

# Modernizing IT Operations in the Age of DevOps



Guidance on Evolving  
Skill Sets for the Cloud

---



25 NW 23rd Pl  
Suite 6314  
Portland, OR 97210

Modernizing IT Operations in the Age of DevOps:  
Guidance on Evolving Skill Sets for the Cloud

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

When sharing this content, please notify  
IT Revolution Press, LLC, 25 NW 23rd Pl, Suite 6314, Portland, OR 97210

Produced in the United States of America

Cover design and interior by Devon Smith

For further information about IT Revolution, these and other publications, special discounts for bulk book purchases, or for information on booking authors for an event, please visit our website at [ITRevolution.com](http://ITRevolution.com).

## Preface

In March of this year, we at IT Revolution once again had the pleasure of hosting leaders and experts from across the technology community at the DevOps Enterprise Forum in Portland, Oregon. The Forum’s ongoing goal is to create written guidance to overcome the top obstacles facing the DevOps enterprise community.

Over the years, there has been a broad set of topics covered at the Forum, including organizational culture and change management, architecture and technical practices, metrics, integrating and achieving information security and compliance objectives, creating business cases for automated testing, organizational design, and many more. As in years past, this year’s topics are relevant to the changing business dynamics we see happening across all industries and the role technology has to play within those changes.

At the Forum, as in previous years, participants self-organized into teams, working on topics that interested them. Each team narrowed their topics so that they could have a “nearly shippable” artifact by the end of the second day. Watching these teams collaborate and create their artifacts was truly amazing, and those artifacts became the core of the Forum papers you see here.

After the Forum concluded, the groups spent the next eight weeks working together to complete and refine the work they started together. The results can be found in this year’s collection of Forum papers.

A special thanks goes to Jeff Gallimore, our co-host and partner and co-founder at Excella, for helping create a structure for the two days to help everyone stay focused and productive.

IT Revolution is proud to share the outcomes of the hard work, dedication, and collaboration of the amazing group of people from the 2018 DevOps Enterprise Forum. Our hope is that through these papers you will gain valuable insight into DevOps as a practice.

—Gene Kim

June 2018

Portland, Oregon

*Whenever you are asked if you can do a job, tell 'em, "Certainly I can!" Then get busy and find out how to do it.*

—**Theodore Roosevelt**

## Introduction

DevOps adoption continues to rise as enterprises migrate to the cloud as part of their digital transformation initiatives. The combination of DevOps and cloud infrastructure is disrupting the traditional job expectations of enterprise infrastructure and operations (I&O) professionals. Expectations for I&O professionals used to involve “rack-and-stack” provisioning of servers; executing batch scripts; responding to and closing tickets in ticketing systems; and monitoring the status of any number of tasks, systems, jobs, or environments. Now, however, I&O professionals are being asked to do things outside of their comfort zone. This is being driven by not only management or the business leaders, but also the acceleration of DevOps, continuous delivery practices, and the migration of compute workloads to cloud computing infrastructures.

Gartner predicts that by 2020, more compute power will have been sold by infrastructure as a service (IaaS) and platform as a service (PaaS) cloud providers than sold and deployed into enterprise data centers.<sup>1</sup>

The days of physically provisioning servers and environments are quickly coming to a close. Provisioning cloud environments is done through code and related API calls. As a result, I&O professionals are being pressed to learn coding and software development life cycle best practices in order to integrate into DevOps initiatives. Unfortunately, many organizations are finding that many of their I&O professionals are not able to or willing to acquire this new skill set, forcing these organizations to reduce the headcount in IT Operations and transition the headcount to Development. This headcount shift is a very real but unspoken undercurrent of the shift-left

---

<sup>1</sup> Pettey, Christy. “Top 10 Technology Trends Impacting Infrastructure & Operations.” Hype Cycle Research Methodology | Gartner Inc. October 30, 2017. Accessed May 09, 2018. <https://www.gartner.com/smarterwithgartner/top-10-technology-trends-impacting-infrastructure-operations/>.

paradigm, and it can be very costly, both in terms of dollars and even more so in the loss of knowledge of the core business.

