Global Knowledge®

Expert Reference Series of White Papers

DevOps is the Missing Piece for Businesses that Demand Change

1-800-COURSES

www.globalknowledge.com

DevOps is the Missing Piece for Businesses that Demand Change

Abbey Wiltse and Nasser El-Batal

What is DevOps?

If you are involved in development projects, managing releases or working to keep operations running, you are well placed to leverage the knowledge of DevOps. Are you a developer trying to keep pace with business demand for innovative solutions? Or are you in IT operations—dealing as best you can with the technical backlog of "fixes," but unable to get out of firefighting mode? DevOps works to address these real-life challenges. It's also an area where CIOs recognize there's a large skill gap for DevOps professionals. This white paper will bring you up to speed on what DevOps is and how it can help IT deliver value to the business and keep up with business demand for change.

Like-minded individuals started DevOps back in 2009, with the first DevOps Days conference held in Ghent, Belgium. A groundswell movement then started, and many "best-of-breed" vendors came to market with DevOps enabling technology, such as Puppet, Jenkins, and Chef. With Gartner's recognition of DevOps in 2011, many in IT started to take notice of the early DevOps adopters, such as Netflix and Etsy, and the large software vendors (IBM, HP, etc.) also entered their products into the DevOps market space. DevOps Days are now held globally and formal certification programs are in place to support the new skillsets required to support an organization through a DevOps adoption initiative.

Based on the nature of DevOps as a grassroots movement, there are currently many different definitions for DevOps. There is not standard best practice or body of knowledge in place today for DevOps, though there are plenty of good practice examples of the DevOps principles working to increase IT workflow and to quickly deploy well-tested features into the production environment.

DevOps Growth from Lean, Agile Scrum and IT Service Management

Building upon the IT best practices of Lean, Agile Scrum and IT Service Management, DevOps adds that "missing" layer to tie together the service lifecycle workflow across Development and IT Operations, while leveraging the latest in automation technology.

Just as there is no formal body of knowledge for DevOps, there is no one definition for this term. DevOps be described as a philosophy that breaks down barriers between IT Operations and Development through collaboration and communication.

DevOps utilizes automation where possible to ensure rapid and frequent releases of quality software to meet business demand.

One big cultural change that underpins the DevOps approach is based around a “fail-fast / fail-often and learn” attitude. This involves creating a high-trust, blameless working environment within IT, with a focus on continual learning and proactive management to identify and eliminate delivery issues before they occur.

Why DevOps Now?

DevOps has gained momentum in addressing the challenges of many stakeholders: the business, Development and IT Operations.

Most all business processes today are enabled by IT services, and indeed IT can underpin core disruptive business strategies (think Uber or Airbnb). The business demands an IT division that is responsive to its strategic timelines, and IT organizations everywhere are trying their best to keep up—more often than not at the cost of a healthy work life balance for employees in IT (more on this topic later).

Developers are looking to better leverage their investment in agile skills and supporting technology in response to business demand.

IT operations needs to ensure that maintainability, reliability, security and service level requirements are included in the application design phases, and to bring to the table their skills to leverage automation throughout the deployment pipeline.

What DevOps Is Not

Sometimes it’s better to define DevOps by “what it isn’t,” for example:

- It’s incorrect to think that DevOps is *only* about cultural change or automation—you need both to achieve the benefits of DevOps, but each on their own does not make a DevOps environment.
- Many organizations approach DevOps as a one-time “project” or put together a single team to “do DevOps.” This approach may get initial traction, but the principles of DevOps need to be adopted across the complete organization to be fully effective.
- So many companies say “I have ‘xxxx’ tool, so I’m doing DevOps!” Without changing your culture, adopting new best practices, and formalizing the deployment pipeline, you really haven’t changed your working practices to exploit the full value of DevOps.
- With DevOps, we no longer need IT Operations—the role of IT Operations does need to change. The focus now shifts to provide self-service environment management capabilities for Development, to proactively support continuous monitoring within the production environment and to greater involvement in the requirements gathering and design stages of the development lifecycle.

- DevOps means the end of IT Service Management—our best practices for managing the IT service lifecycle are a key element of DevOps adoption success. IT services still need to be designed, managed, and improved throughout their lifecycle.
- DevOps means we don't need documentation or processes. Many people swing the pendulum towards anarchy as opposed to the blend of flexibility and control elements needed to respond to business demand and still deliver quality code into a stable, service level target managed environment.
- Another challenge stated by organizations is that DevOps is only for "Unicorns"—those small start-up organizations that have set the good practices examples in DevOps adoption, such as Netflix or Etsy. Large organizational "horses," such as IBM, Amex, Target, GM, and the U.S. Government also have experienced good success with their DevOps adoptions.

