



## TRANSFORMING ENTERTAINMENT THROUGH DEVOPS

Pressure to rapidly release new products revealed cracks in one of the largest U.S. broadcasters' software development processes. The company ultimately took the bold step to adopt Agile and DevOps. This required a fundamental shift in processes, technology and culture. But the benefits were worth it.

# Transforming entertainment through DevOps

One of the world's largest entertainment companies wanted to offer enhanced viewing experiences to its customers. But it was taking months to translate new business ideas into live applications that could support this vision. And even then, the end product was mired in quality issues.

Yet the business turned this around within two years. The company radically redesigned its operating model. It adopted Agile and DevOps practices that reduced environment provisioning and software deployment cycles from three months to 26 seconds, saved 85% of testing efforts through automation and increased sales by 55%.

## What are Agile and DevOps?

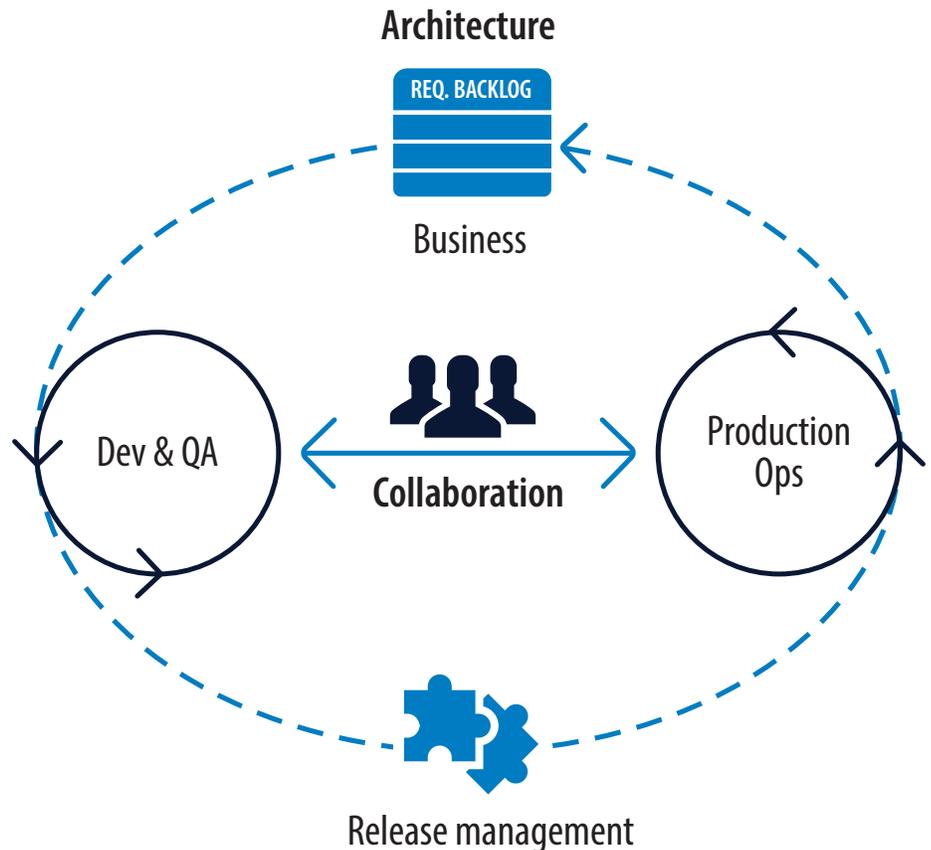
In essence, Agile and DevOps are both approaches that reduce time and increase quality by collapsing down the traditional chain of events in a software development project, and encouraging collaboration and rapid decision-making between the different people involved.

More specifically, Agile is an iterative software development approach wherein large functionality is broken down into smaller parts and developed by teams comprising experts from different areas (business, development, testing, etc.). The teams are equipped to swiftly respond to changes at any point in the development cycle.

DevOps is a set of practices that integrate the development and operation activities of software development processes to enable faster development and deployment of software, with higher efficiency,

less risk and better quality. Cross-functional collaboration, which is the bedrock of DevOps, nurtures innovation in developing business solutions (see Figure 1).

Figure 1. DevOps model — breaking silos to form a network



Source: Infosys

## DevOps vs. traditional waterfall model

The waterfall model, a term used to describe the traditional approach to developing software, involves a sequential set of processes to design, develop, test and launch a product. Each team finishes its part of the project and passes it on to the next team to continue. There is little communication between the teams. This can become tricky in large projects since there is no scope to incorporate rapidly changing requirements in such environments. Issues found in the later stages also lead to dangerously long delays and high rework costs.

