



TRANSFORMING THE NATURE OF OPERATIONS

DevOps is not only a tooling framework, but also a methodology that can take incumbent firms to new levels of agility and dynamism. For this to happen, cross-functional DevOps teams need to be introduced that break down silos between development, testing and operations. This article – the fourth in our case study series on one global bank's DevOps transformation – explores how operations were transformed within these agile units, with processes simplified and sometimes automated to realize the full benefits of DevOps.

With operations cognizant of the applications they support, and developers aware of operational issues that arise, a self-service culture persists that enables teams to deliver products faster, better and cheaper.



Introduction

Netflix uses DevOps to deploy code thousands of times a day, while Nordstrom software releases increased from twice yearly to weekly using DevOps.¹ Compare these companies to a large global bank that struggles to deploy code every month, and it's no wonder that DevOps is touted as a savior for many financial service players that have to keep fintech upstarts at bay.

DevOps is not only a tooling framework, but also a methodology that can make a large firm nimbler and more dynamic.

DevOps also fundamentally impacts the bottom line — it can enable firms to achieve 50% higher market capitalization growth over a three year period.²

The first three articles described how small, agile, cross-functional DevOps teams were introduced at a leading global bank, breaking down silos between development, testing and operations. They also described how DevOps tooling framework was used, introducing continuous and automated integration, testing and delivery of software with DevOps as a practice.

This article explores how operations were transformed within these agile units, with operations processes simplified and sometimes automated to realize the full benefits of DevOps.

Transform IT operations for the DevOps world

The bank's operations used to impose a particular working style. There were many handoffs between teams, set up in a waterfall model of engagement. Operations engineers knew little about the applications they supported, and developers lacked awareness of operational issues.

This style of work resulted in frequent delays. One quick fix went to the bottom of a pile of quick fixes, and tasks were sent back to previous groups for clarification.

To overcome these problems, application developers learned to manage their own operations, with guidance from Infosys DevOps experts. Through self-service, the bank's developers were empowered to write code that reconfigured the system or updated servers based on the unique requirements of their applications.³

On the operations front, success required picking well-rounded operations engineers who had a team-player mentality. In the new agile operating model, these engineers make it easy for developers to build highly automated self-service solutions. The developed code is easier

to manage and support in the follow-up processes considered next.