This paper is intended to offer guidance for I&O professionals so they can evolve their skill set and become an integral part of DevOps initiatives and competent in cloud infrastructure code. Indeed, it is the view of these authors that the expertise of I&O professionals is needed in DevOps processes. DevOps is not synonymous with DevNoOps. Though the expertise of I&O professionals is necessary and important to DevOps initiatives, the way this expertise is deployed and used may be different than what has traditionally been the case. I&O professionals may need to learn to be comfortable working as part of a line of business or product-focused team rather than a centralized, isolated, interrupt-driven department.

The intent of this paper is to provide guidance for individuals and functional teams that find themselves in traditional Operations roles that are being disrupted in the current climate. This includes roles in traditional infrastructure, application support, and service management. The following is a non-exhaustive list of these functional areas. These roles will evolve in the coming years as new demands are placed on IT Operations organizations. We also expect to see new roles arise as organizations evolve. Professionals supporting IT in these capacities will likely see parallels with the new requests and expectations in their company.

- Infrastructure
  - ✦ Compute/Virtualization
  - ✦ Network Engineering
  - ✦ Infrastructure Architecture
  - ✦ Storage Administrator
  - ✦ Facility Management
- Application Support
  - ✦ Release Management
  - ✦ Level 1/2 Support Organization
  - ✦ Database Administration
  - ✦ Systems Administrator/Engineering
  - ✦ Security Engineering

- Quality Assurance (QA)
- Centralized Build Tools and Support
- Service Management
  - Change Management
  - Operations Center/Incident Management
  - Helpdesk
  - Vendor Management

There are, no doubt, many roles that fall into the gaps between the examples listed here. In general, if the practitioner is involved in the provisioning and maintaining of in-house or traditional datacenter infrastructure and associated operations, this guidance is intended to provide an actionable path to the technologies and processes associated with DevOps, as well as the technologies involved with the cloud deployment life cycle. Another way to look at it is if you manage or work from a ticket queue of any kind, this guidance is intended to be relevant to you.

It should also be noted that the audience for this guidance also includes the leaders of people in any of the aforementioned roles. No cultural or technology shift can take place without the explicit support of leadership. Leaders should become acquainted with the material in this paper and work with their teams to determine which aspects of the guidance are most relevant to their workforce and how to thoughtfully begin this transformation at both a personal and organizational level.

## Businesses

With the advent of the public cloud, the supply of infrastructure has become virtually infinite. Additional infrastructure is available whenever the company needs it and can be disposed of whenever the company no longer needs it. Where operations were previously constrained by the practical considerations of ordering, racking, and stacking equipment, it now has virtually infinite, elastic capacity at its immediate disposal—or even its automated provisioning and disposal—with features like autoscaling. The limitation on infrastructure is now just cost: how much the company wants to or is willing to spend. The cost-effectiveness of that infrastructure becomes the overriding concern.

An important part of IT Operations in the future, then, is its ability to provision—and, importantly, de-provision—infrastructure cost-effectively. IT operators now ensure that instances are right-sized (not too powerful, not too weak!), that instances are shut off when they are not being used, that appropriate trade-offs are made between less expensive long-term storage and more expensive short-term storage, and that reserved instances and spot instances are used to manage costs. Costs must now be controlled continuously as usage changes, rather than once when equipment is purchased (as it was in the days of datacenters).

Aside from cost, one of the key drivers of cloud usage and DevOps is the business need for speed, whether it is speed to market for new products, speed to providing IT capabilities for use internal to the business, or speed to getting MVPs or capabilities into users' hands to get feedback. It is important that IT Operations not become the bottleneck in realizing this speed as the cloud and DevOps help accelerate the enterprise. As a result, IT organizations are moving toward a self-service model, where those creating product are able to move as quickly as possible because they are never waiting for Operations to provide the infrastructure or services they need. More and more, the role of IT Operations is to create and curate those self-service capabilities and to ensure that they operate reliably, securely, and cost-effectively.

