

# What Is AIOps, BizDevOps, CloudOps, DevOps, ITOps, NoOps?

A Gentle Introduction  
to Digital Business Transformation

# Table of Contents

<a href="#"><u>Introduction</u></a>	1
<a href="#"><u>Defining IT Operations</u></a>	2
<a href="#"><u>The IT Maturity of Your Company</u></a>	14
<a href="#"><u>How to Build Reliable IT Operations</u></a>	17
<a href="#"><u>Summary</u></a>	25
<a href="#"><u>Final Thoughts</u></a>	27



# Introduction

---

Recently, the IT world has been experiencing an explosion of different terms related to operations. The good old days—when the global order was defined around a rule of thumb and IT was separate from business—are gone, never to return.

Instead of simple rules, we're dealing with buzzwords that cause growing confusion and frustration among managers, directors, and CTOs. This, in turn, leads to people misusing those terms, causing even more confusion and frustration.

**The goal of this guide is to help you understand the terminology around IT operations, define your company's business needs, and make educated decisions.** It is not, however, an extensive guide on carrying out the digital transformation of your business.



# Defining IT Operations

---

With so many different Ops terms being tossed around, it's essential to define them before you can decide what comes next for you and your business.

We'll discuss the most prominent terms; the ones you may already be familiar with or have heard about or have seen used. It's not our intention to compete with a Wikipedia entry or any other formal glossary, so we're only going to tackle what you absolutely need to know.

## What is ITOps?

The term "ITOps," or "Information Technology Operations," likely isn't unknown to you. It's a common way of referring to all IT-related operations in a broader sense.

Nowadays, every business domain relies on computer-related technologies. It's hard to imagine a business that operates without a computer network or Internet access.

## Goals of ITOps

The purpose of ITOps is to meet those basic requirements. **It's generally responsible for delivering and maintaining all the services, applications, and technologies necessary to run a business.**

Since this definition is very broad, IT companies often draw lines between different groups within ITOps. For example, they create silos to better define their teams' domains and areas of responsibility, such as:

- administration and maintenance,
- network management,
- systems management,
- technical support (help desk).

Not only teams are siloed, but also processes and technologies, as the companies focus on domain-related issue troubleshooting and usually follow waterfall-based approaches. **While such approaches may seem very specific, they are ineffective and disparate from a development point of view.**

What's more, the definition excludes research and development (R&D) teams by default, since they're thought to be the same type of internal clients as production or office employees. It all depends on the size and structure of a given company.

## Why is ITOps not good enough?

**The traditional ITOps model is seen as outdated and no longer effective for a number of reasons, including:**

- the pace of today's business,
- the constantly changing technological landscape,
- the growing need to shorten the reaction time to business needs.

The existing definition of ITOps is far too broad. The specialized groups we've mentioned before cannot be isolated anymore if you're to keep up with the speed of change.

In order to overcome the challenges of ever-changing business needs and fast-paced market requirements, **ITOps and R&D teams have joined forces to create DevOps**. But before we dive into the world of DevOps, we should mention another term that had its momentum some time ago: **CloudOps**.

## What is CloudOps?

**CloudOps is frequently compared to ITOps and thought of as an alternative to it, but limited to the cloud.** ITOps is meant for traditional data centers, while CloudOps relates only to the cloud.

As cloud computing became more popular and not limited to a public cloud alone, CloudOps gained in popularity. Many larger organizations needed to optimize their resources in a modern and more effective way, leveraging private cloud solutions or hybrid clouds.

## Features of CloudOps

CloudOps is an approach that offers tools and best practices, much like DevOps, but only within the realm of computing resource management.

Two features set CloudOps apart from ITOps:

- 1) cloud-specific flexible provisioning and scalability of environments,
- 2) built-in task automation.

The term and the idea behind it are rather unpopular these days, since they represent such a narrow specialization of the widespread approach to modern Ops.