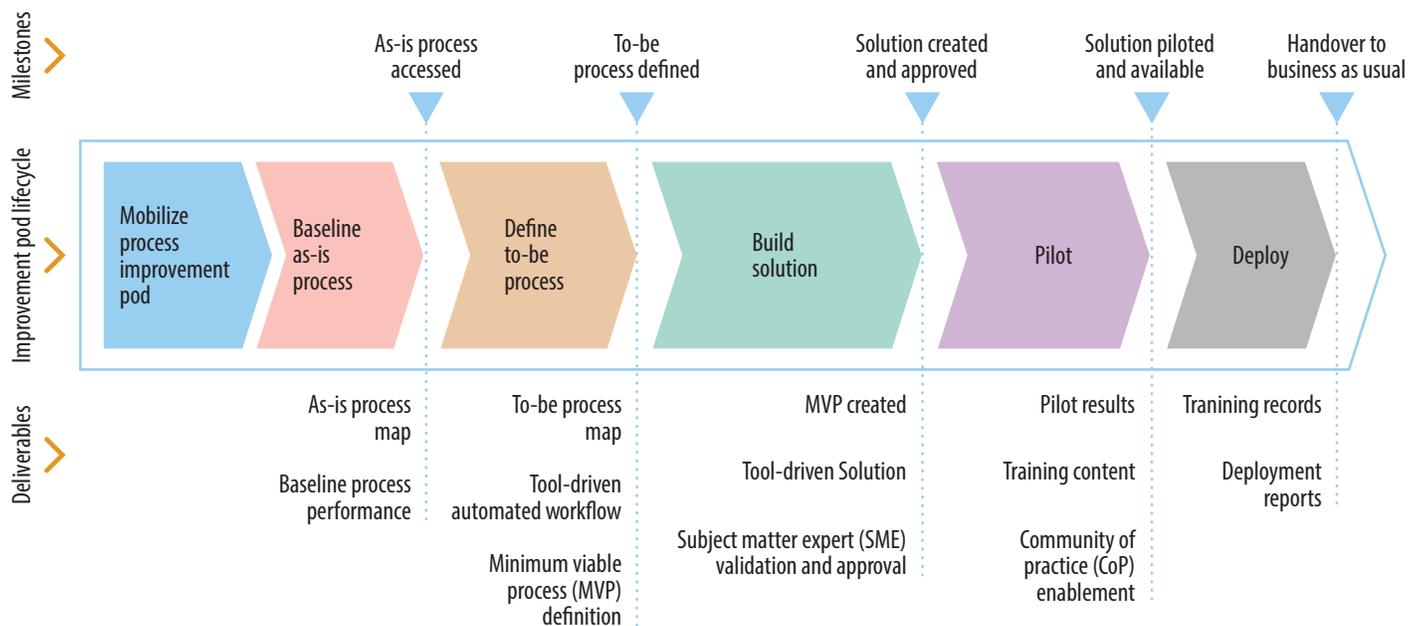
Improving the process of operations

An automated solution or a minimum viable process (MVP) was implemented for every process in the workflow.

Process improvement agile teams were formed to evaluate every process in the IT operations area. They designed either an automated, tool-based solution ("zero effort") or the MVP that balanced risk and quality.

The improvement life cycle is shown in Figure 1.

Figure 1. The minimum viable process is a tool-driven solution approved by a subject matter expert



The Ops Processes

Improvements to operations in DevOps show how agile operations processes really are and how they can integrate automated solutions.

Service assurance, infrastructure provisioning, release management, change management, access and problem management, event and incident management, and knowledge management are all processes that

were either automated or made more effective.

Each of these are discussed in more detail below.

Service assurance

Before its transformation, the global bank had over 30 quality gates in its service assurance process, making agility difficult. Software gate reviews lasted up to six weeks, and there were numerous handoffs between team

members. Effort was duplicated due to overlapping quality gates, and ownership was a problem.

In the MVP, duplicate gates were removed. Automation was introduced to trigger gating reviews, and pathway-based gating was used for simple changes. Some gates were fully automated, so that experts didn't need to sign off on code. This meant that reviews could be scheduled during nonworking hours.

Infrastructure provisioning

The provisioning process can last six to nine months in large, monolithic enterprises. There is a sizable mismatch in demand versus supply. Additionally, provisioning follows a service level agreement (SLA) mindset.

The new process accelerates provisioning through virtual infrastructure and the cloud. Automated provisioning and asset management were enabled using strategies like infrastructure-as-code. This eliminated redundant approvals and validated provisioning with reduced turnaround times.

Release management

Batch releases lead to long wait times. Human error and security violations are hard to avoid because of the manual nature of such releases. Additionally, handoffs require significant coordination of different individuals within the project.

In the MVP, one-click deployment of software into the production environment was achieved using tools like Ansible.

Change management

Changes typically take 15-20 days before deployment. A priority lane change management process was introduced, and this removed a significant number of process steps. Teams could use this process only if they were mature enough, however, with maturity governed by how well teams followed DevOps as a practice principles. Principles included no manual deployments and adherence to minimum operating standards (MOS).

Application programming interfaces (APIs) were also developed to interface application life cycle management and service management tools, to allow for exchange of data between build and run teams.

Access management

Access requests for systems and applications can take up to 21 days, with a complex process workflow and a long turnaround time for access to the production environment.

The new MVP included role-based access to production data. A new platform was created for role-based entitlements, enabling “one identity” verification. Also, read-only access to the production environment was rolled out to speed problem investigation and diagnosis.

Problem management

Problem investigation used to be conducted ad hoc, and application teams became involved late in the process. Lax problem investigation SLAs resulted in long lead times for problem diagnosis.