This, in turn, implies a very different type of interaction between Operations and delivery teams: one that is based more on working together to solve problems, rather than on Operations “doing work for others.” Operations specialists can serve as coaches or consultants on operational considerations for delivery teams; they can design and deploy platforms that improve the capabilities of the delivery teams. They relinquish some control as delivery teams use their services as a platform, but they improve their own lives and outcomes for the business.

The changing role of Operations is directly tied to achieving business outcomes: supporting organizational agility and speed, contributing operational expertise to all dimensions of IT delivery, and managing infrastructure cost-effectively in a world of an infinitely elastic supply of infrastructure.

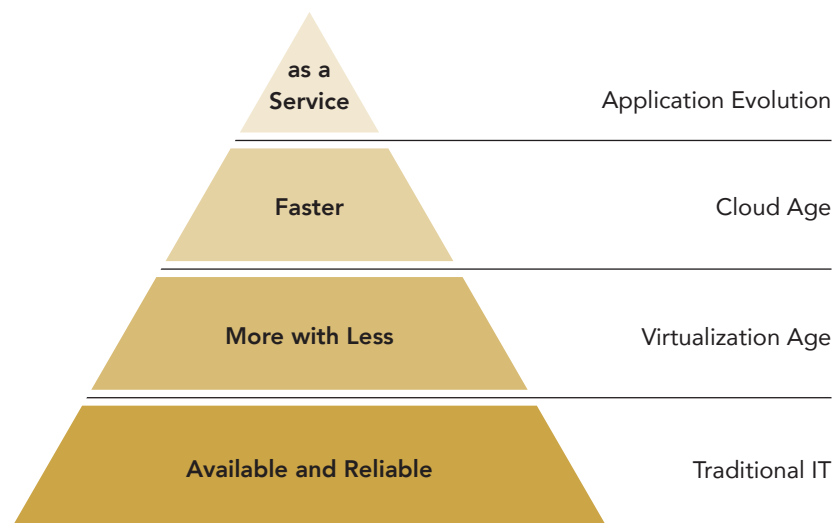
This paper will elaborate on this new role for IT Operations and how to get from the current state to that ideal role. Though the change is dramatic, it can be approached incrementally and in collaboration with other parts of the IT organization and the broader enterprise.

# IT Shifts

Traditional Operations teams must change how they think about their job. It is no longer just about availability, mean time to restore (MTTR), and mean time between failure (MTBF). Instead, it is more about the speed at which the infrastructure teams can deliver the needed resources and services required for the business to thrive while also driving reliability, resilience, and cost-effectiveness.

## Operations Focus Shifts

The early days of IT were hyper-focused on reliability and availability. The transition to virtualization also brought with it a transition to a new focus area: doing more with less. The emergence and success of public cloud providers like AWS, Azure, and GCP changed the expectation for the speed at which IT resources and services could be delivered. This applied pressures on IT organizations and their staff to deliver in new ways that were not comfortable for their operating models without sacrificing the previous focus areas. We now exist in an environment where the methods for developing applications have fundamentally shifted and are in direct response to the increasing pressure to deliver value to the business in shorter time frames. This makes it an imperative for IT to be an engine for value instead of the role of cost center, again without sacrificing the core tenants of stability and efficiency.



***Figure 1: Development and Emphasis of IT Abstraction over Time***



These changes can be extremely daunting for any IT professional or IT leader. It is not easy to find that the work and expertise you have applied up to this point are insufficient for new demands. Fortunately, there are a few key areas that, given regular focus and attention, will prove useful in increasing your ability to change yourself to best meet these new requirements: organizational and technology shifts.

