



DevOps: The New Frontier of Software & Project Development

WHAT IS DEVOPS?

In life, there are concepts, terms and tools that mean different things to different people. In some cases, “abstract” means something is vague; in others it refers to a summary of something. Likewise, in baseball, “ball” can refer to a pitch that is not a strike. Or, it could simply refer to the actual ball in play.

DevOps is similar. At its core, it's a way to merge Development and Operations. The benefits of a DevOps Model are increased agility, speed in development/delivery, enhanced reliability, greater scalability, better security and greater collaboration. However, DevOps doesn't have purely defined methodologies or practices to accomplish this; rather, it's a group of practices and principles.¹ It's based on changing behavior, not implementing a specific tool or product.

In most organizations, the typical set up of IT Departments used to be keeping different groups segregated from each other. An engineer focused on engineering solutions; a developer focused on developing the product; a tester focused on testing the product, and the departments spoke only as required. This often led to bottlenecks and delays in completing a project throughout the SDLC.

Testing is a great example of such a bottleneck. In a traditional SDLC, testing is a singular phase and often falls late in the project lifecycle. This approach makes things difficult because if flaws are uncovered, it's more challenging to figure out where



the issue arose and what the best solution would be. Since the testing phase often comes close to release deadlines, it's a nightmare waiting to happen.

In a DevOps model, this wouldn't be an issue. DevOps stresses communication and collaboration amongst departments so testing is going on throughout the SDLC, as built in activities within other SDLC phases.

“In the past you had discrete phases such as Requirements, Design, Build and Quality Assurance (QA),” Dennis McDonald, VP of Business Transformation at Paradigm Technology said. “So if something failed, you would have to go back to earlier phases of the cycle such as Requirements and basically walk through the process over again. In a DevOps model, you have the capability within later phases such as QA to make those corrections on the fly and retest it, because it is now part of the shared responsibilities of the different team members.”

¹ Forsgren, N.; Humble, J.; Kim, G.; Brown, A.; Kersten, N. (2017). 2017 State of DevOps Report.

Technology has made the world smaller and the need for rapid delivery of products and services has increased, putting a greater strain on IT departments, causing rapid burn out, more stress and a greater likelihood for human error. This is particularly true in the small-to-medium (SMB) business sector. These organizations have to do more with less in order to compete more effectively. The collaborative nature of DevOps alleviates this problem by enabling users/customers to receive new software or updates quicker, allowing organizations to adapt and roll out new services in less time.

Competing more effectively is a clear benefit for SMB's, but enterprise organizations often struggle with visibility issues. There aren't enough pairs of eyes on everything going on in every department, which sometimes causes a jumble and leads to the bottlenecks detailed above.

In a 2014 interview with Mike Baukes, he says he constantly ran into these issues with clients, particularly in the financial sector, and that it is difficult in larger organizations to see how everything operates together.²

“After hitting our heads against this problem so often, and trying to automate solutions to build these services, it became pretty evident that if we were to stop and look at the composition of these companies that the best way to get everyone engaged and understanding these services is to use an emerging philosophy... DevOps is really about getting cohesive collaboration inside an organization in order to deliver software for the benefit of the business.”

HOW TO EFFECTIVELY IMPLEMENT DEVOPS

Implementing a DevOps Methodology might seem like an obvious step for all organizations. The potential benefits are obvious but it's not quite that simple. It's important organizations approach this new method in a thorough, well thought out manner.

However, DevOps requires experimenting before getting it implemented successfully, and smaller organizations tend to be more flexible about this process. These organizations usually have fewer departments and team members that must be trained, so it's easier to switch course and adjust if something isn't working. For larger organizations, a more methodical approach is required. There are more people involved, more decision makers to get buy-in from, more processes in place and more employees that must be trained. There is also less room for error, especially for organizations that try and do too much, too soon. So starting small and scaling up is the best approach to take.

There are obviously differences in the nuances of effectively implementing DevOps specific to each company, but there are also plenty of guiding principles for companies of all sizes to adhere to. A starting point for any project is planning. Speak to different teams about their issues and departments; identify ways that DevOps could solve these issues and assess which issues make sense to address first.

EDUCATE/GET BUY IN

DevOps represents a drastic shift in philosophy and culture, so it's important to educate employees about the purpose behind the shift and how it would impact their daily responsibilities. Teach them to understand all stages of the project lifecycle and how to perform tasks that have typically fallen to other team members. Developers must know how to perform testing responsibilities, testers must know how to perform development tasks, and so on. Explain that sharing responsibilities will reduce development cycle times and potential issues will be caught and fixed earlier, freeing up time to work on new projects.

² Vanian, J. (2014, May 9). Implementing DevOps in Large, Complex Organizations: An Interview with Mike Baukes.

“DevOps changes the culture of how a company does business. Team members need to understand this and transform how they do things. It is not ‘big bang.’ Change Management is critical,” McDonald said.

BUILD THE RIGHT TEAM & GIVE THEM THE RESOURCES TO BE SUCCESSFUL

Whether it is building a sports team or a DevOps team, understanding how to build the right type of team is paramount to success. It’s not just talent that makes a team successful; it is cohesion, collaboration and trust.

