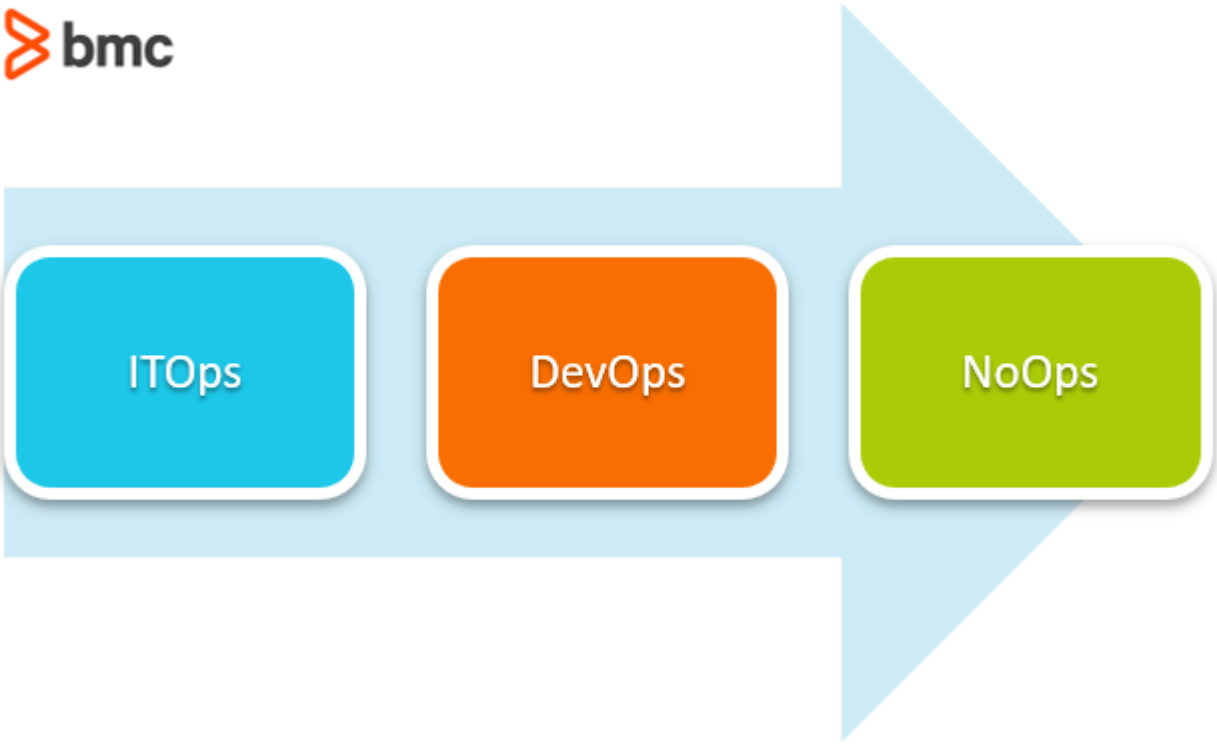
ITOPS VS DEVOPS VS NOOPS: THE IT OPERATIONS EVOLUTION

The modern technology landscape is an uber-competitive, constantly evolving ecosystem. With technology integrated into all aspects of modern life, all companies must continuously evolve to:

- Meet increased consumer demand and market conditions
- Continue to provide quality products as quickly as possible

Historically, IT departments acted as a single team, but they have been increasingly divided into specialized departments or teams with specific goals and responsibilities. This increased specialization is vital for quickly adapting to the evolving technological landscape. However, this division has also created some disconnect between teams when it comes to software development and deployment.

DevOps, ITOps, and NoOps are some concepts that help companies to become as agile and secure as possible. Understanding these concepts is the key to structuring the delivery pipeline at an organizational level. So, in this article, let's take a look at the evolution of ITOps, DevOps, and NoOps.



(This article is part of our DevOps Guide. Use the right-hand menu to go deeper into individual practices and concepts.)

What is ITOps?

ITOps (or TecOps) is shorthand for IT Operations. IT Operations is the most traditional concept of the three we'll discuss, and it's also the basis for these more modern practices.

Any IT task can come under the ITOps umbrella regardless of the business domain, as almost every business domain relies on IT for day-to-day operations. ITOps can apply to virtually any field.

(Understand how operations management can vary from service management.)

ITOps basics

In its most fundamental form, ITOps is the process of delivering and maintaining all the services, applications, technologies, and infrastructure that are required to run an information technology-based product or service. Therefore, ITOps views software development and IT infrastructure management as a unified entity that is a part of the same process. The main difference of ITOps is how it handles delivery and maintenance.

ITOps typically covers all the following job roles:

- System administrator: server and device management
- Network administrator: network and connectivity management
- IT service desk: operations and help desk

The above roles represent the people who are responsible for delivering IT changes and providing long-term support for the overall IT services and infrastructure.

ITOps goals

ITOps are geared more towards stability and long-term reliability with limited support for agile and speedy workflows. Generally, agility and speed are not the primary concerns of ITOps at all. Thus, ITOps will seem inflexible with rigid workflows. This approach is also geared towards managing physical infrastructure with release-based, highly tested software products where reliability and stability are key factors.

This inflexible nature is also the major downside of ITOps. However, it may be an excellent choice for monolithic and slow-moving software developments, such as in the financial services industry. Yet ITOps becomes obsolete in rapidly evolving software developments. As modern software developments come under this category, ITOps is not a suitable candidate for such environments.

What is DevOps?