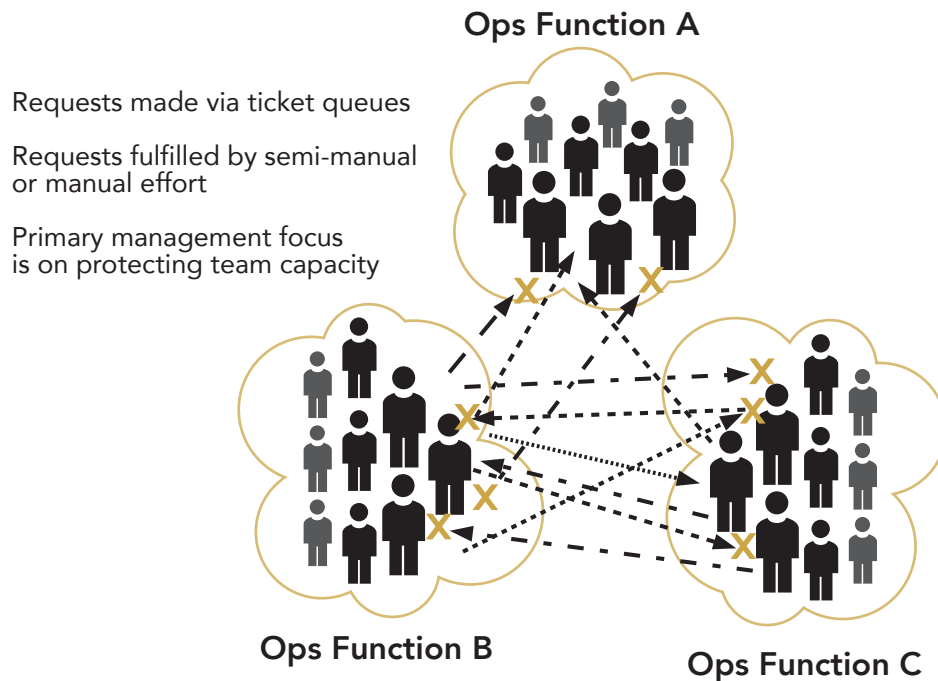
## Organizational Shifts

For decades, the traditional IT organizational model has been based on grouping people by job function or similar skill sets. The most noticeable functional groupings have been major divisions like Development, QA, and Operations. If you look inside those divisions, you'll find that functional grouping is also the dominant organizational pattern. For example, it is common to find groupings inside Operations like Linux SysAdmins, Windows SysAdmins, storage, network, firewalls, NOC, DBAs, security, and more.

A major downside of grouping by job function is that these groups tend to adopt a siloed way of working. In simple terms, a group is said to be “working in a silo” when its members find themselves working in a disconnected manner from other groups. These teams end up working from different backlogs with different incentives and priorities (and often a different management chain).

If work could remain within a single silo, there wouldn't be an issue. Unfortunately, this isn't how work happens. In an enterprise, work has to flow across multiple functional silos to satisfy customers. Silos lead to increased bottlenecks, slow handoffs, miscommunication, tooling mismatches, delivery errors, excess rework, and conflict (usually the finger-pointing type).

When teams are working in functional silos, they tend to become siloed specialist labor pools. Requests are made of these specialists via ticket queues (a common source of bottlenecks and miscommunication). Requests are fulfilled as one-offs through semi-manual or manual efforts. Variability is high. Priorities and context are difficult to gauge. Primary management focus is on protecting team capacity (not the needs of the broader organization). The presence of these functional silos working as specialist labor pools undermines DevOps transformations and leads to today's all too familiar symptom of “everything costs too much and takes too long.”

