



EBOOK

IT Perspectives: Getting Started on a DevOps Platform



NetApp®



Table of Contents

Introduction >>

Cultural Change >>

Vision and Goals >>

Vertical Approach >>

People and Process >>

Organizing the Team >>

Continuous Operating Model >>

Results and Lessons Learned >>

Introduction

The most common question that often comes up with those interested in DevOps is, “Where do I start?” We started by pulling together a small team of DevOps visionaries to define whom to serve and what services to provide. We choose to start small, in phases, to build the capabilities and tools needed to accelerate our DevOps journey, from CI/CD to containers, automation, and cloud.

This eBook covers how NetApp got started by acknowledging DevOps as the cultural change and organizing the team around roles, platforms, and delivery milestones. Key to our success has been thinking of DevOps like a train that rolls down a never-ending track, taking on new cars (features) and passengers (apps) with no end in sight.

— **Michael J. Morris**, Senior Director, NetApp IT

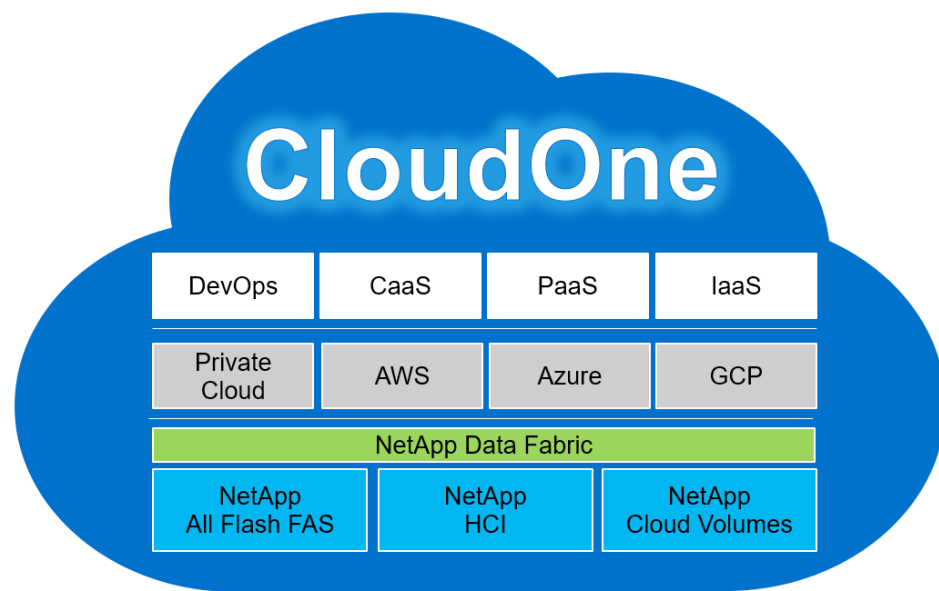


Getting Started with DevOps

DevOps is part culture, part process, and above all, the combination of developers and operations teams working in a coordinated, efficient, and automated way to rapidly deliver production software changes without disruption. Two years ago, NetApp began its journey with a vision to build a DevOps platform to provide automation and CI/CD release models the application development teams need to create cloud native applications using micro-services architectures running in containers.

We call the platform “CloudOne” as it delivers one consistent user experience, irrespective of the cloud destination, either private cloud or public cloud. Today we provide DevOps as a Service and Infrastructure as a Service and will be offering Containers as a Service and Platform as a Service in the next 12-18 months.

Our DevOps journey has just begun—and it has not been perfect—but we are moving in the right direction. I want to share how we got started.



Powered by NetApp® and IT Customer-1

Cultural Change

Traditional approaches and governance methods often impede DevOps as software developers want—and are rewarded for—speedy change delivery, new features, new technology and innovation. On the other hand, Operations wants—and is rewarded for—stability, predictability, control, and above all, non-disruptive operations. With a DevOps approach, you must break out of the traditional mode and get the developers and ops staffs to work collaboratively on projects from start to finish. It is a cultural change with the challenge of bridging disparate wants to allow rapidly delivered production software changes.

To change the culture, one of the first steps we took was to standardize and automate. We intuitively knew Ops likes the predictability and simplicity of standards while developers like the speed provided by automation. We automated everything possible, including standardized processes to speed things up and remove human error.

