Exploring DevOps:

Charting a Course for the Enterprise

CloudBees®

Charting A Course For Enterprise DevOps

DevOps experts Sacha Labourey and Nigel Willie take an in-depth look at how DevOps has evolved to the innovative and collaborative DevOps culture of today.

There are three chapters in this eBook, as outlined below.

1. Enterprise DevOps:
An Introduction

Before charting your DevOps journey, it is imperative to be aware of why previous ways of working were abandoned. Let's ensure those challenges are mitigated as the DevOps program gears up.

2. Enterprise DevOps:
Understand Your DevOps Starting Point

Given an inherited technology estate, along with embedded process and practices, it is sometimes difficult to know where to start. Creating a capability matrix and defining your current technology stack can serve as a foundation.

3. Enterprise DevOps:
Context is King

Technologists often understand the technical landscape they work in, but in today's environment it is critical that the technologist understand their product from a business perspective to prove value.

1. Enterprise DevOps:
An Introduction

If you are reading this eBook, we can assume that you are interested in the process required to roll out a DevOps initiative across an enterprise. If that is the case, we'll also assume that you wish to successfully embed DevOps into the DNA of your enterprise and drive a transformation in software delivery. In our view, companies who have made a success of DevOps have recognized it involves cultural change across the organization and not merely the latest in a line of initiatives driven from Head Office.

That said, your management will have objectives and a reward structure in place and it is critical you understand this if you are to be successful. You could consider delivering any transformational program as being similar to sailing into the wind. You can't head straight to your destination but must tack. You need to know your ultimate destination, but you also need to know that which needs to be delivered for your key stakeholders — which may not strictly be strategic.

We should also state that we have a very simple personal definition of DevOps and it is focused on outcomes, not capabilities. Successful DevOps provides the ability to deliver discrete technical artifacts rapidly to production to drive value to your business.

Nigel Willie, an expert DevOps practitioner, will now wind the clock back over twenty years to his early days in the industry:

"When I started work, there was a small team of us sitting together around a bank of desks. We had someone who represented the business, somebody who wrote the screens (albeit old 3270 green screens for branch staff), a developer, an individual who wrote the functional specifications, someone who wrote the technical design specifications and a tester. The individual writing the screens would ask the business representative to wander around the desk and check if the green screen looked how they wanted it to look.

"Before any of us knew the correct taxonomy, we exhibited several of the characteristics of Agile and DevOps, albeit with none of the automation. In addition, the assets we delivered were more monolithic and the cadence was far lower than would be acceptable today."

So, what happened?

Business engagement (or lack thereof): A colleague once said to me, "First the business moved us into the basement, then into another building, then another city and finally another continent." In many companies, IT lost contact with the business and became a cost center that delivered solutions rather than being a close partner. It is critical that those of us in technology become better at communicating with our business and understand what the asset we are accountable for does in their eyes. I see far too many technologists with little business contextual awareness.

Silo-ization: Within larger enterprises, there was a move to monoskilled pools. A large pool of developers, a pool of testers, a pool of business analysts, a pool of DBA's and so forth. This moved us away from a product to project approach, with a pool of labor brought together for a specific task. I lived through this process and know that it was driven by IT itself, or at least by CFO's within the IT part of the organization. The rationale came from the idea that product teams sometimes had downtime when demand for new features was not pressing. By creating pools of labor, the IT organization could react more rapidly to business priorities.

Why care about history?

It is our belief that anybody commencing a DevOps initiative should be aware of why similar ways of working previously were abandoned and take steps to ensure those challenges are recognized and mitigated against as the DevOps program gears up. It is imperative that the business is fully engaged and committed as key stakeholders in the value that can be gained from DevOps. You need to avoid the view of IT being a cost and ensure that you can clearly articulate value. Ideally, your business should start to articulate the value they see from DevOps to others.

Additionally, if your company previously moved into silos, an awareness of the rationale behind this is crucial. For example, look to put a process in place to identify when business demand for product enhancements starts to increase or decrease to ensure you avoid the perception that resources are not being optimized.

NEXT: Understand Your DevOps Starting Point >>

Before any transformation within your organization begins, we recommend that you:

- Understand what outcomes you are trying to achieve.
- Understand what objectives your board has and identify which short-term objectives add risk to the long-term strategy.
- Understand why your company looks like it does now, how you got here and what motivators led to the current culture and environment.