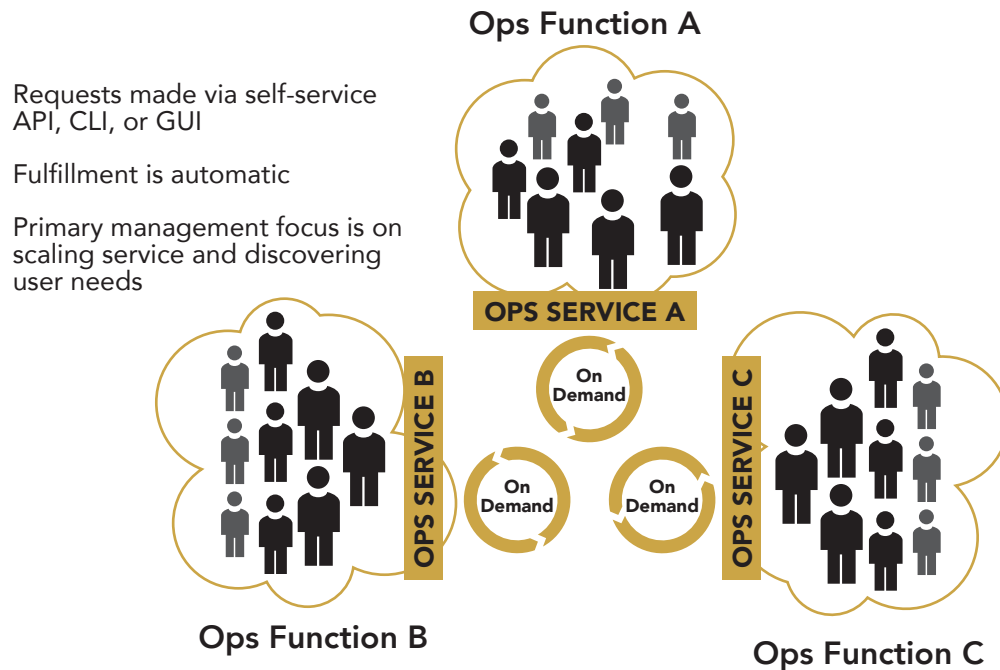


*Figure 2: Siloed Specialist Labor Pools*

Today, high-performing organizations are moving away from siloed specialist labor pools. The new model is to remove handoffs wherever possible (e.g., cross-functional, product-aligned teams) and create internal service providers wherever handoffs remain.

Internal service providers are focused on creating on-demand, pull-based self-service for others in their organization who need their specialist services. In an Operations context, examples of this could be for environment management, network tasks, database tasks, on-demand health checks, monitoring/metrics setup, incident response capabilities, and more. The key is that these internal service providers aren't doing the work for the requestors. Rather, they are building and scaling services so they can stay out of the other teams' way.

Requests to these internal service providers are not dissimilar to those made to externally run cloud services. Requests are made via self-service methods (via API, command-line interface [CLI], or GUI) and fulfillments are automatic and close to real time.



*Figure 3: Internal Service Providers*

## Technology Shifts

The number of tools and requirements applied in modern IT has grown considerably. Parallel to this growth, the market has also seen a dramatic increase in products, both software and hardware, that enable direct management and access through APIs. The combination of APIs with tooling built to consume APIs makes for a powerful combination for IT organizations addressing new demands. In this section we will outline briefly some of the capabilities that have become critical in supporting DevOps and modern IT. These capabilities do not represent the whole of tooling and capabilities that IT organizations need to deliver, but they do provide a straightforward outline of new capabilities that IT must now deliver in a modern environment.

### Configuration Management

Configuration management tools have been adopted almost universally in the application and development side of the house to drive consistency in their applications, operating systems, and environments. Infrastructure and platforms are now delivered with APIs, which have provided opportunities for configuration management

tools to manage that layer as well. Configuration management is a capability that can deliver environment consistency through most of the entire stack and throughout the entire value stream.

Examples of current configuration management tools include open source and vendor products like Puppet, Chef, and Ansible.

### Code and Artifact Management

There is no denying the tremendous growth of new software being built for both customer and internal use cases. Each one of these applications generates massive amounts of data throughout their life cycles. Code and artifact management tools have become critical in organizing, protecting, and distributing that data. This includes supporting code bases as well as managing container images.

Examples of current code and artifact management tools include open source and vendor products like GitHub, GitLab, Nexus, and Artifactory.

### Continuous Integration/Continuous Delivery

Continuous integration and continuous delivery tools have emerged to help both Development and Operations teams move from code commit to code deployment with minimal handoffs, increased efficiency, and better visibility throughout the entire process. Continuous integration/continuous delivery solutions provide the framework for building and automating that process and giving organizations more confidence that their code can quickly move from committed to deployed.

Examples of current continuous integration/continuous delivery tools include open source and vendor products like Jenkins, GitLabCI, and CircleCI.