The MVP assigned problem diagnosis tasks in the sprint backlog. Problem records were also automated, and automated alerting was introduced so that team members could monitor root causes if they were not found in 72 hours.

Event and incident management

False alerts make uncovering important events and triggers difficult. Handoffs cause delays, and significant human effort is needed to diagnose repetitive incidents and known errors.

In the new MVP, agile teams were empowered to manage low-priority incidents with automated pass-through of incidents to the team, significantly reducing resolution time. Tools were introduced to remove false alerts, and agile teams were enabled to develop self-service scripts for active service monitoring, incident resolution and proactive maintenance.

Knowledge management

Information obsolescence was a critical issue in the global bank due to lack of proper version control, and a process-heavy workflow governed information updates.

Version controls were put in place through page headers.

A new knowledge management system was deployed as part of the MVP, allowing teams to collaborate and jointly create, manage, edit knowledge, and then remove once no longer relevant to the process.

The power of automation to take DevOps to the next level

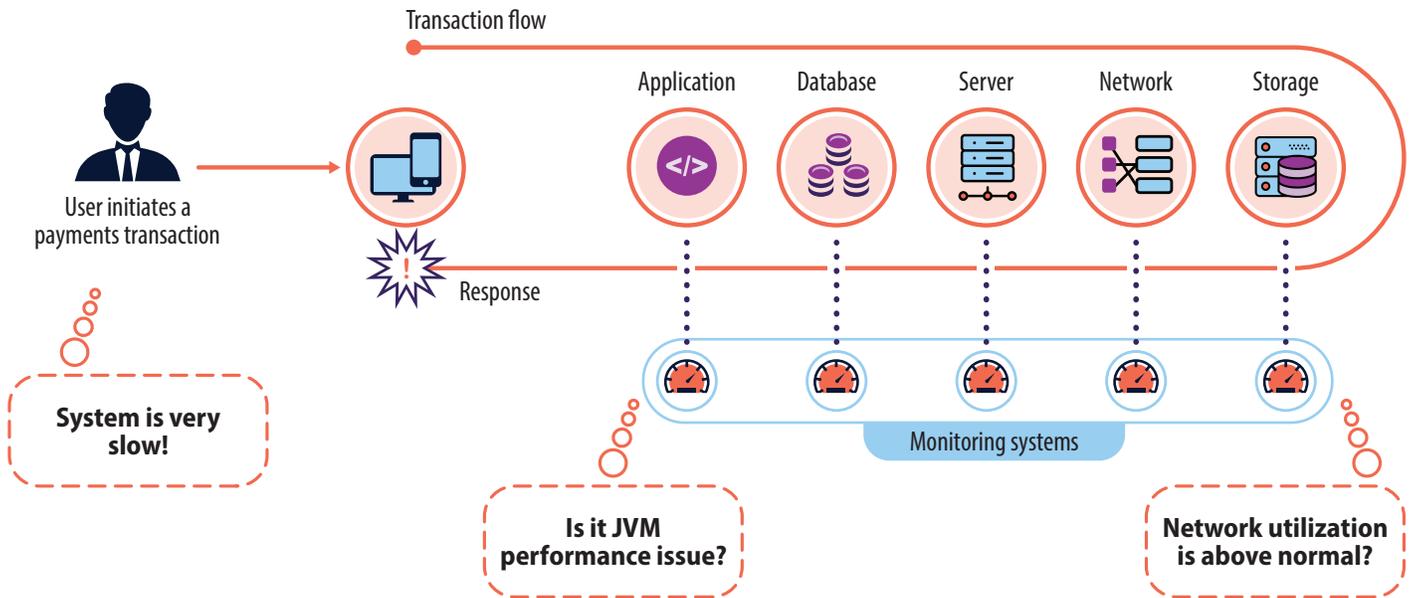
This article series has so far covered better operations processes, an agile team structure, and tools that enable continuous software delivery, validation and deployment of software. But the power of automation to improve monitoring and provide problem diagnostics is another part of the puzzle that CIOs and DevOps problem solvers need to know about.