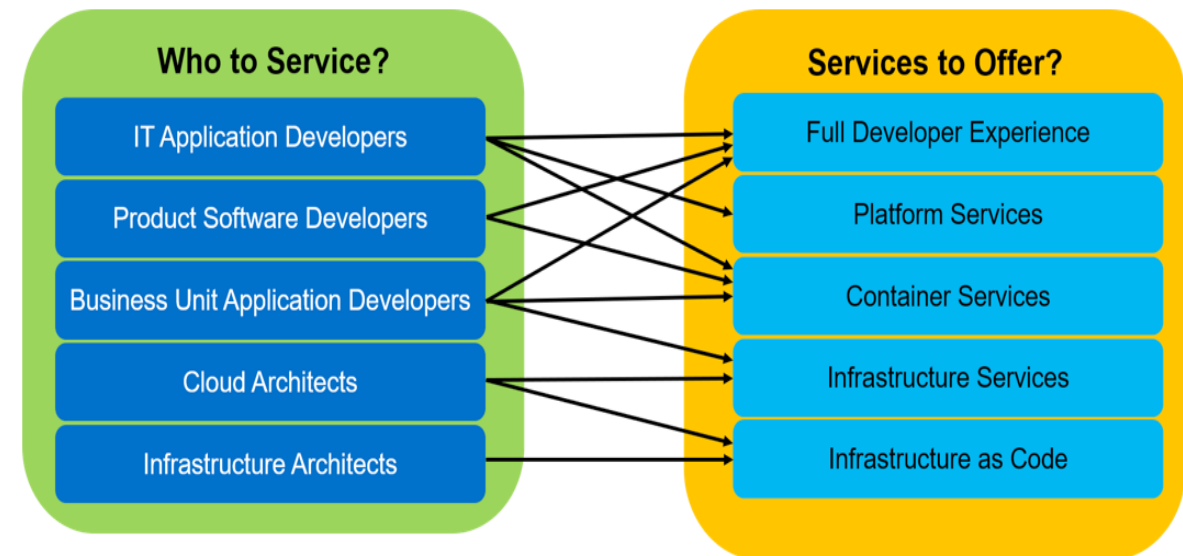
We decided to integrate the Dev and Ops teams to change perspectives, experiences, and motivations. It turns out developers can do Ops and own what they create and release. At the same time, Ops can be part of the development process. Ops processes, resolutions, fixes and improvements became part of the development delivery backlog. This improves production software stability and prioritizes new feature delivery. It also helps make developers own the ramifications of the changes they deliver.

Vision and Goals

As the IT organization inside NetApp that runs over 350 business applications, we have development teams responsible for making changes, updates, and upgrades to applications. This requires many change orders and processes. The changes tend to be large, so we need to minimize the possible disruptions to the business as part of change control. This takes time and resources. To create a future endpoint that the teams could conceptualize and understand, we created a service vision.

Who are we trying to serve? We decided to build a DevOps platform with the necessary automation and standardization to allow for frequent small production changes to our business applications. We wanted to focus our developers on writing code and releasing application changes, not dealing with other IT services. We also wanted to follow the DevOps principles of moving fast and making small changes to minimize deployment risks. The DevOps service was focused at IT application developers, product software developers, and business unit software developers.

What are the services we wanted to provide? To serve this community, we needed to provide a full developer experience, platform services, container runtime servers (CaaS), and IaaS. For example, the IT Application Developers want services that provide a full developer experience along with platform services and container services. Cloud Architects, on the other hand, are more interested in infrastructure services and Infrastructure as Code.