### Containers

Container technology has emerged in two key areas. First, it has become the preferred framework for packaging and distributing new applications to provide more consistency as well as reduce time at numerous points in the software development and delivery life cycle. Second, it is being leveraged to transition many applications from a Linux operating system to a container package in order to reduce operational efforts associated with patching and OS management.

Examples of current container and container scheduling tools include open source and vendor products like Docker, Kubernetes, Mesos, Azure ACS, Google GKE, and AWS ECS.

### Cloud and Platforms

Easily the broadest capability in this section, IT organizations continue to look at various cloud and platform options to provide the best and most cost-effective solution for meeting business needs. This capability can be delivered on premises, in the public cloud, and often via a combination of the two. It is also likely that many organizations will standardize on multiple platforms to meet their diverse needs.

Examples of current cloud platforms include products like AWS, Microsoft Azure, and GCP.

### Analytics

Visibility has arguably emerged as the most critical of the capabilities listed here. An organization's ability to see and understand what is happening in the environment is its only consistent means of identifying constraints and responding to unexpected outcomes.

Examples of current operational visibility tools include open source and vendor products like Sensu, Datadog, New Relic, App Dynamics, Logstash, and Kibana.

### Serverless

Serverless, often referred to as functions as a service (FaaS), is the most recent abstraction of the evolution of IT infrastructure. With FaaS, server management, configuration, and capacity planning is completely hidden from the user. Serverless code is uploaded to the cloud platform, and the cloud provider dynamically manages the resources. This removes the burden from the developer and makes hosting code a low-friction utility.

Examples of current serverless/FaaS offerings include AWS Lambda, Google App Engine (GAE), and Azure Functions.

# Guidance

So far, this paper has focused on the various challenges facing IT Operations organizations and their members. Now that we have sufficiently provided the context for change, we will provide some initial guidance to help make the transition to being a modern Operations professional.

## Guidance 1: Change of Perspective

The first step to prepare yourself for these inevitable changes is to recognize that in order to deliver the new expectations we must first change how we view what we deliver; we must change our mindset. It is no longer sufficient to simply keep the lights on. That mentality is why most IT professionals and their departments are seen as a cost center to the business. Instead, your perspective on IT needs to be that it is an engine for innovation and business value. This is what the business wants from IT, and it should be what we want as well.

Being mindful to how we respond to projects and user needs will help change the view of consumers that IT is the department of no. That view is typically well founded, as IT organizations regularly say no for a wide range of valid reasons. (The adoption of cloud, including “shadow IT,” is frequently a direct result of not being able to get required resources in a timely manner.) Instead, IT needs to become more inquisitive and interested in the needs of the business and the challenges being solved. By asking pointed questions and utilizing a consultative approach to your work, you have more opportunities to share your expertise in more valuable ways, as well as identify other potential solutions to the challenges your users are trying to solve. You can move from a “No” response to a “Yes, and...” response.

Changing the way we measure IT is a critical component to drive behavioral change. We should not eliminate MTBF and MTTR from our measurements. Those items continue to be important. Instead, they should be de-emphasized slightly with a greater focus on measurements like time to delivery.

We must also make changes in how we set expectations. We’ll go into this in much more detail in Guidance 3, but for the time being, it is important to recognize that these changes will need to become core to how you prioritize work and interact with the developers, application owners, and other consumers.

## Guidance 2: Develop a Code-First Mentality

Given that cloud resources are provisioned and de-provisioned by automated software systems, it follows that software code is the language used to interact with these systems. Most current I&O professionals are comfortable with shell scripts and interacting with a CLI. The problem is that continuous delivery practices, the short cyclical life span of cloud resources, and the scope of cloud environments makes manual or semi-manual Operations efforts inefficient, and Ops quickly becomes a bottleneck.

A development best practice is to tag, track, and test any code changes. In the past, this was virtually impossible to do with infrastructure changes, but the cloud changes this. For the first time, app environment changes can be tagged, tracked, and tested/vetted alongside the apps they host and support. The development practice of using version control systems such as git or GitHub is a skill that I&O professionals will need to acquire.