AI and machine learning (ML) capabilities can also be deployed to automatically recover from incidents, predict service availability, and take automated, proactive actions to avoid service disruptions.

End-to-end application performance monitoring

In the old way of working, many monitoring solutions were used across server, application and network layers. This traditional setup created silos, and these hid the true cause of performance issues.

Figure 2. Traditional monitoring hides the true cause of issues in server, application and network layers



On the other hand, end-to-end performance monitoring supervises business services from the end-user perspective. It features built-in automation to quickly isolate and diagnose application performance issues. Additionally, it monitors events at the transaction and process levels.

Tools were implemented to leverage these capabilities: App Dynamics for distributed applications, and CA APM and CA SYSVIEW for mainframe applications.

Advanced problem diagnostics using Splunk

An issue can take up to 20 days to resolve, and delays are created due to insufficient information to run diagnostics.

Advanced problem diagnostics collate data from every component in the application landscape. This information is harnessed quickly to identify the source of the issue.

The bank used Splunk in the DevOps life cycle; this tool aggregates logs from every component (e.g., application, database, network router) in the service life cycle to look for issues in real time.

Additionally, the event alerting system creates proactive reporting and informs operations experts of potential failures. Even if events are not alerted, logs can be analyzed later to determine the root cause of problems.

AI and ML

Many AI and ML tools are available to optimize operations processes. Three use cases were of particular interest for the new process solution:

	<h3>Incident prediction</h3>	<p>The commercial off-the-shelf (COTS) product had an intelligence engine that predicts when an incident or event is likely to occur, at which application component, and in which infrastructure element. The prediction is built on a large amount of prior knowledge of past incidents, so it is highly accurate. At the same time, the engine learns from actual events compared to the forecast and optimizes itself automatically after each incident.</p>
	<h3>Automated self-heal</h3>	<p>The chosen COTS product automatically triggers workflow processes that take corrective action when events occur, restoring the services to normal state. Specialists develop scripts based on incident volume and trends, which are then deployed automatically at a targeted location.</p>
	<h3>Preventive maintenance</h3>	<p>The product runs scheduled processes that perform housekeeping activities like clearing hard drives or cache in the server and reviewing backup logs. The scripts ensure that the health of the application landscape is adequate and help in incident avoidance.</p>

But DevOps doesn't happen overnight

The full technical solution has now been described. However, there is still some distance between implementing a technical solution and actually delivering cultural changes that impact the bottom line in a significant way. According to the Puppet Labs State of DevOps report,

mature DevOps teams complete code deployments up to 8,000 times faster than lower-performing competitors.

*Mature teams can also complete deployments in less than a day, with a 50% reduction in failure rates after mature DevOps teams have been implemented.*⁴

In order to capitalize on DevOps and nurture an agile organization, the bank's DevOps mindset had to become a routine part of the software

life cycle. Behavioral experts were brought in to assist the transformation further, giving individuals, teams and leadership more power to do the right things, at the right time.

This journey of behavioral change is the subject of the next article in this series (Article 5).

References

- ¹ <https://techbeacon.com/devops/10-companies-killing-it-devops>
- ² <https://puppet.com/resources/whitepaper/state-of-devops-report>
- ³ <https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/transforming-it-infrastructure-organizations-using-agile>
- ⁴ <https://puppet.com/resources/whitepaper/state-of-devops-report>

Authors

Alok Uniyal

Vice President and Head-IT Process Consulting
alok_uniyal@infosys.com

Harry Keir Hughes

Senior Consultant-Infosys Knowledge Institute
harrykeir.hughes@infosys.com

About Infosys Knowledge Institute

The Infosys Knowledge Institute helps industry leaders develop a deeper understanding of business and technology trends through compelling thought leadership. Our researchers and subject matter experts provide a fact base that aids decision making on critical business and technology issues.

To view our research, visit Infosys Knowledge Institute at infosys.com/IKI

For more information, contact askus@infosys.com



© 2019 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.