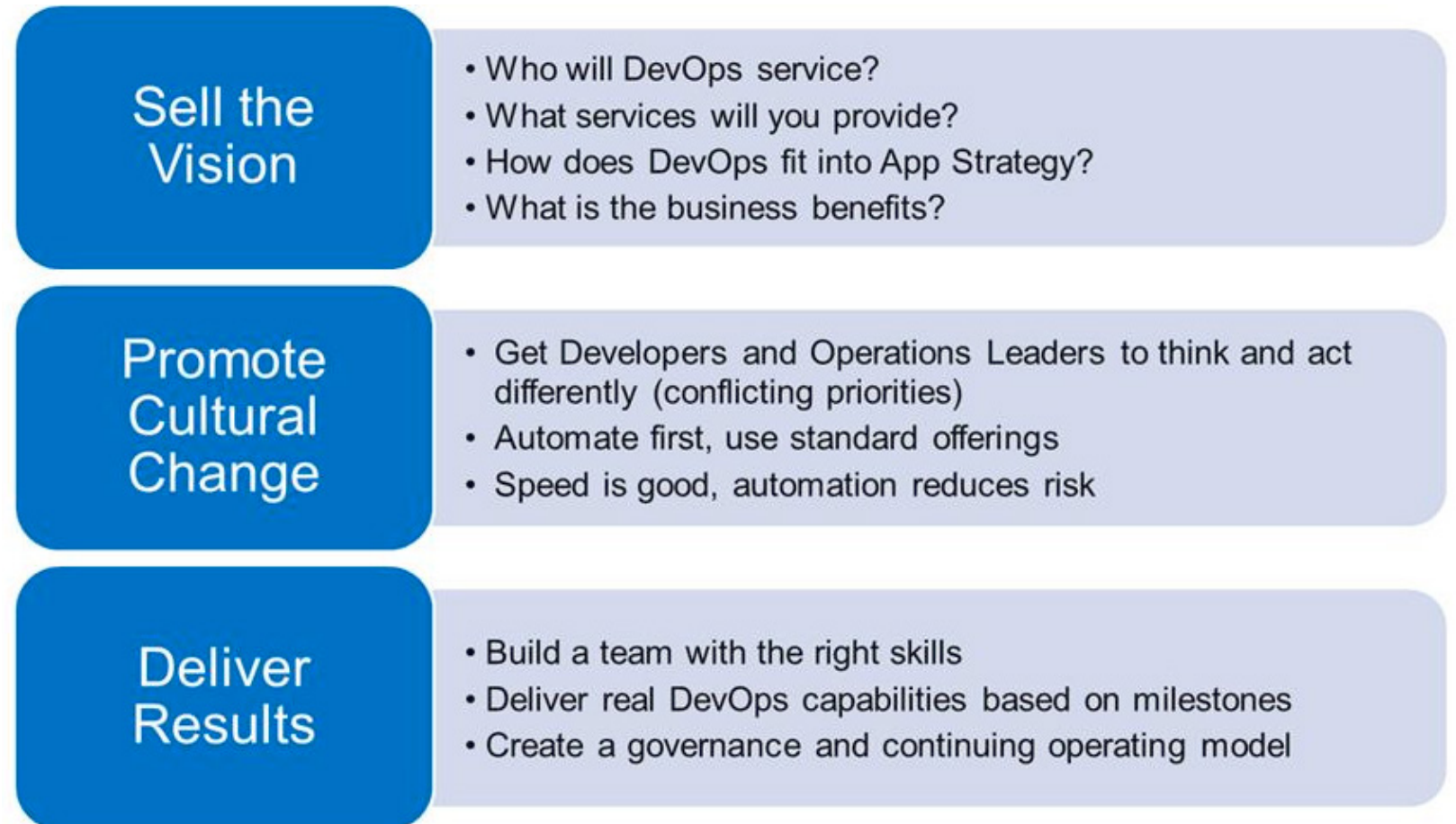
As an end-to-end solution as a service, CloudOne provides speed, control, standardization, and cloud capability. We built everything around four guiding principles:

- “It’s a service” with a self-service catalog of pre-defined capabilities to promote simplicity and supportability.
- Focus app developers on software development, not infrastructure and automation.
- Security built-in beyond traditional network security controls DevSecOps.
- Private or Public Cloud use of containers to deploy applications components at any locations while remaining cloud agnostic.

Identify and Empower a Champion

Given DevOps is a new way of looking at traditional practices, tools, and philosophies, it needs a champion or advocate. I serve as our champion and am empowered to change the culture, sell the vision, and deliver results. It helps that I am a member of the CIO leadership team and have visibility into key business strategies and roadmaps, along with access to key decision makers.

To change the culture, it was recognized we had to get developers and operations teams to think and act differently, despite conflicting priorities. I advocated that we first automate using standard offerings. The mantra became, "Speed is good. Automation reduces risk." To sell the vision, we identified who DevOps would serve and what services would be provided. It included how DevOps fits into our application strategy and the business benefits. To deliver results, we had to build a team with the right skills, set milestones to deliver real DevOps capabilities, and create a governance and continuing operating model. Without a champion, CloudOne would not have come to fruition.



Our Entry Point: Tie to Application Portfolio

Before we could build a DevOps platform for application development, we needed to understand our application portfolio. Our Enterprise Architecture team owns the application portfolio and they have used the Gartner TIME model to classify our applications into four categories: T=Tolerate, I=Invest/Strategic, M=Migrate, E=Eliminate.

Tolerate represents those applications that NetApp IT owns and we are not going to do more with in the future. We're just going to keep them running until such point that we can get rid of them.

Eliminate are applications that represent a business process that isn't valuable to NetApp anymore. They may be old and/or unsupported, allowing us to sunset the application and free up resources.

Invest-strategic represent good business processes running on good applications and good platforms. IT aims to do more with apps that fall into this category to deploy new capabilities or enhance current functionality.