DevOps, on the other hand, insists on cross-functional teams and therefore more interaction between developers, testers and designers. It involves shorter cycles of development, testing and deployment. It is driven by the "fail fast, early and often" mantra. With a continuous feedback mechanism, modifications at any stage can be easily fixed and hence present fewer risks.

## Transforming the viewer experience through Agile and DevOps

The subject of this case study is one of the world's largest entertainment companies, distributing satellite television services in the Americas. At the start of this decade, over-the-top services began to gain popularity. With OTT services, media content could be directly distributed to viewers via the internet, bypassing the broadcast platforms. More people shifted from television sets to devices such as mobile phones, tablets and laptops to access media content, and advertising sales began to move to online platforms.

The likes of Netflix and Amazon were already leading in this domain, and the sales of the company, which still

used traditional ways of content distribution, were affected. This company wanted to get into the OTT streaming service business as soon as possible. It intended to deliver rich and unified experiences across multiple devices and browsers. It also hoped to improve its website — a critical channel for online sales, OTT content and self-service capabilities.

However, there were numerous barriers to achieving this. The time to turn an idea into reality was too long. Very few people involved had a complete understanding of the projects they were supporting. There was no collaboration between teams and no team would take initiatives to fix the issues when the project encountered roadblocks. Thus, the costs of projects spiraled and the quality of the resulting products were poor.

It seemed impossible to address these issues using the existing monolithic architecture and processes. This is what drove the company to consider the radical transformation of adopting Agile and DevOps — a completely different approach to what had been tried previously. The objective was to break down silos and empower teams to work together in order to deliver high-quality applications faster, efficiently and in cost-effective ways. The ultimate outcome of this was to:

- Develop next-generation TV viewing experiences.
- Implement new business ideas and offers quicker.
- Minimize testing and deployment efforts to support more projects.
- Reduce turnaround time through close collaboration between different teams.
- Mitigate risk by failing fast and failing early.

### The three pillars of transformation

Infosys was engaged to assist in the transformation journey. Given the scale of the transformation, a three-pronged approach to disrupt the existing process, technology and culture was designed to achieve the goals.

#### Process transformation:

Evaluation of existing model through the DevOps maturity model, identifying areas of improvement through value-stream mapping and implementing DevOps practices across the entire IT portfolio.

#### Technology transformation:

Architectural transformation, moving from a monolithic waterfall approach to a lightweight architecture made up of separately creatable and deployable parts.

#### Cultural transformation:

Reconfiguration of the operating model to increase collaboration across business, development, operations and quality teams; automation of the tracking system; promotion of a DevOps culture; and a minimization of efforts where possible.

With people, technology and process innovation as the guiding principles, Infosys helped the company define a new operating model and develop a large-scale DevOps and Agile adoption program across the entire portfolio.

## Process transformation

The first task was to evaluate existing practices in order to identify existing shortcomings and challenges.

A team then had to develop a DevOps model that would meet the set goals.

The company, together with Infosys, first created a DevOps maturity model and evaluated the company using criteria associated with it. The DevOps maturity model is a matrix with five criteria: culture; design; build and deployment; reporting; and testing. It also has five maturity levels: crawl, walk, run, sprint and fly.

The company also conducted a value-stream mapping exercise linked to the DevOps maturity assessment. The value-stream mapping exercise assessed the current state of processes used to develop a product, identified the improvement areas and suggested areas where processes could be changed without reducing value.

Cross-functional scrum teams were created, dissolving boundaries between the development, business, operations and quality teams. Each team was assigned the complete ownership of an application — from development to deployment.

A “shift left” testing approach was introduced. Shift left is a practice in which the testing team gets

involved early in the development cycle to identify issues and enable quick rollback in case any problems are identified.

At first this process went smoothly, but quickly challenges started to arise. With implementation across multiple applications, each team developed its own way of doing things. The tools, processes and practices were not common across applications. The teams were drifting away from the important principles of Agile and DevOps: collaboration, efficiency and reusability. This started to increase costs.

To tackle this, a central governance team was constituted immediately. This team was composed of experts from the entertainment company and Infosys. It brought together the synergies between different application teams and increased reusability. Having all the teams aware of what the others were doing meant that if one team had faced a challenge previously, the other teams could benefit from how that team had solved it instead of starting from scratch.