DevOps provides a set of practices to bring software development and IT operations together to create rapid software development pipelines. These development pipelines feature greater agility without sacrificing the overall quality of the end product. We can understand DevOps as a major evolution of traditional ITOps that is an outcome of the Cloud era.

(Explore our comprehensive DevOps Guide.)

DevOps basics

DevOps combines cultural philosophies, different practices, and tools with a greater emphasis on team collaboration. Moreover, DevOps will bring fundamental changes to how an organization handles its overall development strategy. As mentioned previously, a modern software delivery team consists of multiple specialized teams such as:

- Development
- Quality assurance (QA)
- Infrastructure
- Security
- Support

DevOps aims to bring all these teams together without impacting their specialty while fostering a more collaborative environment. This environment provides greater visibility of the roles and responsibilities of each team and team member.

Automation also plays a key role in DevOps to facilitate an agile and rapid SDLC. It enables offloading most manual and tedious tasks such as testing and infrastructure provisioning into automated workflows. Tools to facilitate this automation include:

- Continuous Integration and Continuous Delivery (CI/CD)
- Infrastructure as Code
- Automation frameworks
- Automated monitoring
- Security tools

DevOps goals

The gist of adapting DevOps in your organization is that it can power previously disconnected tasks such as infrastructure provisioning and application deployments through a single unified delivery pipeline.

For example, in a more traditional development process, developers will need to inform the operations team separately if they need to provision or reconfigure infrastructure to meet the application changes. This process can lead to significant delays and bottlenecks in the overall delivery process.

However, DevOps streamlines this process by allowing separate teams to understand the requirements of each other. It enables them to foresee these requirements and address them promptly. This process can be automated in some situations, eliminating the need for manual interaction to manage the infrastructure.

DevOps is well situated for modern, cloud-based, or cloud-native application developments and can be easily adapted to meet the ever-changing market and user requirements. There is a common misconception that DevOps is unsuitable for traditional developments, yet DevOps practices can be adapted to suit any type of development—including DevOps for service management.

What is NoOps?

NoOps is a further evolution of the DevOps method to eliminate the need for a separate operations team by fully automating the IT infrastructure. In this approach, all provisioning, maintenance, and similar tasks are automated to a level where no manual intervention is required.

NoOps and DevOps are similar in a sense as they both rely on automation to streamline software development and deployment. However, DevOps aims to garner a more collaborative environment while using automation to simplify the development process.

On the other hand, NoOps aims to remove any operational concerns from the development process. In a fully automated environment, developers can use these tools and processes directly even without knowing their underlying mechanisms.

NoOps basics

NoOps is solely targeted at a cloud-based architecture where infrastructure can be less of a burden or the complete responsibility of the service provider.

Serverless architectures are perfect examples of NoOps software delivery where developers only need to create their applications and simply deploy them in the serverless environment, eliminating any infrastructure or operational considerations.

NoOps may seem like the perfect operational strategy. Unfortunately, it lacks proper process management or team management practices baked into the method. Due to that, it may hinder the overall collaboration within a delivery pipeline as well as put more burden on the developers to manage the application lifecycle without any operational assistance.

In most cases, NoOps will be an ideal method to complement DevOps practices by introducing further automation to a delivery pipeline while preserving the collaborative multi-team environments.

Choosing ITOps vs DevOps vs NoOps

In the above sections, we discussed the impact of each of these methods on the software development lifecycle. But what is the ideal solution for your organizational environment? Let's summarize the primary characteristics of each method to find out the answer to that question.

ITOps

- Stability and Long-term support over speed and agility
- Strict, inflexible yet tried and tested workflows
- Primary focus on the IT operations side to streamline the overall IT infrastructure to ensure business continuity
- Geared towards managing physical infrastructure across multiple business domains
- Well suited for legacy release-based enterprise software developments

DevOps

- Bring fundamental changes at an organizational level with a focus on streamlining the overall delivery process
- Increase collaboration and introduce automation throughout the application lifecycle
- Aims to create more flexible and rapid delivery pipelines while increasing the overall product quality
- Can be adapted across any application type, architecture, platform from cloud-native developments to legacy enterprise developments
- Greater flexibility to select tools and platforms depending on the user requirements
- As DevOps is based on CI/CD principles, software is constantly evolving to stay up to date with the ever-changing technological landscape
- Faster feedback cycles to quickly fix and improve the product

NoOps

- Automate everything
- Eliminates the need for separate operations teams while providing all the necessary automated tools and platforms for developers to manage the software delivery
- Relies heavily on cloud services such as serverless computing and containers to provide an environment where there is no concern on infrastructure
- Focuses on speed and simplicity at the cost of flexibility and granular controls.
- Ideal for cloud-focused workloads

As you can see, ITOps and NoOps excel at their domains, whereas DevOps can be considered a more universal approach.

A continuing evolution

ITOps is slowly becoming obsolete due to its slow rate of adaptation to the current technological landscape. (In fact, AIOps is rapidly moving in.)

NoOps is an idealistic approach where everything can be automated. However, it is still a way off as

some critical aspects such as testing and advanced infrastructure and networking configurations require manual intervention.

Finally, we will come back to DevOps. DevOps has gained high popularity due to its adaptability to almost all development environments while improving the agility, speed, and efficiency of the software delivery process. Approaches like NoOps can even be integrated into the overall DevOps process to enhance the DevOps approach further.

Related reading

- BMC DevOps Blog
- BMC IT Operations Blog
- Top DevOps Trends Today
- SRE vs DevOps: What's The Difference?
- What Is DevSecOps? Combining Development, Security & Operations
- Top AIOps Tools & How To Choose