Patterns for Success: Lessons Learned When Adopting Enterprise DevOps

Enterprises that have successfully embraced DevOps are well on their way to accruing real benefits. For every success story, however, there are many more perceived or real DevOps failures. Here's how to make sure DevOps earns a great reputation in your organization.

Executive Summary

While success with enterprise DevOps is never guaranteed, the promise it holds for organizational transformation has never been greater. The benefits of DevOps – a practice that integrates the activities of developers and IT operations to enable a more agile relationship – have never been more possible achieve, including faster time to market, reduced IT costs and lower total cost of ownership. In fact, it can be argued that in today's fast-paced global markets, enterprises either need to adopt DevOps or risk succumbing to more agile competitors.

Generally speaking, the benefits that enterprise DevOps offers the business are broadly measured, using the following hard and soft metrics:

- **Accelerated product/application lifecycles and velocity.** Through automation, real-time problem identification and continuous refinement, DevOps promises to accelerate application development time, decrease maintenance costs and increase ROI.
- **Automation-driven software development lifecycles (SDLC).** Because enterprise DevOps pushes the envelope on automation, it largely replaces time-consuming manual steps, and measures key metrics associated with automation to quantify benefits over time. Metrics include completed projects, reduced project backlogs, improved cost tracking and more.
- **Reduced touchpoints and frictionless enterprise workflows.** In most organizations, multiple stakeholders are involved in enterprise IT, from applications requirements definition, to rollout and maintenance. Many business managers and functional groups participate, as well. All of this adds complexity, cost and time to development. DevOps identifies the touchpoints that exist among these silos and reduces them through pragmatic automation.
- **Enhanced engineering maturity.** By addressing cultural concerns, tools integration and processes used across the software delivery lifecycle, IT professionals become more skilled in using the latest technologies available.

This whitepaper addresses the most common reasons for DevOps failure, and advises on how to succeed with enterprise DevOps, based on our experience, perspectives and insights from our work with large organizations across the insurance, pharmaceuticals, retail and other industries.

Common Strategic Failures

With such clear-cut benefits, why are many DevOps initiatives not making the grade? Based on our work with clients, we have identified seven common denominators of failure in organizations that failed to realize the expected DevOps benefits.

1 Lack of a common definition

DevOps frequently means different things to different people. Even those serving in various technology positions will have their own definition, perception and opinion of DevOps. A lot of this ties back to personal experiences, roles and organizational silos. As a result, it is common to see a significant amount of confusion around what DevOps is, or what it should do in the context of the organization.

We define "enterprise DevOps" as a form of DevOps that includes most, if not all, of the following components: Smart environments, enterprise continuous delivery, enterprise release management, feedback amplification and use of next-generation DevOps technologies (see QuickTake, page 3).

2 Organizational resistance

A lack of clarity and consistency around how the enterprise defines DevOps can lead to noise, chatter and ambiguity. Each of these roadblocks can lead to organizational resistance to change and a failure to successfully embrace DevOps. Some of the common excuses we hear include:

- Our applications are too complex to migrate to DevOps.
- Our applications and processes are already working pretty well; if it ain't broke, don't fix it.
- We have multiple silos within the organization, with each doing its own thing. There's no point in trying to bring these groups together.

3 Cultural issues

DevOps is frequently blocked by cultural issues within the organization. While it may have been defined on paper and presented to stakeholders in PowerPoint decks and spreadsheets at various town halls and meetings, DevOps can still fail to become real. Lack of leadership and commitment are frequently to blame. Without ongoing executive support and progress toward execution, DevOps becomes a non-starter before it even begins.

4 Technology spread

Large enterprises are often complex ecosystems. The technology spread that exists among distributed applications, legacy systems and commercial, off-the-shelf products is prevalent in organizations today. Any effort at embracing DevOps ends up becoming a short-lived Band-Aid solution to a tactical problem.

While technology spread is a fact of life within any large organization, it's also one of the key factors that makes DevOps so exciting. When a mix of mainframes, legacy software and emerging cloud-based digital systems exist side by side, enterprise DevOps offers the potential for inclusion of all technology types and platforms. Implemented correctly, it can deliver on the promises of lower cost of ownership, increased engineering maturity and reduced time to market, among others.

5 Divergent tools

In our more than a decade of helping organizations with enterprise DevOps – even before the term "DevOps" came into existence – we've seen an enormous collection of tools in nearly every customer landscape. Not surprisingly, many tools are practically duplicates of each other in various technical silos, with many of the toolsets doing the same things. Take, for example, the tool chain for a specific function, such as version control for managing source code, where GitHub might be used by some groups and IBM Rational ClearCase or CA Harvest SCM by others. Similarly, when base scripts for managing and compiling software are called for, different teams may well be using a mix of Ruby, shell scripts, Bash, PowerShell or similar systems. Meanwhile, other toolsets are used to store binary code libraries in repositories and to measure the quality of the code that's written.