Migrate applications represent valid business processes that are needed, but the apps that automate them may be old, unsupported, or operating on end-of-life infrastructure. These apps require an investment to migrate functionality to a different application.

Our EA team looked at the Invest and Migrate applications and determined that some applications just need new infrastructure.

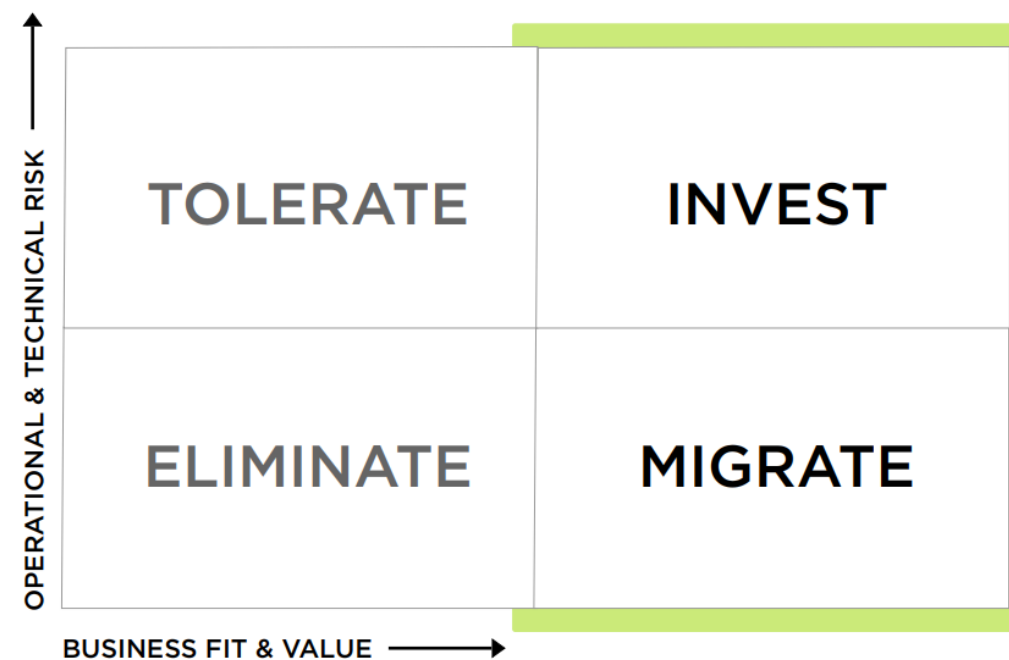
For this we use Infrastructure as a Service (IaaS). Some applications need to be completely rebuilt from the ground up. That's where a DevOps platform would come into play to rearchitect applications into containers using microservices.

We then determined the level of "cloudiness" required. This is where our next level of classification, called the 5Rs (introduced by Gartner), comes into play. Each of these Rs (rehost, refactor, rearchitect, rebuild, replace) has a

different level of effort associated with it, and is aligned to the strategic value that the application provides.

In a nutshell our strategy is to:

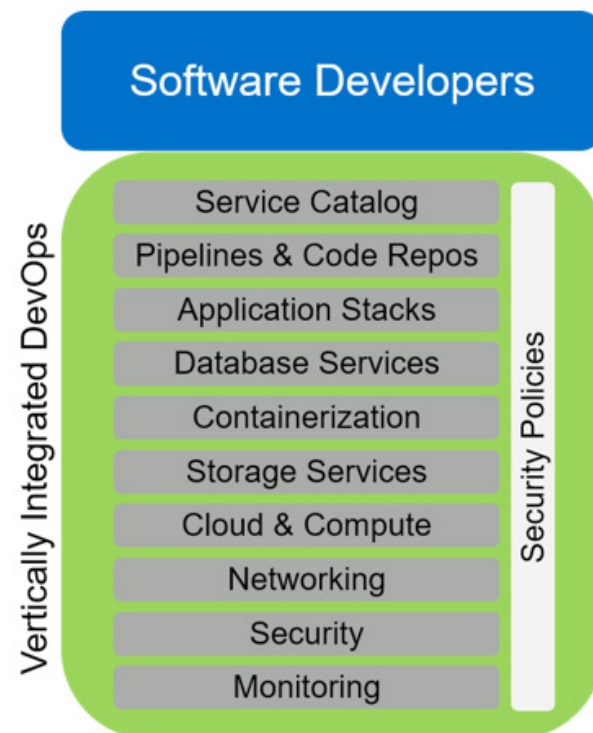
- Keep the business running by leveraging SaaS (Replace) for common business processes, i.e. Commodity Services and System of Record type applications.
- Transform the business by leveraging PaaS and DevOps for strategic and innovative applications.
- Leveraging IaaS as the bare minimum for all packaged apps to gain Refactor and Rearchitect efficiency if possible.



