

# ITOPS, DEVOPS, AND NOOPS EXPLAINED



We live in a world defined by technology, speed, and innovation – and this applies to products and the companies that make them. Companies have to continually shift to meet increasing needs so they can get a useful product to market as quickly as possible. Vital to this process is the information technology (IT) team. Every company in the 21st century must rely on technology; indeed for some companies it is their product, perhaps an app or a vital piece of hardware.

There are some prevailing theories on how to structure your IT teams to address your needs. Historically, IT departments seemed to be one single-minded team, but recently IT teams can be structured into separate departments that have specific goals and responsibilities.

A simple reason for a need to differentiate IT needs and responsible teams is that a lot of IT can be divided into two areas: the hardware and software of running a company, and the actual development of software, both internal and external products.

In this article, we are exploring ITOps, [DevOps](#), and related concepts that help companies be as agile and secure as possible. Understanding these concepts is key to structuring your company in the way that works best for your products and customers.

## What is ITOps?

Let's start with ITOps, as it is the most traditional concept and the basis for related concepts. At first glance, information technology operations, often known as ITOps or sometimes TechOps, can seem to cover anything IT-related. Whether your company builds a software product, sells clothing, or

offers tours to Europe, you must have an IT team to provide the tools and processes inherent in working in the 21st century.

Starting with what is included under the ITOps umbrella of activities lets us then define other operational concepts that cover what ITOps does not include, like program and application development.

ITOps often refers to three areas: network infrastructure, computer operations and help desk, and server and device management. Though these areas are flexible depending on your company, these are the most common divisions for ITOps.

Network infrastructure is the hardware and software that is the structure and safety for the company's network. This area can include:

- Installing and managing networking functions for internal and external IT communications
- Managing security and regulatory compliance
- Providing remote access for network users, such as VPNs, two-factor authentication, etc.
- Management internal phone system management
- Overseeing network health and altering network personnel with any issues

The computer operations and help desk area often includes:

- Managing the physical data center, including equipment locations, electricity, cooling, battery backups, etc.
- Auditing network configuration and security to report to outside institutions like corporate auditors, local and state regulatory bodies, business partners, etc.
- Overseeing backups and disaster recovery
- Building and managing an internal IT infrastructure library
- Managing the internal IT help desk
- Creating, authorizing, and managing user roles

The area devoted to service and device management often includes:

- Configuring, maintaining, upgrading, patching network applications and infrastructure
- Managing network and individual user storage to ensure proper disk, memory, backup, and archiving
- Setting up, updating, and fixing PCs and mobile devices for individual users
- Licensing and managing software as needed for user devices

Using these three areas as the basis for ITOps allows a company to examine what other technology needs it has. The next common area is DevOps.

## **DevOps**

DevOps, short for software developer operations, refers to the team of IT application developers in an organization. If we consider the ITOps team responsible for ensuring security, compliance, and reliability, then DevOps works to deliver new or innovative products in as short a time as possible (time to market). Whereas the ITOps team is constantly reviewing and updating internal networking issues and regulatory compliance, the DevOps team is constantly breaking rules in order to improve a final deliverable or product.

Inherent in the name, developers are easily recognized as developing applications or programs. But DevOps can also refer to the many areas that ITOps may not touch, such as business analysis, design, coding, testing, and custom software. Because the DevOps team is working on hard deliverables, they often find that it is easier to maintain their own network administration for pre-production environments.

In older IT structure models, developers would necessarily loop in ITOps to any change they made or any issue found while testing. As programmers and developers become more crucial to companies, this older-fashioned structure can hinder the agility and speed that the DevOps team needs.

Structuring your IT responsibilities under the DevOps theory, then, holds that ITOps should relinquish some areas of responsibility to provide DevOps with the tools they need to develop rapidly and innovatively. In practice, ITOps and DevOps must collaborate and communicate so that these product deadlines are both timely and frequent, while still maintaining the reliability and security that ITOps \ provides.

On a larger scale, DevOps can also refer to a sense of culture within an organization. This culture seeks both organizational stability and change – while reducing risk. While that can seem conflicting at first, companies that successfully adopt the DevOps culture often reap the benefits of deploying agile processes that result in more data and testing automation to increase the rate and success of product releases.

## NoOps

While many companies still strive to reach a DevOps culture, providing separate but vital teams of IT operations and developer operations, some industry experts consider the step beyond DevOps.

Shorthand for No IT Operations, NoOps is a theory that extends from DevOps: that ITOps shouldn't be involved in any tech-related actions. NoOps holds that an IT environment can eventually be automated to the point that no in-house team is required and dedicated to manage the network.

In some ways, NoOps seems like an idyllic concept. Let's consider the positive effects of a NoOps environment:

- **With ITOps automated, the likelihood of human error decreases significantly.** Many issues that arise in IT environments are based on humans making a change in a complicated and complex network. If the change isn't vetted through every single possibility and later stages of the lifecycle, there can be issues that create network outages, reducing work time for many employees. NoOps would eliminate downtime and improve performance because no human intervention would be required in a fully automated system.
- **DevOps can accelerate all collaboration.** In a NoOps work culture, there are no ITOps, so the friction that exists between their regulatory and compliance role and the speed and innovation role of DevOps.
- **ITOps can elevate their role.** ITOps proponents worry that a fully-realized NoOps culture would remove any need for them. Industry analysts recognize this, but also believe that ITOps can become a more consultative and strategic team, partnering with developers and other teams in the company to ensure better access to the tools they need. Some suggest that ITOps can even embed into the developer team, though working with the agile framework of DevOps.

Still, some in the industry believe that NoOps isn't the automated utopia we imagine it to be. With only a single team of developers, and no ITOps, the burden of too much responsibility could hinder app developers from rapid production. This burden means that a subdivision of developers could be possible, such as infrastructure developers separated from application developers.

This subdivision inherently questions whether a true NoOps environment is possible, or whether a re-imagining of ITOps is preferred. If ITOps can become more automated-focused, they can continue to provide a consultative and innovative approach to IT across the entire company – while still allowing DevOps to move forward with as few tethers as possible.

## **Deciding what works in your company culture**

With this understanding of three common theories of IT team structures, you can now take a hard look at your company. Start with how things are currently structured and review what is working and what isn't. What complaints do the IT team members have? If there's a rift between IT and developers, try understanding their day-to-day work and whether a DevOps culture needs to be more fully embraced.

Whichever path you decide, understand that the mission of your company will always have an impact on how these teams function. It's never a one-and-done decision, but something that must continually be reviewed and perhaps revised to best serve your employees – and your customers.

*Last updated:5/31/17*