Remember, rather than logging in and making changes in an ad hoc manner, your changes ought to be edited in code, checked into a version control system, and pushed into your delivery pipeline, where these changes can be tested alongside everyone else's. It's not as important that you deliver something correctly once. It is most important that you repeatedly deliver small changes iteratively.

It should be noted that a code-first mentality is not limited to traditional programming languages. PaaS and IaaS, for example, make heavy use of JSON and other DSLs to describe, provision, and maintain many aspects of a cloud application footprint. Learning to code as a first step might start with familiarization with JSON, YAML, XML, and so on. Simultaneous to that, becoming a familiar consumer of an API via REST calls through a client like PAW, utilizing the new knowledge of JSON (for example), is an excellent next step.

## Guidance 3: Be a Product and Service Delivery Professional

The accelerating move from traditional infrastructure to platforms and cloud services means that many of the traditional roles of IT are decreasing in relevance. Much like the move to software as a service (SaaS) products, this shift causes executives to reconsider their IT investments and increasingly seek to invest in areas that provide business differentiation. Modern business operations view IT as a series



of products that must deliver differentiated value to the organization in order to thrive.

It seems daunting at first to reframe Operations in terms of providing a product, because this shift in thinking makes it clear that you are competing with external products for customers. Rather than being a negative, consider this tension to be a positive aspect of this product focus that motivates you to consider whether your product is providing a solution that is a better choice than competing options. If you aren't able to make a case for using your product over external solutions, that's a strong indicator that you should reconsider your approach to that product.

The good news is that you have some very strong home field advantages that your competitors do not. You're very close to your customers, so things like gathering product feedback, providing education and user support, and even marketing your product are much cheaper and have much shorter feedback loops, giving you an opportunity to iterate quickly to meet the needs of your customers. You are able to design your product offerings in a way that uniquely addresses the specific needs of your organization in ways that would be edge cases if you were an external provider.

It is important to note that you are always in competition with external service offerings (e.g., AWS) and you must, therefore, ensure that you are presenting your services in the same modular, easy-to-consume way that those providers offer.

## Guidance 4: Guidance for Leaders

No shift, whether cultural or technological, can take place without the support and involvement of an organization's leadership. The role of many technology leaders is to prepare their teams for significant change occurring either locally within the teams and organization or externally through industry initiatives and best practices.

It's important for the leader to keep in mind that the people on their teams have a real emotional connection to the work they do. Indeed, they often self-identify with that work. When a disruptive change results in the redefinition of that work, it is no longer a simple case of training or re-education. Team members need to find purpose in the work they do as they master it.

As a leader seeks to enable a transition into a DevOps model, they first need to become familiar with the specifics of DevOps (the guidance in this paper and the resources included in the reference section can help). Next, have a series of conversa-



tions with the impacted roles and teams about a thoughtful plan to train people into the new technologies and processes associated with DevOps. Again, it's important in this work that the leader keep in mind that this is not just an educational transformation; it's a personal, emotional, and cultural change, and teams need to feel like they have ownership of the process rather than feel like victims of it. Look for ways to align past passion with future work and responsibility.

A leader seeking to shift to a DevOps model should also work to decompose the work into tasks and goals that are attainable by the teams given their current workload. It's imperative that the steps teams take toward DevOps not only clearly benefit from their involvement but also can be seen as something they can succeed at. The initial disposition of some may be to show that this new methodology will (and should) fail. This is a natural reaction to large-scale change and should be anticipated.

## Conclusion

*It is not necessary to change. Survival is not mandatory.*

—quote often attributed to **W. Edwards Deming**

Traditional enterprise IT professionals can be nervous about the evaporation of physical infrastructure and the technology shifts to software, cloud, containerization, and serverless infrastructure. However, as has been discussed in this paper, there is hope! This new paradigm does not need to be a path to extinction. Instead, it should be seen as an opportunity for evolution. Business leaders and IT Operations teams should recognize this pivotal moment of change, embrace the opportunity, and lean in to the new, more integrated organizational models. They must become the champions of the revolution and help their fellow technologists embrace the tools, platforms, and new operational models that will help propel their organizations into the future.

