EPLEXITY | aws partner network

Navigating the next evolution of application development

EPLEXITY

Contents

01 Introduction

02 How has software been built? Waterfall vs. DevOps

03 The next evolution: Microservices

05 What will you need to do differently to capitalize on modern application development?

08 Learn more

EPLEXITY

Introduction

Building and maintaining a competitive edge often requires you to evolve with your customers' needs and deliver innovative experiences. Meeting this demand entails listening to your customers, experimenting with new ways to deliver value, and iterating accordingly. Now that software is at the core of most businesses—across every size, geography, and industry vertical—staying ahead of this cycle means evolving development practices to become more nimble and agile.

This eBook will provide insight into various development models and prescribe recommendations to help you adopt a culture for delivering new customer value and proactively building a competitive edge.

EPLEXITY

How has software been built?

Waterfall vs. DevOps

As enterprises look to evolve their development practices, many are moving away from a Waterfall model—a linear approach to software development, that tends to be less iterative and more structured. Because each phase of the Waterfall model is dependent on each other, changes "downstream" can have massive impact on previous tasks. In particular, testing occurs only near the end of a project, meaning modifications can cause a project to entire restart entirely. In response, many have historically undergone long, meticulous planning cycles to minimize modifications. Using this model can massively limit agility and incur excessive costs.

In the past couple of decades, many enterprises have embraced (initially) Agile and (more recently) DevOps cultures for software development. The former calls for regular testing and greater collaboration amongst disparate teams to accelerate development. But as this practice started building success, it became clear that the new bottleneck was operations as development teams were slowed by infrastructure provisioning and management. Thus, DevOps was born.

DevOps takes the ideas of the Agile model a step further, folding in operations teams to further accelerate the development process. With technology to automate infrastructure provisioning and management, many now benefit from even faster development cycles.

These technology advances represent only part of aim to create higher quality applications at a greater speed. The next evolution in this endeavor used to bolster DevOps practices is microservices

EPLEXITY

The next evolution: Microservices

Microservices is an architectural approach to software development used in many DevOps environments that is composed of small, independent services that communicate to well-defined APIs.

In traditional development models, everyone is working on the same codebase. Conversely, with a microservices architecture, everyone works on their own codebase, independent of other teams. This enables smaller, frequently released application updates, as opposed to batched changes occurring at regular intervals. As a result, enterprises can accelerate development cycles, foster innovation, and improve software scalability. Additionally, by divvying up software into smaller, well-defined modules, organizations can improve maintainability of code and overall software quality.

Organizations can support their DevOps practices and microservices architectures with additional technologies to drive greater efficiency. Namely, containers and serverless technologies.

Containers are a software packaging method that aim to reduce the operational effort of deploying applications by consolidating an application's code, configurations, and dependencies into a single object. Whereas traditional virtual machines require their own operating system, containers run off of a single OS kernel, creating greater resource efficiency. They can be spun up in an instant, accelerating workload processes and application delivery.

Serverless compute technologies can be used to simplify IT management and dramatically reduce costs. This is done by abstracting the undifferentiated heavy lifting of software development—such as managing software to handle application reliability and scaling—away from the developer. Instead, serverless compute technologies run your code in response to events and handle all of the corresponding backend administration. This means you no longer have to provision, deploy, update, monitor, or otherwise manage servers and your applications scale automatically. Furthermore, you use (and pay for) only the compute resources you need, eliminating all idle capacity.

EPLEXITY

What will you need to do differently to capitalize on modern application development?

Tapping into the increased agility, cost-effectiveness, and productivity that modern development practices enable requires more than just updating your technology toolset. Rather, it requires that stakeholders across the organization think about their roles differently—that people are willing to take on some new responsibility in the name of delivering new customer value. Some of the key changes that many organizations must make as part of this transition include:

Treating development as a more iterative process

Traditionally, software development has leveraged long, time-consuming pipelines where developers spend months and months working on software before it is tested. This approached rested on the hope that developers could address every possible issue before pushing their release down the pipeline. In recent years, organizations have found that testing code and iterating based on the findings of such tests more frequently is a more effective approach. Cloud-based infrastructure resources can be provisioned on-demand, which makes performing individual tests faster and this approach more feasible than doing so on-premises.

Collaborating to complete tasks earlier ("Shifting responsibility left")

As developers are asked to iterate more, they also need to be aware of things that are often considered to be someone else's responsibility using Waterfall methodologies. For modern, iterative development processes to work effectively, deployment methods and other such considerations need to be top of mind for everyone in the development pipeline—the developers themselves included.

Including security closer to the front of projects

As part of this "shift left" mentality, security is no longer the job of one team near the end of the development pipeline. From the moment that development begins until the software is released into production, security needs to be top of mind. This reduces the chance that vulnerabilities in software are not noticed until it is close to release, in turn eliminating re-work that slows down development cycles and time to market.

These changes are subtle, but they will require buy in from stakeholders across the organization. To streamline this new way of operating, it is important to align with additional best practices (detailed on the following pages) when modernizing your application development processes.

As you go about modernizing your application development practices it's important to embrace tools and methodologies that have made others successful. From a technology standpoint, here are three best practices that have benefited modernization initiatives.

Continuous integration and continuous delivery

Update applications and infrastructure quickly by automating continuous integration and continuous delivery (CI/CD). Automated CI/CD practices help you release better features faster and eliminate manual code deployments. When your infrastructure includes hundreds or thousands of components, automating with CI/CD allows you to scale and react faster, while minimizing the risk of human error.

Security

Building authentication, authorization, and compliance auditing directly into every component of your application, while also securing your infrastructure, helps improve your security posture and development efficiency. Security teams and developers can work in tandem to code more securely, test their application for vulnerabilities and scan their code for malicious content. For example, using an image scanning solution, you can detect vulnerabilities of container images or image dependencies.

Monitoring and logging

ABeing more thoughtful about logging and monitoring is one of the biggest changes organizations typically must make as part of this transition. AWS and solutions from our Partners make it easy to store and analyze log files from each service, as well as collect metrics, set alarms, and view requests from end-to-end as they travel through application components. With increased observability, you can rapidly detect and respond to issues and improve application performance.

By adopting these best practices to deliver modern applications, you can build applications that are more secure, reliable, scalable, and quickly available for your customers.

EPLEXITY

Learn more

Evolving your development practices is key to driving greater agility and realizing your business's full value. Learn how EPLEXITY and AWS can help provide the tools you need to get started.

To learn more, visit eplexity.com

EPLEXITY

eplexity.com

888-501-5979