## Technology transformation

Adoption of DevOps and Agile meant shifting from traditional technologies to a new technology stack relevant for Agile and DevOps. Automation is the foundation of DevOps. Infosys enabled automation by deploying continuous integration, continuous deployment, pipeline as code delivery and dual environment deployment practices.

## Continuous integration

This is a practice in which code developed by many developers is integrated with a shared repository several times a day. The codes thus integrated are automatically built and tested. Any changes or issues are relayed back to the developers immediately, who then work on the next version. This is a “fail-fast-and-fail-often” model. It ensures problem detection at early stages to avoid major setbacks later.<sup>1</sup>

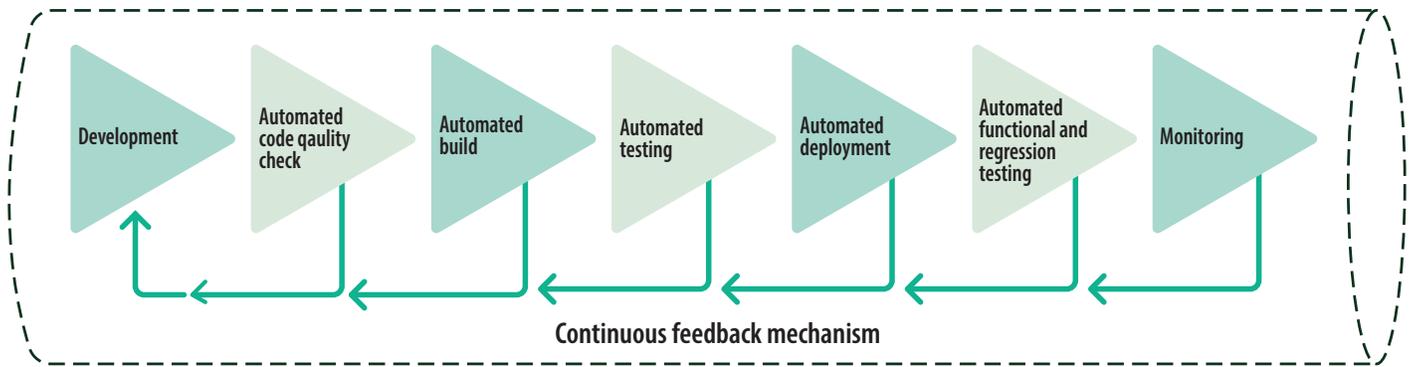
## Continuous deployment

This requires that each change in the system is easily deployable through automation tools. The code, which is continuously delivered, is immediately tested by the end users. Their suggestions flow back to the developers, who can then incorporate the changes in next version. This enables continuous improvement of the product instead of having to wait until the end of the project to realize what went awry, at which point it might be too expensive to correct.

## Pipeline as code delivery

The practice of using CI and CD to develop and deploy code resembles a pipeline, as shown in Figure 2. It has a continuous feedback mechanism for continuous improvement. This pipeline gets triggered when any code is developed or changed. The code quality is checked, unit tested and integrated with the repository. That version is then tested and auto-deployed. This is the basis of automated one-click deployment.

Figure 2. Pipeline as code delivery



Source: Infosys

### Dual environment deployment practices

This is a strategy for releasing software code that eliminates downtime. In this approach, two identical hardware environments are set up, say blue and green environments. One of the environments is active at any stage while the other is idle. New code is released in the idle environment and thoroughly tested. Once the results are satisfactory, the user traffic is routed to this environment. If any issues are encountered, traffic is immediately redirected to the idle environment that was working correctly. This ensures there is no downtime for the application, either during code deployment or in the face of any other issues.

### Tools and infrastructure

In this enterprise, CI/CD practices were used to implement an integrated pipeline to achieve one-click deployment. Many tools were used for automation and for streamlining the processes. Jenkins, an open-source automation engine, acted as an orchestrator for the automation of the build, test and deployment processes. Selenium was used for testing automation. Automated tests could be run on multiple browsers and machines in parallel. Docker was used for automating deployment of

the code. Infrastructure as a code was implemented. IaC helped DevOps teams automate the management of multiple versions of code and different environments (development, test, etc.) that the DevOps model entailed.

A dashboard was created to monitor the entire pipeline, providing a detailed view of progress at each step. A vendor-agnostic, on-demand cloud infrastructure was used to reduce dependency on a single cloud service provider. The company could use and pay only for the cloud resources it consumed. This ensured higher utilization, flexibility and cost-efficiency.