Vertical Approach

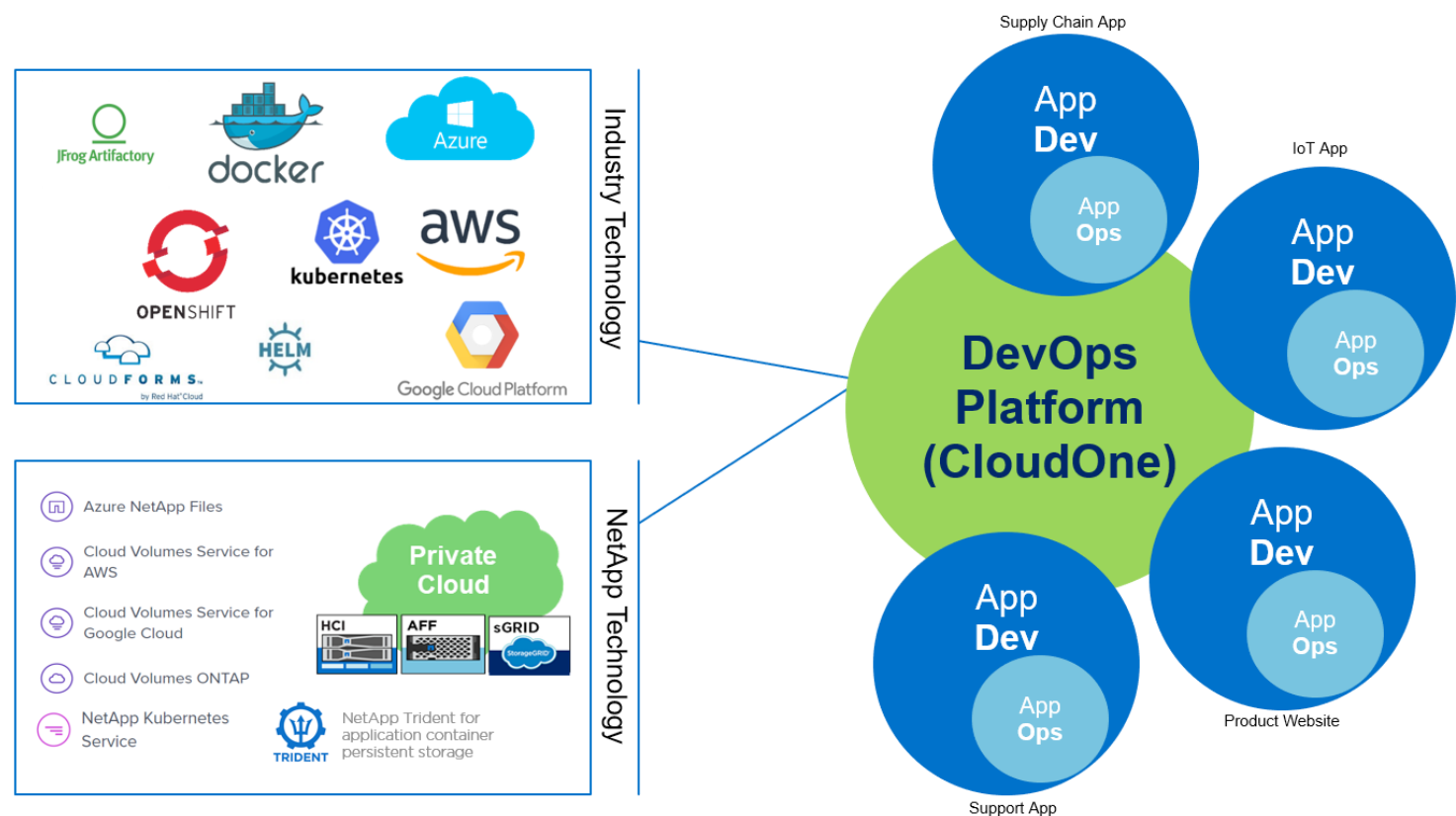
Instead of building horizontally and making one layer of the technology stack DevOps ready, e.g. storage-as-code infrastructure, we chose to build a vertically integrated, on-demand full-stack development environment. It includes a service catalog, application stacks, database services, containerization, storage services, networking, security, monitoring, cloud and compute, and pipelines and code repositories.