“Upper Management needs to be able to trust that DevOps team members have the skills necessary to go outside the scope of their past responsibilities,” Kim Arbogast, Sr. Paradigm Project Management Consultant, said.

Dan Cornell says the most successful DevOps teams he’s seen have combined existing internal employees with an influx of new outside talent because it allows for familiarity with established internal decision-making processes and enough new thinking to spawn new ideas.³

“You’ve got to understand your culture, understand the broader integration elements of it, and make a decision on if it is the right thing to do for your company... there is no ‘one size fits all,’” McDonald said.

START SMALL

When a company tries doing too much too fast, they may be biting off more than they can chew. They risk longer project delays and unnecessary backlash from customers and senior management. For those championing a switch to DevOps, the consequences can be severe and result in the end of the shift altogether.

“Having buy-in and trust from senior leadership is essential to making a DevOps shift work,” Arbogast said. “It can’t be done overnight, and it must be well thought out before starting such a transition.”

DevOps is ongoing and iterative. A strong governing model must be in place; starting with a Proof of Concept (POC) in order to demonstrate the value and cultural shift required to scale DevOps within the organization is usually the best way to accomplish this. So, a sensible plan would be taking a singular issue that needs to be solved and use a DevOps approach to try and solve it.

It is also important to identify the right people to begin leading the shift. Gene Kim says that while any team member can lead, the most common people are the Director of Operations, Chief Architect or Director of Development.⁴

“What these leaders do well is scope large enough so they can solve something of genuine significance and everyone appreciates they created a material improvement for the company, but small enough where they can also manage the risk.”

CAN DEVOPS WORK IN A DISTRIBUTED LABOR MODEL?

One hotly debated question regarding DevOps is whether it can function in a Distributed Labor Model, with part of the team located domestically and another part located abroad. There are many factors to consider. Many companies have invested heavily in a Distributed Model, so pulling the plug on such a model is not possible due to costs, potential impact to other lines of business and preexisting contracts. In addition, DevOps is built around continuous communication, continuous implementation and continuous integration - all of which are significantly hindered by distributed labor.

There are multiple schools of thought when it comes to DevOps functioning at a high level in a distributed labor model. Peter Bendor-Samuel believe DevOps teams get optimal output if the team size is between 8-16 people and can be fed by two pizzas. An ideal DevOps team should be close enough to each other that they can throw a nerf football back and forth.⁵

Another factor to consider is that a Distributed Labor Model is usually built for specific projects, whereas a DevOps team should be project agnostic to reduce turnover and maximize the cultural consistency it’s intended for. However, others in the technology industry believe DevOps can function effectively in a Distributed Model if the right parameters are in place.

³ Chickowski, E. 10 ways to build highly effective DevOps Teams.

⁴ Denisco Rayome, A. (2017, April 3). How to implement DevOps: 5 tips for doing it right.

⁵ Bendor-Samuel, P. (2017, October 19). Creating IT Value: Can DevOps be delivered in a distributed labor model?

Others say that certain aspects of DevOps are not well suited for Distributed Labor, but there are also aspects that can work. For organizations that require support 24 hours a day, a strong onshore leader with a strong technical background and experience with lean management must have constant interaction with the offshore team. Tasks such as automation, engineering, deployments and non-urgent alerts are best suited for offshore work in a DevOps model.⁶

Other ways of increasing effectiveness of a Distributed Labor DevOps Model include shift overlap, where there are a few hours a day where both teams are able to communicate any issues or challenges in real time. In addition, having a mature and experienced team makes things smoother and fosters an environment of trust where team members have the power to make decisions without scheduling meetings all the time.

FINAL THOUGHTS

DevOps can have a profound impact on an organization, increasing agility and efficiency, eliminating unnecessary bottlenecks, fostering a culture of collaboration and helping organizations compete more effectively in the marketplace. However, there is no “one size fits all;” each organization must approach the shift by first understanding their own environment and culture, then establishing a plan of action that is able to be modified along the way.

DevOps is more about people and culture than methodology, so identifying the right team and leadership is essential. Strategically it typically makes the most sense to start small and identify a DevOps team that is project agnostic but, again, each company must decide what makes the most sense.

The bottom line is that each organization must ask itself if moving to a DevOps model makes sense. If the answer is yes, then it must ask and answer one more question: What does DevOps mean to them?

ABOUT THE AUTHOR

Jordan Jurkowitz, Senior Talent Acquisition Manager at Paradigm Technology: Jordan has been with the organization since 2014, focusing on projects across all lines of business. An avid writer, he joined Paradigm with a degree in Journalism/ Mass Communication from the Walter Cronkite School of Journalism and numerous publications in a local newspaper.

An award-winning end-to-end strategic solutions provider, Paradigm Technology is a leader in Digital Transformation, working for 25 years with the Fortune 500. We enable our clients in: Customer & Vendor Centricity, Data Governance, Real-time Analytics, and Cloud Adoption. Our Data Scientists help make sense of IoT and Big Data to predict and answer the hard questions to grow revenues, reduce costs and avoid risk. Paradigm's Business Transformation practice supports this through IT simplification, change management, and robotic automation.



www.pt-corp.com | 480-473-7111 | info@pt-corp.com