2. Enterprise DevOps:

I Wouldn't Start from Here — Understand Your DevOps Starting Point

There is an old joke in the UK about a couple of city dwellers driving through the countryside and becoming lost. After several minutes driving around, they see a local farmer. Drawing next to him they roll down the car window.

"Excuse me, could you tell us the best way to get to the nearest town?" The farmer looks at them, nods, pauses and says: "Well if I wanted to go to town I wouldn't start from here."

If you are introducing DevOps across a large enterprise you will probably have a lot of empathy for the farmer's view. Given an inherited technology estate, along with embedded process and practices, it is sometimes difficult to know where to start. You are also likely to inherit at least some vociferous individuals advising why current practices have to remain as they are.

We shall start by stating that it is a given that DevOps adoption revolves around creating a culture that provides support for autonomy and removes centralized command control behaviors. It is not delivering a raft of tools. That said, there is plenty of excellent content that details the concepts and behaviors required. We are not going to attempt to replicate that. Rather, we shall concentrate on providing thoughts and experiences on potential approaches to identifying an automation foundation that enables DevOps in a large, established organization.

Create a standard taxonomy of capability

In large organizations, one of the key challenges is to ensure that a standard taxonomy is used. This assists in optimizing knowledge sharing and understanding the available solutions and experience in the enterprise. The enterprise architect function should own the Enterprise Information Management (EIM) solution and it makes sense for this to detail the standard capabilities you are looking to deliver.

Defining these capabilities is critical. Technologists, as a rule, are better at defining solutions than requirements. By focusing on capabilities rather than solutions (or tool names) you drive a higher quality of debate. It is also easier to identify gaps. In a cross-technology organization, it can highlight surprising areas of strength. For example, the mainframe platform tends to have strong automated deployment capabilities which are often missed when continuous integration and automated testing is discussed.

Define your current technology stack

The next step after defining your capability matrix is to detail the current solution(s) in each capability space. If you are fortunate, you will have an EIM solution that is comprehensive, current and already has all this information. It is more likely that your records will be, at best, partial. In the first instance, focus on placing the current technologies in the correct categories rather than defining which solutions are valid and which are not strategic. If you can achieve a first pass quickly, this can start to drive discussions and decisions. Stakeholder communication is also critical. While not normally a fan of duplication, you need to detail the solutions available to your engineers. Most EIM solutions are great as repositories but are not intended as communication media. We recommend an approach that enables your community to understand the options available to them within the organization. Finally, we should consider the gray market in technology. People across the organization are likely to be using solutions to problems which are not officially recognized or documented in the EIM. There may be solutions already in the organization that will add great value at larger scale. Conversely, potential solutions may have already been tried and issues identified. A culture that encourages communication and sharing of experience and information will enable you to deliver more quickly and avoid replication of effort. Enabling others to feel they can safely share information without repercussions is key. There is obviously a balancing act between encouraging innovation and avoiding an uncontrolled proliferation of technology.

NEXT: Context is King >>

Create a resource page where your capabilities, tools and information are readily available.

This could include:

- Capabilities
- Solutions available for the capabilities
- Precis of each solution
- How to obtain access to the available solutions
- Links to any training material
- Links to vendor pages
- Links to social media content (Stack Overflow, internal blogs and forums, etc.)

3. Enterprise DevOps:
Context is King

Nigel was manning a booth at an internal DevOps conference, and this conversation transpired:

NIGEL:
"So, how can I help you?"

TECHNOLOGIST:
"I need all the DevOps tools."

NIGEL:
"Why do you need all the tools?"

TECHNOLOGIST:
"My manager has put it in our objectives to adopt all the DevOps tools by the end of this quarter."

Nigel knew he had a communications issue; whenever targeting a mass market, his message needed to be simple. He also knew he had an issue about nuance and context.

We now have two issues that need to be addressed. In this section, we'll concentrate on the criticality of contextual knowledge. In a technology company it is very easy, and dangerous, for DevOps adoption to become synonymous with the rollout of automation via a set of tools. Automation is key, but it's certainly not the sole task and a monomaniacal focus on tool adoption introduces many risks.