As new toolsets roll out and gain their own followers, each team develops a vested interest in them. Any effort toward standardization and consolidation of tools quickly becomes an enterprise nightmare, with significant overlap in functionality and wasteful spending on software, people and training. A more pragmatic approach is to work in a structured, disciplined and non-confrontational way, where the problem is first identified and then agreement is reached on a plan to move toward a smaller, universal toolset that is universally adopted in, say, two years.

6 Architectural differences

Even within the same technology stack, it is not uncommon to see architectural variations. Unfortunately, each architectural variation calls for its own specific implementation to make DevOps successful. This adds complexity, time, effort and cost to making DevOps real and meaningful across the architectural variants.

Quick Take

Defining DevOps: Five Key Pieces for Getting from Here to There

To succeed, an enterprise DevOps initiative should incorporate the following five characteristics.

- **Smart environment:** In conventional approaches to software development, a three-step process is followed for software implementation. The business defines its requirements, IT performs the development, testing and support, and operations staffs the data center and watches for problems that may arise.

  With DevOps, the roles are very different. Yes, business managers still work with technology teams to define requirements; but now, IT and operations work together to perform development and testing, using common tools that automate these functions as much as possible.

- **Enterprise continuous delivery:** Once a piece of software is ready for testing, successful DevOps organizations deploy the software in a smart testing environment, using virtual machines rather than physical servers. The virtual machines are frequently provisioned in an on-demand, cloud environment, with automated development and testing.

- **Enterprise release management:** This is where true alignment exists among different technology teams. Characteristics include a common, agreed-upon end date for each release, with detailed plans in place to detect early trouble signs and to seamlessly process release rollbacks when needed.

- **Feedback amplification:** Analytics are required to gain insights into all the event streams, log data, metadata and build histories generated by modern digital systems. Being able to channel this data into a single source and draw real-time insights from the patterns that emerge is a key tenant of success.

- **Next-gen DevOps:** While DevOps may not be new, the ways in which successful organizations develop, test and deploy modern applications are. Fortunately, early adopters of rapid-release technologies, such as micro-services, containerization, the cloud and platform as a service, are well-served by enterprise DevOps.

7 Technical debt

Finally, "technical debt" can make any effort to embrace DevOps like walking on thin ice. By technical debt, we mean poor quality software that may have been rolled out to the organization prematurely some years back but was tested to be functionally adequate at the time. Although architectural and engineering flaws may have been identified, the software was cleared for release with the intent to clean it up later. Unfortunately, the clean-up never happens, and the debt balloons over time.

In the wake of market demands and business pressures, many internal developers strive to get their code pushed into production quickly using the manual processes they know best. Human nature is such that they would continue to slog it out manually rather than change their ways, embrace DevOps and begin automating different process elements.

Predictors of Enterprise-Scale Success

Organizations that succeed with enterprise DevOps share some common practices. Think of these as predictors of success that form the building blocks of enterprise DevOps. If the underlying foundation is missing or weak, organizations will fail in their adoption journey.

Here are the building blocks that we've found will ensure success with enterprise DevOps:

- **A big picture view.** Both IT and business managers have a firm grasp of their company and the markets it serves. Consensus exists among team members as to where they are, where they are going and how fast they must move to get there.
- **A detailed plan.** Unlike many organizations that think of enterprise DevOps as a magic switch – flip it on, and the benefits of DevOps begin to flow in – successful programs are based on agreed-upon plans. They have a medium- to long-term commitment from top management, and the people and resources are in place to make the journey work for everyone.
- **A realistic budget.** No matter how organizations look at their budgets and spending patterns, enterprise DevOps needs seed funding to be successful. Making an initial budget available is one of the best things organizations can do to ensure long-term success.
- **Full executive support.** Enterprise DevOps does not work without executive support and sponsorship. Successful DevOps organizations first secure executive support by defining, budgeting for and staffing key roles.
- **Bottom-up stakeholder buy-in.** As much as executive support is needed, it's almost never a good idea to push down a mandate from above. Successful programs lay the groundwork by building bottom-up stakeholder buy-in at all levels.

Success Patterns to Follow

No two organizations that succeed with enterprise DevOps follow the same path. However, there are six specific patterns, or characteristics, that we've found leaders can customize to their unique situations. Just as objects in the everyday world must follow the laws of physics, organizations that stray too far from any of the following success patterns cannot expect to see great results.