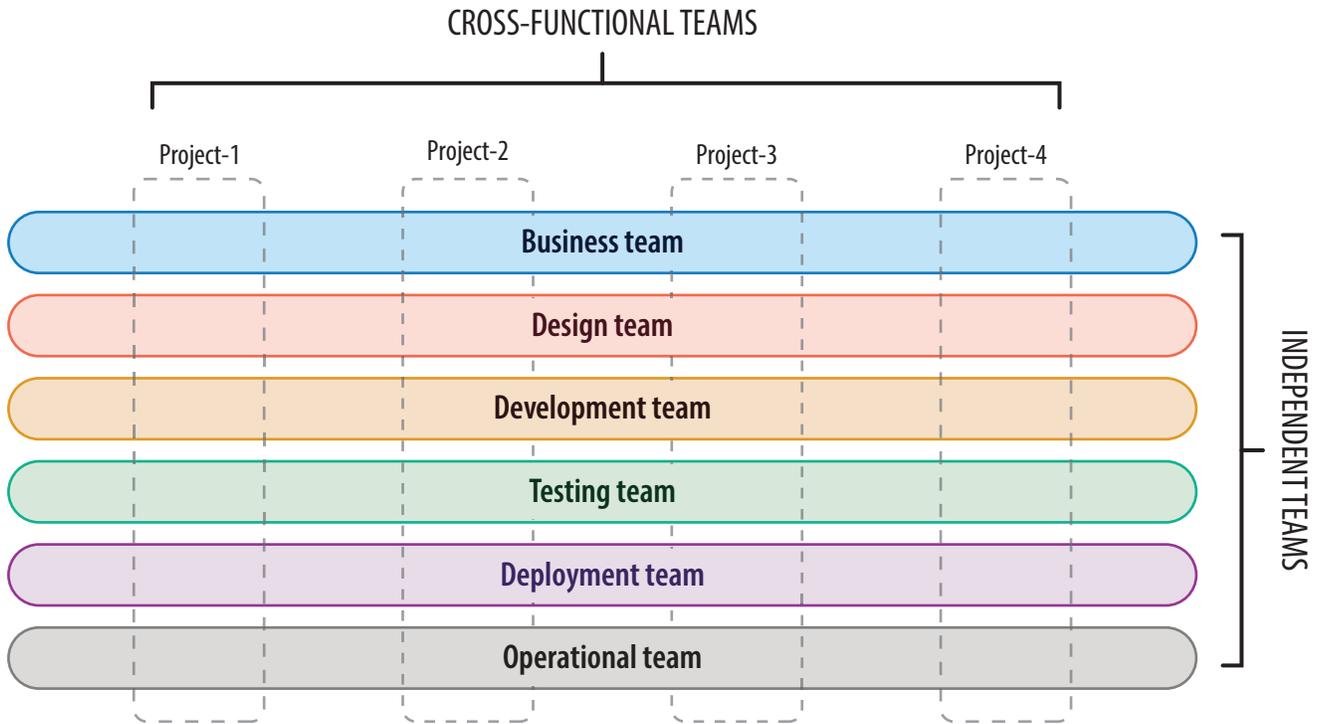
### Cultural transformation

In the traditional model, the teams were unaware of the work of other teams. Agile and DevOps disrupted this model of working in silos. Cross-functional teams were created, with experts from different areas such as business and development testing for each application (see Figure 3). They were in charge of the entire process up to delivery, which brought a sense of ownership to the teams. The end-to-end expertise in a single team meant a seamless development cycle. Any concerns, issues and roadblocks could be immediately resolved. The whole team was now aware of the status of the application.

Prior to Agile and DevOps adoption, each team had a different way of working. To bridge the gap between business and IT, the role of strategic product analyst was created. Strategic product analysts were responsible for the overarching product and ensuring that communications happened across different relevant teams. Agile coaches from Infosys trained the teams to help them adapt to the new way of working. Workshops were conducted to promote the DevOps culture.

An entirely new set of skills was required to make all of this work. For example, the teams now had full-stack developers and DevTesters, as well as SQA testers who focus on risk-based testing after supporting automation of repetitive tests. Many people in the company were reskilled for these roles, and in some cases experts were recruited. Shared DevOps teams were created, and experts in DevOps and Agile were hired on a temporary contract basis to “handhold” the teams until they gained expertise.

Figure 3. Cross-functional teams



Source: Infosys

However, implementing this change was not easy.