DevOps and Technical Debt

One of the key challenges working in IT is the constant accrual of what's called technical debt. In most IT organizations, Development and IT Operations are different functional organizations, with their own success measurements and skill sets. These different perspectives and measurements result in conflicting goals, reduced quality, less agility to respond to business demand, higher impacting outages, and the accumulation of technical debt.

Development shops are focused and measured on delivering application functionality and usability into production in quick response to the business demand. Their work is usually managed and funded as capital-expensed projects, and once their code is deployed into production, they're off to their next project.

Quite often, the "non-functional requirements," or NFRs, relating to the reliability, maintainability, and operability (the "-ilities") ease of the application once deployed into production are overlooked or left out in order to meet the time to market demand of the business.

On the other hand, IT Operations is focused on providing a stable production environment that leverages automation to proactively detect and repair outages, relies on a resilient designed architecture and manages each environment (Dev, Test, Pre-prod / Staging and Production) to standard footprints for OS, middleware, network and security. They leverage IT Service Management best practices to ensure the stable running of their infrastructure environments to deliver agreed-upon service level targets for service availability.

Because of the different goals of Development and IT Operations shops, this inevitable descent into technical debt starts slowly and builds over time. Because of compressed timeframes in Development to meet business demand, code gets deployed into production with errors/issues, and IT Operations spends a lot of time firefighting or fiddling with workarounds to keep things running smoothly. Just like financial debt, technical debt builds over time, just as making the minimum payment on your debt payments puts you into a downward financial spiral that you cannot escape from.

This is the infamous “throw it over the wall to Operations” syndrome, as Developers move on to their next project. IT Operations knows where the issues are, either in the code or supporting infrastructure, but they have neither the time nor budget to fix these issues as they are busy just trying to keep the lights on. In the meantime, Development is trying to process and release more and more code to meet the business demand, often releasing into unstable environments which increases the demand on IT Operations to continue firefighting.

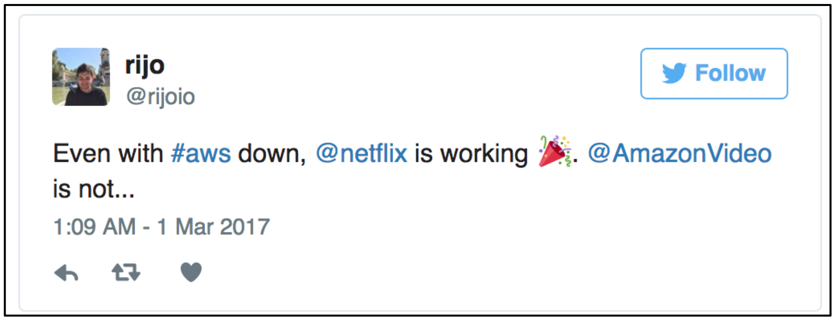
Eventually, everything slows down—Development’s ability to release more code and IT Operations staff bearing the brunt of the human toll, ultimately feeling ineffective and burned out by this vicious cycle. This is what we call technical debt. A key concern of DevOps is how to manage and ultimately eliminate this technical debt.

Does DevOps Really Work?

Does DevOps really deliver on its value proposition? The answer is yes! Let’s take a look at Netflix.

Netflix is hosted in the Amazon Web Services (AWS) cloud environment. They have adopted the “fail fast, fail often and learn” principle of DevOps. They created open source code they call their “Simian Army,” with functionality such as Chaos Monkey; code that routinely introduces failure into their production environment during regular business hours to test their resiliency and improve where there are design failures.

In early 2017, AWS cloud services went down, with Amazon’s own streaming service unavailable to clients. But the Netflix service kept running throughout this failure as a result of their approach to continually improve.



rijo
@rijoio

Even with #aws down, @netflix is working 🎉. @AmazonVideo is not...

1:09 AM - 1 Mar 2017

When DevOps is adopted to its fullest capabilities, a day-in-the-life looks a bit like this:

- Developers design functionality and usability requirements to meet business needs, along with the maintainability and operational support requirements—the “Non-Functional Requirements.”
- They create their code and automate their tests in environments they have the ability to quickly establish (i.e. spin up VMs representing the live environment) while leveraging standardized technology stacks that are maintained on their behalf by IT Operations.
- Developers leverage Agile and DevOps continual integration / continual delivery techniques to release well-tested code frequently in small batches to the deployment pipeline.
- Feature deployment to production is automated where it makes business sense, and can happen during the business day, improving work-life balance for IT Operations.
- Technical debt is reduced or eliminated all together, increasing morale.
- Postmortems are blameless learning opportunities and IT becomes a more humane place to work.