## References

- Atwell, Josh. “Choosing Infrastructure Extensibility to Accelerate DevOps Success.” *TechSpective*. March 01, 2017. <https://techspective.net/2017/03/01/choosing-infrastructure-extensibility-accelerate-devops-success/>.
- Atwell, Josh. *Using Infrastructure as an Accelerator of DevOps Maturity*. NetApp Presentation, <https://onlinexperiences.com/Launch/ASPortal.htm?ESH=23D54A1A-4BB5-4B56-AFDF-EEA9ADE8ED1D>.
- Beyer, Betsy, Chris Jones, Jennifer Petoff, and Niall Murphy. *Site Reliability Engineering*. Sebastopol, CA: O’Reilly Media, 2016.
- Cox, Jason, Chivas Nambiar, Paula Thrasher, *Continual Learning: Building a DevOps-Inspired Career Track*. Portland, OR: IT Revolution, 2017. <https://itrevolution.com/book/continual-learning/>.
- Orban, Stephen. *Ahead in the Cloud: Best Practices for Navigating the Future of Enterprise IT*. CreateSpace Independent Publishing Platform, 2018.
- Pettey, Christy. “Top 10 Technology Trends Impacting Infrastructure & Operations.” Hype Cycle Research Methodology | Gartner Inc. October 30, 2017. <https://www.gartner.com/smarterwithgartner/top-10-technology-trends-impacting-infrastructure-operations/>.
- Roberts, Trevor A., Josh Atwell, Egle Sigler, and Yvo Van Doorn. *DevOps for VMware Administrators*. Upper Saddle River, NJ: VMware, 2015.
- Schwartz, Mark. *The Art of Business Value*. Portland, OR: IT Revolution Press, 2016.
- Schwartz, Mark. *A Seat at the Table: IT Leadership in the Age of Agility*. Portland, OR: IT Revolution Press, 2017.
- “Skills Framework for the Information Age (SFIA).” IT Governance GOVN - SFIA Foundation. May 08, 2018. <https://www.sfia-online.org/en>.
- Stroud, Robert. “The Future of Operations: A Developer.” Forrester. July 6, 2017. [https://go.forrester.com/blogs/17-07-06-the\\_future\\_of\\_operations\\_a\\_developer/](https://go.forrester.com/blogs/17-07-06-the_future_of_operations_a_developer/).

## Resources to Grow Your Modern Technical Skills

- Analytics (e.g., Splunk, DataDog, Grafana, ELK)
- Cloud/PaaS (e.g., Amazon Web Services, Microsoft Azure, Google Cloud, OpenShift, Pivotal, CloudFoundry)
- Code and Artifact Management (e.g., GitHub, GitLab, JFrog, Nexus, Visual Studio Team Services)
- Coding (e.g., Python, Go, Java, Visual Studio)
- Configuration Management (e.g., Puppet, Chef, Ansible)
- Containers (e.g., Docker, Kubernetes)
- Continuous Integration/Continuous Delivery (e.g., Jenkins, CloudBees, TravisCI)
- Monitoring (e.g., Nagios, Prometheus, OneView)
- Operations Management (e.g., PagerDuty, Rundeck, VictorOps)
- Periodic Table of DevOps Tools (<https://xebialabs.com/periodic-table-of-devops-tools/>)
- Release Automation (e.g., CA Automate, Electric Cloud, IBM, Xebialabs, Visual Studio Team Services)
- Testing (e.g., Blazemeter, SauceLabs, Visual Studio Team Services)
- Workflow (e.g., Atlassian [Jira, Trello, Confluence])

## Contributors

- Josh Atwell, Developer Advocate and Solutions Marketing Lead, DevOps, NetApp
- Jason Cox, Director, Platform & Systems Reliability Engineering, Disney
- Damon Edwards, Chief Product Officer, Rundeck
- Ron Forrester, VP, Nike.com Engineering, Nike
- Jayne Groll, CEO, DevOps Institute
- Mark Imbriaco, Global CTO, DevOps Transformation, Pivotal
- Mark Schwartz, Enterprise Strategist, AWS
- Scott Willson, Director Product Marketing Release Automation, CA Technologies