“Changing mindsets to adopt the DevOps and Agile way of working was the hardest of all challenges. Technical and process challenges were easier in comparison,” says the project manager who handled this project. There was initial resistance to the change and disruption that DevOps brought in. “Fail fast and often” was not an easy idea to inculcate. People were threatened by the changing roles and skill requirements.

To alleviate the concerns, Infosys conducted multiple workshops, demonstrations and value articulation sessions for development and operations teams. This ignited curiosity and made adoption easier. Cross-functional demos and team-building activities helped in fostering trust. Overall, “show and tell” succeeded in assuaging fear and changing mindsets.

With practice, people started appreciating the benefits of the Agile and DevOps culture. The transparency,

trust and sense of ownership that came with this culture guaranteed that they would not fall back into the older way of working.

## Impact of transformation

Though the path of transformation was fraught with many challenges, the project was a success. With the dual environment deployment strategy, there was no downtime in the live environment even if there were issues in the back end. Services worked seamlessly even during the times of highest number of visitors to the website. This meant more revenues (downtimes previously had cost hundreds of thousands to the company).

### Faster — improved cycle time

Deployment times decreased significantly. Environment provisioning and deployment (the process of implementing the code — testing,

configuration, installation), which took around three months with traditional methods, shrank to a staggering 26 seconds. Automating regression test execution saved 85% of testing effort.

### Stronger — better quality

Fail fast, early and often helped detect the defects early in the development. This prevented defects from percolating to the later stages. The reduction in rework efforts and quality issues resulted in annual savings of \$500,000.

### More — increased revenue

The number of product releases increased multifold after DevOps implementation. With more features and products came more sales. Peak day sales revenue increased by 55%, with the Agile and DevOps approach being an important contributing factor. There were no leakages or failures, which generally happen when the websites or services are not accessible to users. The number

of releases per year doubled every passing year, with more than 300 product features delivered in 2018.

The necessity for faster and more efficient creation of high-quality software led to a revolution in the way software is developed and delivered at this entertainment giant. It is not alone. DevOps adoption has surged in the past few years, with 56% of global organizations now implementing or expanding DevOps initiatives, according to a recent Forrester survey.<sup>2</sup>

While DevOps and Agile practices come with a lot of promises, they also demand a lot of investment and time.

Charting out the Agile and DevOps implementation plan, setting up the infrastructure and technology, forming teams with the right skill set, and changing the culture can take a couple of years and millions of dollars. So support and confidence of leadership and buy-in from teams are extremely crucial.

Companies might be tempted to pick only convenient practices from Agile and DevOps, and skip the difficult parts. But Agile and DevOps work only if they are adopted holistically. Piecemeal adoption will ultimately result in more processes without any benefits. Also, teams should be

co-located as much as possible and very well connected. If not structured correctly, miscommunications and backlogs can form, leading to dire consequences.

It is always easy to show success in small pilot projects. However, scaling Agile and DevOps in a large organization can be complicated. Changing technology, processes and ways of work in large-scale projects can take time and can be complex. Any failures or roadblocks can have very large and negative impact on the projects.

### It is important for organizations to understand the following before embarking on this journey:

- Appraise the current situation and the challenges.
- Clearly describe what the goal is that you hope to achieve.
- Assess how DevOps fits into this picture.
- Identify the places where DevOps and Agile can add value.
- Make a detailed plan of the DevOps adoption program.
- Continuously monitor progress so that problems can be addressed as early in the program as possible.
- Identify the right tools and frameworks to ensure that automation can work smoothly.
- Use Agile coaches and DevOps experts to train teams and project managers.
- People are the most important pillars of any project. For successful Agile and DevOps adoption, cultural change involving everyone from top leadership to each team member is extremely important. Equip and support them well so that they can adapt to the new way of working.

## References

- 1 "What is a Continuous Integration and Delivery Pipeline, and Why Is It Important?," Codefresh, Dec 20, 2016, <https://codefresh.io/continuous-integration/continuous-integration-delivery-pipeline-important/>
- 2 "Five Key Trends To Benchmark DevOps Progress," Forrester, January 31, 2019, <https://www.forrester.com/report/Five+Key+Trends+To+Benchmark+DevOps+Progress/-/E-RES120481#figure1>

## Authors

### Alok Uniyal

Vice President and Head of IT Process Consulting  
Alok\_Uniyal@infosys.com

### Rachana Hasyagar

Consultant – Infosys Knowledge Institute  
Rachana.Hasyagar@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](https://infosys.com/IKI)

---

For more information, contact [askus@infosys.com](mailto: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.