A Few DevOps Definitions

In order to discuss DevOps from different perspectives, it helps to introduce a few key DevOps principles and their definitions. It’s interesting to note that the origins of these practices can be found in many different disciplines, like Agile, Lean and IT Service Management.

Deployment Pipeline

The deployment pipeline defines the build, test and production environments where software is deployed. By documenting this pipeline, using techniques such as value-stream mapping, organizations can visualize this workflow and identify existing technology enablers as well as any opportunities to apply automation to gain efficiencies.

Continuous Integration

Developers “commit” small changesets of code into the mainline or master branch on a regular basis (at least daily) so that any errors are quickly identified for corrective action, resulting in the ability to release more frequently at a higher quality. Automated continuous integration tools can be used to achieve this goal.

Continuous Testing

Automated tests are built and executed to deliver immediate feedback throughout the deployment pipeline and the complete development to ensure quality software deployments that deliver to the business requirements. The technique of Test Driven Development (TDD), where developers write the tests that their code must pass before coding begins, can be leveraged to derive many of these automated test models.

Continuous Delivery

Software can be built, tested and reliably released at any time. This is achieved through leveraging the continuous integration and continuous testing techniques and requires a clear definition of the deployment pipeline.

Continuous Deployment

Continuous deployment implies that as soon as the software development and the required testing is complete, the change is automatically deployed to the production environment. Even though an organization may practice continuous delivery techniques, they may be constrained by business reasons (policy, risk profiles, change procedures, etc.) in their timing to push the changes into production.

Continuous Monitoring

Continuous monitoring is the automation of monitoring the production environment to ensure early warning feedback of performance or availability issues by IT Operations in order to prevent or minimize service level target breaches and to collect information for root cause analysis of underlying problems for future service improvement opportunities. It is critical that these automated monitoring requirements are identified with IT Operations as part of the service design, build and testing stages of the development lifecycle.

Benefits for the Business

Businesses today are dealing with multiple challenges, including:

- Speed to market demand and strategic differentiation.
- Getting the value from Agile software development techniques.
- The need for an increased frequency of releases.
- Responsiveness from the IT organization.
- Quality and assurance of IT service levels.
- Visibility and auditabiity of business processes and governance requirements.
- Ongoing business value and relevance in today’s fast-paced changing world of disrupters and start-ups.

If we look at how digital technology is transforming our business world, we can see how Uber has transformed the taxi industry, how Airbnb has transformed the hotel industry, and how Amazon has changed our retail experiences. It would be remiss of any business to ignore the strategic advantage to be gained from today’s underpinning digital technology. The reliance on IT as a competitive enabler is higher today than at any other time and business process change almost always includes an IT change for shared success.

Organizations that have adopted DevOps principles are more able to:

- Deliver business functionality efficiently and effectively with service level assurances.
- Compete with smaller, leaner businesses.
- Minimize traditional operating costs, specifically in areas of rework and duplication.
- Create an empowered work environment for their staff based on a culture of continual learning and high trust.

Benefits for Developers

As DevOps is based on the foundation of Agile, it is a somewhat easy shift to adopt some of the key DevOps principles for developers. Projects, where appropriate, should be managed leveraging Agile Scrum techniques to design and code functional changes in smaller change sets as part of each Scrum iteration.

Adoption of the DevOps technique of Continuous Integration is essential to speed up the delivery of functionality changes to meet business demand, by developing small change sets and merging code into the branch at least daily. Also, the technique of Continuous Testing, supported by a Test Driven Development practice, ensures that code is tested at all phases across the deployment pipeline and automated wherever feasible.

These combined practices help to achieve the goals of Continuous Delivery, where the code is tested and ready for release at any time; either immediately (Continuous Deployment), daily, or at a later date and time as per corporate release policies. These practices help to create a positive downstream effect of allowing multiple deploys during regular working hours to minimize or even eliminate the requirement for the mega-effort weekend deployment marathons for IT Operations and project teams alike!