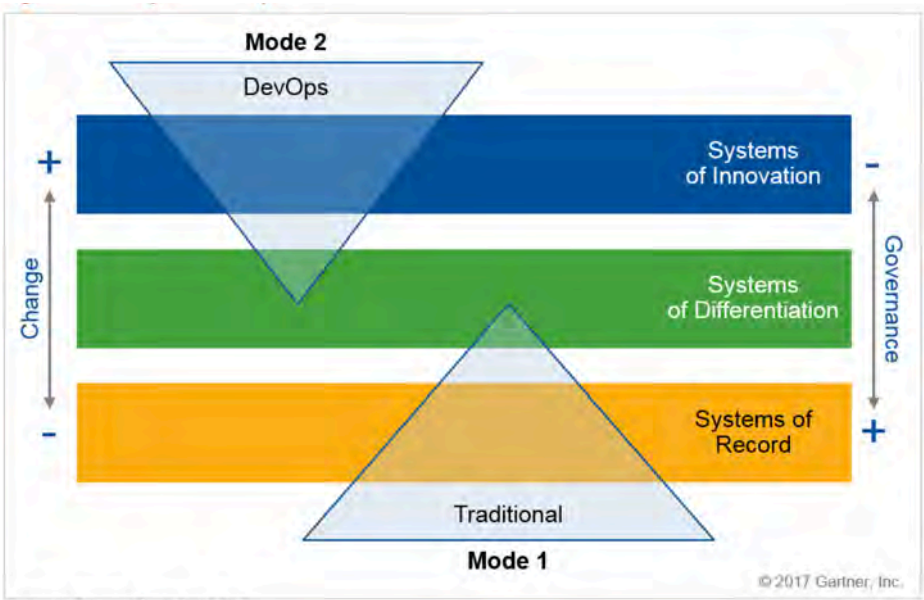
One of the key aims of DevOps is the delivery of consumable content to the business more rapidly. As such, the product backlog will be more dynamic and the need to be in communication with the business is increased. It is critical that the technologist understands their product from a business perspective as that will drive value from your investment.

One of the key concepts that can be used when talking to technicians is the Gartner Pace Layer principle. This is not new or radical, but we believe it is very useful as a means of introducing context into discussions.

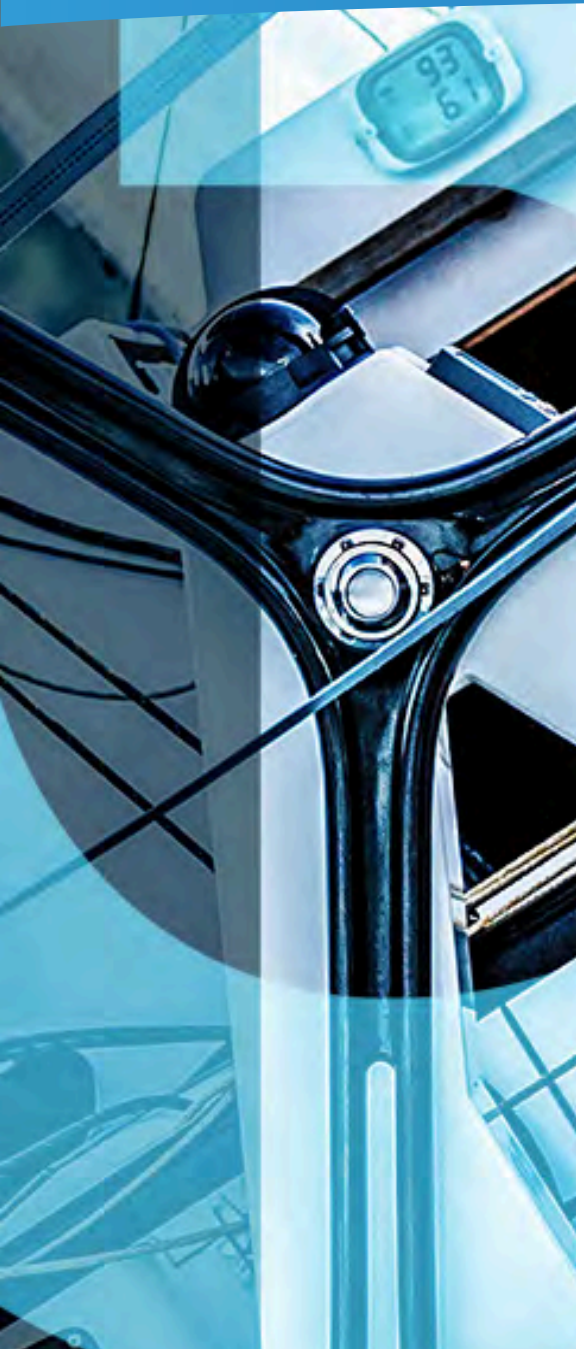
In our experience, technologists often understand the technical landscape they work within (and often are very keen to innovate) but are less confident about the business value of the product they own. At the trade fair, Nigel talked through the three systems the Gartner Pace Layer diagram highlights, after seeing previous versions of the diagram (Figure 1) several years back. He then asked the individual where they saw their product (or aspects of their product) as sitting within the three system types defined. From this, he then moved to a discussion of the areas of automation that would drive the most value for their business.