1. **Define a reference architecture.** Successful organizations begin by defining their end-state target operating model (TOM), using a reference architecture. There are numerous examples of reference architectures available, but key pieces include development tools, testing suites and code repositories.
2. **Create implementation blueprints.** Nothing breeds success like success, so defining two or more pilot projects that can be implemented quickly (ideally, no more than six months from idea stage to launch) is one of the best ways to embrace the DevOps TOM. Good pilot candidates are visible and relevant to senior management and bridge the gaps between the current state and the desired state. Anything less is more of a proof of concept than an actual project deliverable.
3. **Use platform-driven automation.** DevOps success hinges on automation. Platforms should readily expose automated features to applications and developers in a standardized and incrementally sophisticated manner. Leaders add progressively more features and functionality onto the platform over time as more applications are onboarded.
4. **Achieve cultural alignment through adoption.** Having a good platform in place is not enough. Leaders recognize that enterprise DevOps fundamentally changes the cultural fabric of the organization. As such, simply telling people to change their ways of working is tantamount to failure. Successful organi-

zations introduce automation first on one platform, convince more people to use the platform for their applications, and reward behavior and culture change over time.

The best programs are based on a shared management vision, in which three tracks are identified and acted upon from the beginning: Culture, technology and adoption. Some ways to speed adoption include shared group meetings, informal brown bag lunches and the introduction of tools to collaborate better.

5. **Drive enterprise DevOps through dedicated, specialist roles.** Technology is important, but it's the assignment of people to clearly defined and established positions that separates winners from losers. At a minimum, we recommend a DevOps champion, typically an executive sponsor at the vice-president level or above; a DevOps evangelist, usually one per business unit; a DevOps platform architect (usually one for each application area); and senior DevOps platform engineers, as needed. Finally, every program needs a DevOps program delivery owner skilled in leading complex projects.

   If individuals within your organization are already playing at least some of these roles on an informal basis, be sure to set up these specialist roles for success in three ways: Clear their desks of all other organization responsibilities, provide them with adequate training and development resources, and ensure they have a clear organizational charter. Remember, enterprise DevOps must be inclusive. Because teams are at different levels of technical capability, successful programs are designed to help all colleagues improve.

6. **Objectively track and measure.** As mentioned earlier, making the benefits of DevOps unambiguously transparent to all will go a long way toward cementing the program's place in the organization. To paraphrase management guru Peter Drucker, "You can't manage what you can't measure."

   When communicating benefits, it's not enough to simply publish reports. After all, metrics can be seen as "death by spreadsheet" in many organizations. By the time the information makes its way to decision makers, it's often out of date. Successful programs automatically extract key data points through DevOps analytics, and use dashboards and other straightforward tools to deliver real-time insights.

Looking Ahead

To date, our team at Cognizant has helped more than three dozen customers succeed with enterprise DevOps. These customers are large enterprises, with all the associated complexities and challenges.

Based on our experience, we conclude with these final recommendations for every organization embarking on an enterprise DevOps journey:

- Address the foundations first.
- Take a multi-pronged approach, creating separate tracks to drive DevOps strategy, build the platform and encourage adoption.
- Create and define specialized, dedicated enterprise DevOps roles.
- Focus on incremental enhancements rather than "big-bang" initiatives that take too long and pose a greater possibility for failure.
- Use early adopters as internal champions.
- Actively provide coaching, evangelization and support.
- Transparently measure and share benefits to all.

Remember, none of this will happen overnight or without the full and ongoing support of management. Yet, with careful planning and commitment, top organizations have already proved that success with enterprise DevOps is not only possible, but transformative to business, as well.

About the Author

Kapil Apshankar leads Cognizant's Technology Product Services Group, focusing on DevOps, Hadoop product development and next-generation product engineering. He brings a rich and diverse Silicon Valley perspective from spending a decade-plus exploring various technologies.

In 2005, Kapil developed a sophisticated methodology that empowers companies to harness the power of multifidelity rapid prototyping, which fundamentally changes the way they build software products. In 2007, he helped build the first-of-its-kind enterprise experimental learning platform. In 2010, Kapil incubated one of the largest distributed Scrum development teams in India. He is a regular speaker at technical events, including O'Reilly Conferences and The Innovation Enterprise Summits. He can be reached at Kapil.Apshankar@cognizant.com.

About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process services, dedicated to helping the world's leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 100 development and delivery centers worldwide and approximately 255,800 employees as of September 30, 2016, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at www.cognizant.com or follow us on Twitter: Cognizant.

World Headquarters
500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277
Email: inquiry@cognizant.com

European Headquarters
1 Kingdom Street
Paddington Central
London W2 6BD
Phone: +44 (0) 20 7297 7600
Fax: +44 (0) 20 7121 0102
Email: infouk@cognizant.com

India Operations Headquarters
#5/535, Old Mahabalipuram Road
Okkiyam Pettai, Thoraipakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060
Email: inquiryindia@cognizant.com

Codex 2393