There are a few cultural changes required for the Development teams to effectively achieve the full benefits of DevOps. Some of these changes are:

1. Development teams and Scrum Masters need to include IT Operations in the design and business requirements gathering activities of the software development lifecycle.
   - This is to ensure the "-ilities" of the non-functional requirements are designed and instrumented early in the lifecycle, for example: availability, reliability, security, capacity (performance), maintainability (automated event monitoring / self-healing code), service level requirements, etc.
2. Developers need to participate in improvement cycles with IT Operations.
   - IT Operations can work with Development teams to create as much self-service capability as possible to enable them to spin up environments on-demand that accurately represent the live environment.
   - This collaborative approach to leverage automation and "infrastructure as code" enables effective testing environments for developers, so all code is deployment-ready in the Deployment Pipeline as per the organization's release policies.
3. Development's measurements need to be changed.

- To reflect not only the feature time to market and project-based success measurements, but also measurements of application maintainability and supportability effectiveness and efficiency, which have traditionally been operational key performance indicators.

Benefits for IT Operations

DevOps is especially beneficial to IT Operations. The fallout from upstream inefficiencies and accumulation of technical debt is most strongly felt in IT Operations, resulting in work dissatisfaction and burnout. The IT Operations staff knows what issues need to be fixed, but have neither the time nor the budget to fix the underlying problems. They are not only caught up in the day to day firefighting to keep the IT services running, but they are then called upon to work long weekends to support large deployment releases.

So how does DevOps help improve this situation for IT Operations? The main change is having the recognition of technical debt across Development and IT Operations, and setting aside time to focus on elimination of this debt. This starts with consciously not introducing more downstream debt, while dedicating a specific amount of time to work through the "debt backlog," ensuring both IT Operations and Development are participating in this Agile improvement effort.

As noted previously, developers cannot achieve the benefits of DevOps alone. They need the full participation of IT Operations:

- As part of Scrum development teams to participate in requirements gathering and design work.
- To implement and manage Continuous Monitoring capabilities, so errors can be proactively identified and quickly corrected.
- To proactively incorporate a "fail-fast / fail-often and learn" approach to improvement (i.e. running Chaos Monkey open source code to find and fix resilience issues).
- To focus on creating as much self-service capability as possible to enable developers to spin up environments on-demand that accurately represent the live environment to enable effective testing, so all code is deployment ready as per the organization's release policies.

Benefits for Stakeholders

The following IT stakeholders can benefit from further DevOps knowledge:

- All C-suite IT executives
- Product and process managers
- Service and solution owners
- Enterprise, automation and continuous delivery architects
- Development teams (engineers, testers, developers, security, build, QA)
- Lean IT, Project Management and Agile Practitioners

- Operations teams (engineers, support, analysts)
- Release and deployment managers

CIOs are struggling with a skills gap when it comes to ensuring successful adoption of DevOps within their organizations. The skills required include:

- More T-Shaped skills, with deep technical expertise in at least one area and generalist expertise across many different areas, such as:
    - Business processes and practices
    - IT best practices and how they work together (i.e. DevOps, Agile, Development Lifecycles, Project Management, COBIT, TOGAF for Enterprise Architecture, Lean IT, Agile IT Service Management, etc.)
- Soft skills in areas such as collaboration, teamwork, communication, self-motivation, emotional intelligence, customer focus, etc.

Learn More

Learn more about DevOps training and certifications.

DevOps Training
DevOps Certification Training

Conclusion

DevOps has been around since 2009, and continues to gain momentum each year. The DevOps philosophy and practices allow organizations to integrate their existing IT best practices into a cohesive approach to become more responsive to business demand and to deliver quality IT services to enable business goals.

Don't forget, it's not just about the tools—there's a major cultural shift toward collaboration, silo-breaking, risk intolerance and creating a high-trust working environment (i.e. blameless postmortems). These shifts are essential to a successful DevOps adoption. The DevOps benefits described in this white paper cannot be achieved without committed and visionary leadership. A quality DevOps course will address both the cultural AND technical cornerstones of the DevOps movement.

Good luck on your journey!

About the Authors

Abbey Wiltse

With over 25 years of business experience, Abbey Wiltse is recognized as a world-class strategic process solution consultant who partners with her clients to become more efficient and effective in running their IT organizations. Abbey's area of expertise is in the development of IT Service Management (ITSM) processes and in providing program/project management leadership of these implementations. She specializes in the IT Infrastructure Library (ITIL®) best practice, including the translation of these business process requirements to supporting ITSM technologies.

Nasser El-Batal

Mr. El-Batal is the co-founder, President and CEO of Ahead Technology; a seasoned professional with 28 years of experience in the corporate world, IT Service Management Consulting, and a successful record of managing Information Technology Professional Services, IT Service Management Program and best practices Training and coaching deliveries. He is at ease with strategic, tactical and operational matters associated with the Information Technology industry.