It is managed by a matrixed “Platform Team” to remove the burden from DevOps teams on developing and running automation and managing tools and cloud environments. DevOps teams can focus on software production, not tools and technology.



Technology Choices

If you look inside CloudOne, there are products that you probably have heard of like OpenShift, Docker, Kubernetes, and the cloud providers themselves like AWS and Google Cloud. NetApp products fit in there too, whether it is NetApp's Kubernetes Services or Cloud Volumes or Trident for storage provisioning through Kubernetes. In our Private Cloud we use HCI, StorageGRID, and All Flash FAS. These NetApp products along with the industry products make our platform better, which ultimately makes our developers more productive.



People and Process

As with any IT initiative, technology is just one component. We also have process and people considerations. With DevOps we needed effective teams to accelerate developing, deploying and maintaining apps to our organization. We needed processes for code and binary management, monitoring and analytics.

Tiger team creates vision and concept: Two years ago, when we were challenged by our CIO to build a future, next generation vision of our data centers, we were certain of one thing: no longer is there a center of data. Data is no longer centralized in a brick and mortar data center. It has become uncentered and resides with SaaS providers, hyperscale clouds, private clouds, and co-location facilities. To us, next generation is a cloud agnostic, software controlled and orchestrated development platform that allows developers to build and run cloud-aware applications using advanced methodologies like DevOps and CI/CD delivery models. After several weeks of debates, we united on a vision for CloudOne as a DevOps Platform.

DevOps roles: To ensure our platform served the needs of our development community, we identified three roles affected by the actions, objectives, and policies set for the platform. They are the Developer Lead, the Developer Tools Lead, and the Operations Lead.

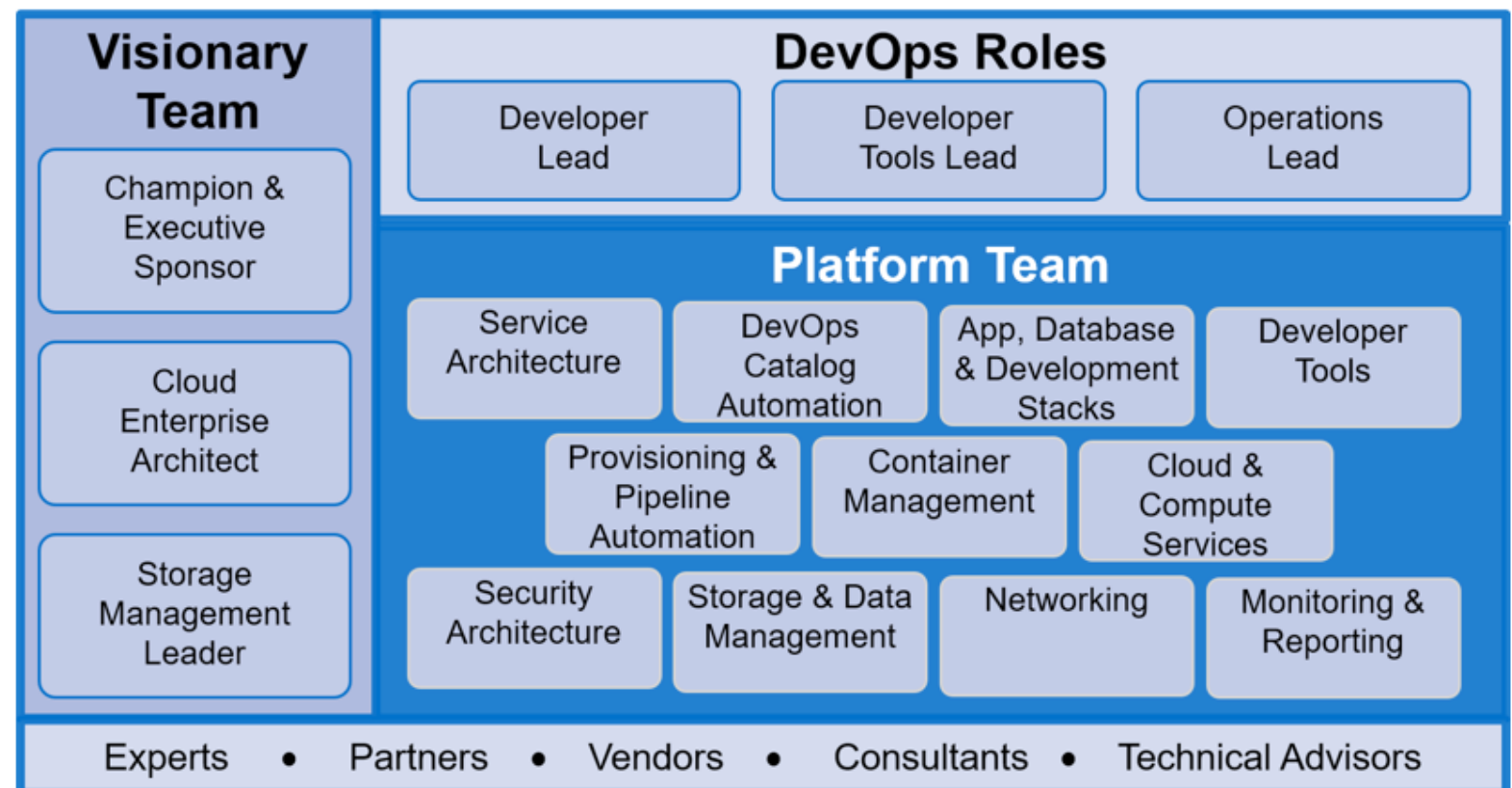
Platform team: It took six months to socialize the concept and generate support. At the same time, we reviewed our existing IT organization and identified the core teams needed to deliver the platform. An Agile “Product Owner” was identified on each of the teams that included:

