THE ROLE OF AGILE IN DEVOPS

As technology continues its blistering pace of advancement towards the next big thing, companies scramble to keep up with the demand of modern society. The arms race is never ending in the pursuit of providing customers with the services and products on which they are willing to spend their hard-earned money. DevOps is one of the new advancements made in the IT sector of business operations that has everyone abuzz.

While it may seem like a lot of hype, DevOps has proven itself to be an invaluable tool in maximizing the potential of software enterprises to build, test, and deploy their products at blazing speeds and with more success than previously found using traditional methods. DevOps has earned its place among the greats as a system for empowering software teams to reach new heights, but what exactly is DevOps?

(This article is part of our DevOps Guide. Use the right-hand menu to navigate.)

The Basics of DevOps

DevOps is more of a cultural shift in how software development is approached than it is any singular system or collection of tools. Having said that, the core pursuit of DevOps systems is really just one thing: collaboration. Collaboration is at the core of everything DevOps wants to achieve. Even the name itself is just a collaboration of the words development and operations. As such, DevOps is a mentality of working together and finding new ways to collaborate better than before.

In pursuit of this, DevOps emphasizes the need for effective communication and extols the merits of transparency. Best practices, tools, and technology that enhance communication are cornerstones of an effective DevOps system. Things such as cloud computing, version control, and automation

play a large role in the success of DevOps, but these tools are only a means to an end. The true secret sauce is the culture of collaboration and the leveraging of goal-oriented and customer-centric ideologies like Agile.

Agile's Impact on Software Development

DevOps owes much of its history and success directly to Agile. In the early days of software development, development lifecycles were orders of magnitude longer than they are today. The entire process took place in siloes cut off from the users and customer-facing teams. The method that was first adopted for software development was called the "waterfall" approach, which held the idea that developers would define a customer need and then develop a single product that met that need successfully before releasing it to the public. This process caused long development cycles in which the developers had no interaction with customers, resulting in the complete lack of feedback and the creation of software that often fell short of user expectations.

The developers would work in their own silo while customer service and operations worked in their respective regions and information passed seldomly and inaccurately between these disparate teams. The result was expensive projects that would often fail to adequately service the needs of the customers. Without feedback and communication, products struggle to improve.

During the earlier days of development when waterfall methods were commonplace, there was less competition due to the various barriers of entry for the software development sector. Those barriers began to rapidly fall to the wayside as technology leaped and bounded past even the most optimistic person's expectations and competition began to sprout from every corner of the world. The empire that is known as Google today is famously known for having begun its life in a rented garage. As more competition appeared on the scene, practices for developing better software arose and Agile proved itself as an invaluable development system.

With Agile, the development lifecycle was dramatically sped up and the process involved customer interactions throughout. The focus shifted from comprehensive documentation to creating operational software as quickly as possible so it could be tested and iterated upon rapidly. Agile also brought about the idea of software as a service (SaaS), where there is no truly "finished product" as the software is constantly being improved upon. Agile turned enormous projects into smaller, bite-sized deliverables that would be churned out on a regular basis with constant feedback.

And Then Came DevOps

As Agile is an approach to development that focused on interaction, so too is DevOps. With DevOps, collaboration with teammates, customers, and executives should become second nature. The main application of Agile methodologies is the utilization of automation for testing and building to embrace the ideals of continuous integration and continuous delivery. DevOps also embraces the power of automation but takes it a step further by emphasizing the importance of creating cross-discipline teams and using automation to empower collaboration between team members and other teams.

DevOps can be seen as an extension of Agile where communication is used as the tool to pull everything together into a fully functioning development unit. DevOps looks to fill the gaps within enterprises by promoting transparency and understanding for everyone - not just developers. Having a deeper understanding of each stage of the process empowers people with the knowledge

of how to best perform their own role as well as how to aid others in performing theirs.

While Agile focuses on creating and testing new software, DevOps takes it a step further and looks to successfully create, test, and deploy the software. This distinction is important because it's the role operations plays in the whole scheme. Agile looks to put developers into teams to speed up development while DevOps creates teams of cross-discipline members where the whole is greater than the sum and everyone can approach the project with their unique perspective.

Agile's main pursuit is speed while DevOps is accuracy. Combining both successfully results in the best of both worlds where teams rapidly build, test, and deploy updates that are stable and marked improvements on past builds. DevOps achieves this through the combination of cultural shifts in mentality and tools that empower the new ideology. Creating a DevOps team isn't an overnight process, but the potential rewards are more than worth the effort.

DevOps: Solutions for You

If DevOps sounds like a good fit for your organization's needs but you want to make sure you get it right the first time, BMC is the IT solution partner you need. Read more about how automation and DevOps systems can help increase the rate at which you deploy products with BMC's free eBook: *Automate Cloud and DevOps Initiatives*.

BMC offers IT Cost Management solutions that help you optimize your spend on IT resources such as cloud services. Check out BMC's free eBook on determining the true cost of clouds to get a better idea for how much implementation will cost your enterprise. DevOps leverages the power of various tools to create the best environment for teamwork and collaboration.

In addition to IT management resources, BMC offers consulting and deployment services. BMC expert consultants are available to work with you to bring their knowledge and expertise to your organization. BMC also provides custom-tailored Deployment Services for your organization to tackle the unique challenges you face. When partnering with BMC, you get:

- **Faster service delivery:** Agile releases that keep up with rapid demand
- **Visibility across data:** Ensure compliance and data accuracy
- **Cost-effective service:** Increased productivity and performance
- **Experienced DevOps professionals:** Equip you with the tools you need for success
- **Conversion or upgrade:** Seamless modernization or total replacement
- ***All tailored for the specific needs of your organization.***

Download or view the Solution Implementation Overview online to learn more about how BMC Consulting Services can help you. Learn more about how DevOps and Application Deployment best practices can enable your teams to create better software faster than ever before. Then contact the experts at BMC to learn more about how Agile and DevOps practices work together for enhanced building, testing, and deployment success.