For example, nobody in the business is going to be delighted if we start innovating around a regulatory reporting product. These are hygiene systems that require accuracy and resilience. There are still capabilities in the DevOps tooling chain such as automated testing, a focus on regression validation, etc. that increase confidence and add value. If you understand the context you are working in, you can make more informed decisions on your priorities.

It is a given that there are interdependencies between different products within the enterprise portfolio. A technologist also needs to understand which other products have a dependence on their product and then use good architectural practices (API's, microservices) to ensure that their product does not become an impediment to the cadence of change of others. In short, within a large organization we see the role of any central function as being an enabler. You provide capabilities and make it as easy as possible for these to be adopted. DevOps and Agile is all about personal empowerment and with that comes accountability. One of the key accountabilities of the technologists is to understand the business context within which they work. It is critical that this is communicated clearly with appropriate guidance to ensure that investment value is maximized.



Source: Gartner (October 2017)



Charting Your Course for Enterprise DevOps? Get Started with CloudBees CI

If you are considering building a foundation for DevOps success, CloudBees® can help. We offer the ultimate continuous delivery solution architected for the enterprise. Teams using CloudBees deliver software 6x faster, on average — as fast as you can think it, you can build it, test it and deploy it. With CloudBees, you'll get:

On-Demand Capacity: Empower teams to focus on delivering value, not maintaining Jenkins®. On-demand provisioning enables teams to start new projects in minutes. With CloudBees, teams go faster...together!

Centralized Management: Simplify your life with streamlined management of your Jenkins servers. Spin up new resources with ease. Upgrade instances once and copy configuration changes everywhere.

Deploy Anywhere: CloudBees CI is made to install and run in a variety of environments. Deploy on Amazon Virtual Private Cloud™, a private cloud with vSphere®, Red Hat OpenShift Container Platform or on your own servers.

Built-In High Availability: CloudBees CI ensures high availability with self-healing Jenkins infrastructure. Should a system interruption occur, resources automatically recover and your continuous delivery pipelines automatically restart, preventing costly downtime.

Security and Compliance: CloudBees ensures everyone is on the same version, eliminating conflicts. You define access via roles and responsibility. Compliance just got a whole lot easier!

Unify Your Toolchain: Frustrated by each team using their own tools? Unify your toolchain with CloudBees and reap the rewards of standardization. Provide a single, unified Jenkins experience to all teams.

Learn more about CloudBees CI: www.cloudbees.com/get-started

About

CloudBees is the industry's leading DevOps technology platform delivering the world's first end-to-end continuous software delivery management system. CloudBees enables developers to focus on what they do best: Build stuff that matters—while providing peace of mind to management with powerful risk mitigation, compliance and governance tools.

Used by 50% of the Fortune 500, CloudBees is helping thousands of companies harness the power of continuous everything and gets them on the fastest path from great idea, to great software, to amazing customer experiences, to being a business that changes lives.

Visit CloudBees at www.cloudbees.com.

CloudBees CI is built on top of Jenkins, an independent community project. Read more about Jenkins at: www.cloudbees.com/jenkins/about

0820v02

CloudBees, Inc.
4 North Second Street | Suite 1270
San Jose, CA 95113
United States
www.cloudbees.com
info@cloudbees.com