- Service Architecture
- DevOps Catalog Automation
- Application, Database, and Development Stacks
- Developer Tools
- Provisioning & Pipeline Automation
- Container Management
- Cloud & Compute Services

- Security Architecture
- Storage and Data Management
- Networking
- Monitoring and Reporting

Eventually we will introduce the role of Site Reliability Engineer to champion change and assist with availability, release and performance management.

Other experts: We recognized other experience would be required. To fill gaps, educate and train us as DevOps novices, we relied on industry experts, partners, vendors, consultants, and others with hands-on technical skills to fill knowledge gaps.



Organizing the Team

With the teams and roles identified, we began defining specific requirements based on the vision. Agile “Product Owners” were created with accountability and responsibility for key areas. These Product Owners were assigned delivery milestones based on the requirements. The milestones were delivered using Agile practices, i.e. frequent inspection and adaptation. While CloudOne started as a project, today it has transitioned it to a continuing operating model.

Milestones

1) Define and document vision: It took us approximately six months to socialize the vision and get buy-in from the key stakeholders who would use the service. This group consisted of IT application developers, product software developers, business unit applications developers, and cloud and infrastructure architects.

2) Document services to be provided: The platform would provide a full software developer experience including a complete automated app-stack deployment. Once the platform was in place, we would then add other services like Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Container as a Service (CaaS).

3) Set requirements for each “product

owner”: For the identified services, each product owner of the service identified tasks that had to be completed to ensure success.

4) Define the toolchain and cloud architecture: The platform team defined the toolchain and automation to build a service-oriented architecture.

5) Set a Minimal Viable Product (MVP) launch date: Go live date established for first release of the new platform.

6) Complete design solution: Prototype of the preliminary platform finished based on identified requirements.

7) Complete Proof of Concept (PoC): Testing conducted of platform to validate it delivered as promised.

8) Complete MVP build: Build efforts concluded with core functionalities.

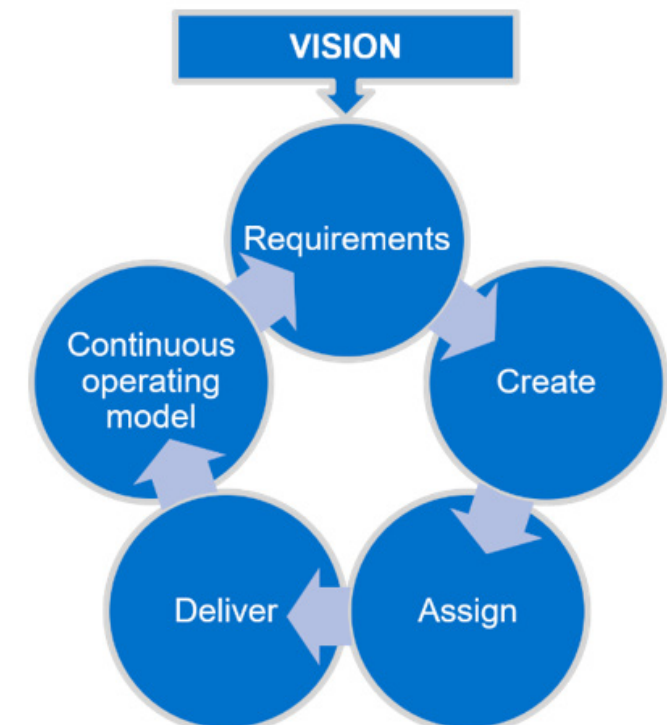
9) Complete MVP test: Test and measure developers’ response to core functionalities.

