organization being one of the renowned products of Microsoft Corporation. It entirely acts within the norms of Internet industry standards. This respective company officially started with 20+ employees in 2006 with Rashmi Sinha -being the CEO & Co-Founder of SlideShare. Now, it comprises of 70 million users with the highest of 150-Alexa page rank (as per Nov.2017). This website was originally meant to be used for business to share slides among employees more easily. In 2012, SlideShare was been acquired by LinkedIn and opened all its functionalities for the public internet users.

Implementation:

SlideShare tried to implement a DevOps model to speed up processes and stay ahead of the competition. The main goal of SlideShare in adopting DevOps practice was to create more interconnected team by overcoming geographic barriers for maximum efficiency. Organization started to implement the DevOps practices in significant phases. The implementation of DevOps practices started with a highly technical skilled team thus further, the team was spreading the technical knowledge across the other teams for limited impact in case of unavailability of any of the resource. The development team was divided and distributed geographically with a complex DevOps Cloud Infrastructure. As an idea of promoting a greater sense of ownership and leadership, developers were provisioned with the access to the production environment. DevOps became one of the keys to the organization's success in quickly reaching 29 million unique visitors per month thus raised the product shares & profit in the market.

Incident:

At SlideShare, engineers had the provisioned access to the production servers and production database - basically production environment. A software engineer was working on the database-related project and was trying out a tool that offered the ability to explore a MySQL database graphically. He decided to reorganize the order of the database columns in that tool and performed the database commit action so that the data would make more sense to him- basically to understand the data flow in the database, the engineer restructured the database schema. But the unfortunate fact is that the database schema in the production environment was changed and committed by the engineer's actions because engineer was unaware that he was actually performing those schema alterations on the production database.

After-Effects:

Due to this incident, SlideShare.net was shut-down for more than 60k live users trying to access the server. After the incident, the engineer responsible did not realize that the tool was actually performing adverse actions on the production database and it significantly took 15+ minutes of collective effort to find out the source of the problem.

Lessons Learnt:

1. While DevOps is enforcing everyone to have an impact in any step to the product cycle, it's a good practice more-over its safer to take a step back every-time when you give access to something and make sure it is actually valuable. In this specific situation of the database outage, we clearly interpret that giving access to production data/production environment was actually not wise and useful and turned out to be very dangerous and insecure. A possible solution is that the developer could have extracted the same exact values/visualizations using a staging database, but with a much more minor impact on the company.

2. It's better to educate developers and make them aware on the workings of the DevOps infrastructure. Many of them have never been exposed to the production environment. Hence, onboarding sessions are very useful and mandatory before deploying an employee/developer to work in DevOps operational environment.

Generic Reasons for DevOps Failure:

1. Lack of DevOps Department:

   Continuous Integration feeds continuous Delivery stage. Implementing a decent CI pipeline and a full CD system requires the collaboration of quality assurance, DevOps team, operational engineers and scrum master. DevOps implementation requires leadership and framework wherein the development and operation's teams work collaboratively.

2. Failing to properly consider suitable Human Resources:

   Main key point in this scenario is to Quantify the team's workload and monitor each individual's performance.

3. Setting unrealistic goals/deadlines:

   A fundamental reason that DevOps is trickier to introduce than other technologies is that it fuses cultural shifts with operations and development. The larger the enterprise, it takes longer for transformation. In this case, it is important to set and track multiple metrics and to mention these metrics must be specific-aligned with the goals of the business.

4. DevOps Transformation:

   To consolidate this point, implementing the DevOps principles and strategies in a new company is very easy but shifting an existing company from Agile principles to DevOps methodology is quite challenging which makes companies have a nightmare during this phase of transformation.

5. Insufficient/Lack of Knowledge:

   The above narrated incident is a clear reflection of lack/insufficient knowledge/awareness towards DevOps. In the initial phase of transformation, company must provide sufficient onboarding knowledge session so that the employees would having their awareness during the development.

Reasons that lead for DevOps Success:

1. Creation of standard infrastructure blueprints and implement continuous delivery to ensure all environments are identical.
2. Automate the build and deployment processes as much as possible and implement a test automation methodology similar to Test Driven Development (TDD)
3. Invest in training and hold blameless post mortems to continuously solicit feedback and enhancement.
4. Organizations with legacy/traditional processes need to look at how then can modernize/enhance the processes to be more agile instead of being the reason why their company can't move fast enough.
5. Assess your operational processes, tool and organization meanwhile modernize to increase agility and transparency.
6. Quality is everyone's responsibility and not just the quality assurance's team.
7. Automate processes after the bottlenecks are removed.
8. Appoint an in-charge and start building a plan for scaling DevOps across the organization.

Key factors for DevOps Transformation:

According to 2017 state of DevOps report there are few quite key findings that can bring lot of changes in the organizations with DevOps implementation.

1. Transformation must include vision, inspirational communication, intellectual simulation and leadership.
2. High performing teams continues to achieve both faster throughput and better stability.
3. Automation is a huge boon to any organization.
4. Loosely coupled architectures and teams are strongest predictor of continuous delivery.

Practices that make up continuous delivery includes deployment automation & automation testing, continuous integrations & trunk-based development and version control has major and significant predictive relationship to deployment pain, it's performance and change failure rate.

Conclusion:

The success or failure of DevOps automation doesn't depend on any single factor, but in-turn comprises the chain of actions. DevOps takes time and significant effort – requiring strong leadership and these must be measured in a way that's more compatible with the organization's goals. If the deployment process relies on humans, then there is a huge chance that the company would get exposed to risk. Hence, deployments must be automated – repeatable as free from potential human errors. According to the recent Gartner report, 70% of the companies in IT market are now focusing on DevOps methodologies. Hence, traditional enterprise companies are now embracing both Agile and DevOps techniques.

Bibliography:

1. “Continuous Delivery & DevOps” Reference Book by – Jez Humble, David Farley
2. https://www.ipexpoeurope.com/content/download/10069/143970/file/2017-state-of-devops-report.pdf
3. https://www.linux.com/news/how-devops-failed-60k-users
4. http://www.tothenew.com/blog/3-devops-misconceptions-failures-and-lessons/
5. https://www.infoworld.com/article/3113680/application-development/5-common-pitfalls-of-cicd-and-how-to-avoid-them.html
6. https://techbeacon.com/4-biggest-reasons-devops-failure
7. https://technologyconversations.com/2017/03/06/the-ten-commandments-of-continuous-delivery/
8. https://dzone.com/articles/agile-and-devops-are-failing-in-fortune-500-compan
9. http://www.tothenew.com/blog/3-devops-misconceptions-failures-and-lessons/
10. https://www.forbes.com/sites/mikekavis/2014/12/18/11-common-devops-bottlenecks/#438941727737
11. https://techbeacon.com/10-companies-killing-it-devops
12. https://dougseven.com/2014/04/17/knightmare-a-devops-cautionary-tale/