10) Launch MVP: First release of platform.

11) Continuous operating model: Begin transitioning away from project approach to ongoing operating practices.

Future milestones were set once our processes were automated across the Software Development Lifecycle (SDLC) as part of MVP. Our next level of DevOps

maturity is to move from continuous delivery to continuous deployment. With the Continuous delivery process today, we must pause to obtain some approvals. With continuous deployment we want to automate all the way to production. This requires some advanced capabilities from an application delivery perspective like blue green releases, canary releases and automated rollback capabilities. Those types of automations we don’t have yet. We’re building those right now, and when we get to that point, then we can eliminate all the manual approvals as part of the application delivery process. Once developers are done, we want changes to be automatically deployed to production.



Continuous Operating Model

DevOps is not a project that starts and ends. It's a practice that never ends and must be maintained, improved, and operated indefinitely. To support it, you need an operating model. Think of it like a train that rolls down a never-ending track, taking on new cars (features) and passengers (applications) with no end in sight. We set goals (using milestones) to focus the teams and iterate toward the goals using Agile practices to retain flexibility. We're constantly going to be adding new capabilities to our platform so we must keep key engineers, architects, developers and ops leaders engaged throughout the planning, governance processes and deliverables. We have created an application supply chain to bring new applications onto the DevOps platform and established a leadership Steering Committee to guide the DevOps program.

What to do?

- Set major maturity improvement goals
- Define a roadmap to deliver new DevOps capabilities
- Establish *Supply Chain* process for applications
- Iterate through specific deliverables using Agile practices

How to do it?

- Keep key engineers engaged
- Establish regular governance meetings to track deliverables
- Hold technology architecture meetings to address technical issues
- Institute Steering Committee of key stakeholders

Results

As a result of our automation efforts, we have significantly reduced the time to provision environments in CloudOne compared to the traditional approach (See *table*).

Lessons Learned

Separate platform ownership: By identifying the Platform Team as the “owners” of the CloudOne platform, we discovered two things. First, having a different team own the platform meant the DevOps teams can truly focus on software production, not tools and technology. We also learned new process automation, platform technology, and clouds can be run and delivered separately from DevOps software development. This opened career path options for our infrastructure engineers.

Getting the base environment built and ready. It was a year-long effort from initial vision to first release. It took time because of the complexity to get DevOps workflows automated across developer tools, platform software, and infrastructure.

Don't build a DevOps ghost town that the development teams will not use. By taking the time to educate the development teams, we were able to establish a more trusted working relationship with open discussions on ways to create a supply chain of new and existing apps to build or migrate to CloudOne.

Provisioning Comparison	Traditional Approach	DevOps Environment
# of tickets needed to provision new application environment	7 tickets to create an environment including: <ul style="list-style-type: none">• Demand initiation• Domain name• VM creation request• Firewall changes• Load balancer changes• SSL certification request• Middleware installation	1 self-service request* Hostspace using a self-service portal <i>*Customized domain name requires additional tickets</i>
Typical Time needed to complete request	1-2 weeks for normal request	2 hours
# of environment provisioned in past 6 months	40 environments	72 environments
Code Build Automation (CI)	Manual build or using exist build automation	Provided as default; part of environment provisioning automation
Code Deployment (CD) Automation	Manual deployment or update takes 1-2 weeks of additional effort	Provided as default; part of environment provisioning automation
Environment Scaling	1-2 weeks of manual effort	Software-driven elastic scaling
Security Scan	Ad hoc	Provided as default; part of environment provisioning automation
Access Provisioning	Submit manual tickets to added/modify access	Provided as default; part of environment provisioning automation
Storage Provisioning	Volumes mounted part of VM provisioning	Provisioned dynamically during deployment

Summary

Like many other large enterprises, our DevOps journey has just begun. We aligned our vision to an application strategy of 70% SaaS with 30% build of apps on an internal DevOps platform. We took a vertical approach to build “up” with a full-stack development environment that provides all the cloud services, automation, and CI/CD release models the application development teams need to build cloud native applications using micro-services architectures running in containers. We have integrated leading third-party tools with NetApp technologies to offer a service for DevOps, Infrastructure, and coming soon, Platforms and Containers. Our journey began two years ago, and I know one thing for certain: we’re moving in the right direction! •

For information on NetApp DevOps solutions, visit [NetApp.com/DevOps](https://netapp.com/devops)