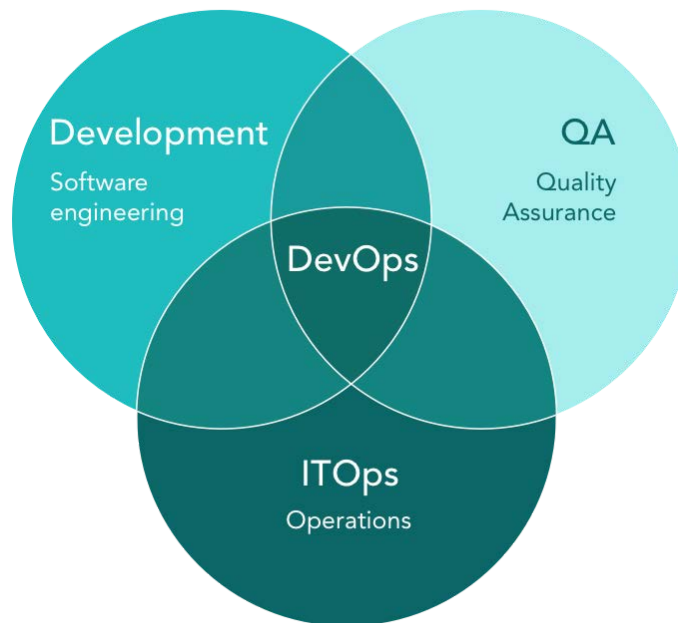
In fact, CloudOps was quickly overtaken when a new, more functional idea called **DevOps** was introduced.

## What is DevOps?

By definition, “DevOps” (“Development and Operations”) is understood as a **combination of software application development (R&D), quality assurance (QA), and ITOps**, with all the best practices, consequences, and methodologies behind them.

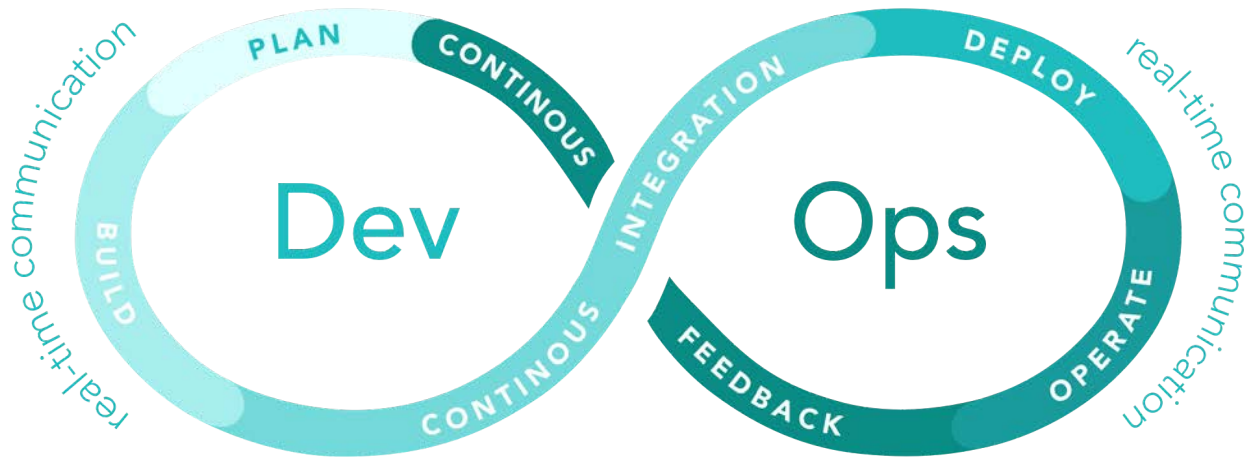
This joint venture is meant to:

- 1) shorten the software development life cycle (SDLC),
- 2) improve the response to market needs,
- 3) shorten the time to market (TTM).



*DevOps as a joint effort and areas of responsibility for R&D, QA, and ITOps*

DevOps companies follow a continuous work cycle, consisting of the following steps:



## Goals of DevOps

DevOps is all about accomplishing its core goals by **automating as many tasks as possible within reason**. An important distinction to keep in mind is that **automation isn't seen as an objective in itself**, but rather a way to:

- shorten the market delivery time of a product or service,
- improve the release quality,
- shorten the time between releases,
- shorten the mean time to repair (MTTR).

The principles of DevOps look great on paper, but they demand strong interdepartmental communication. Because of this, they may not necessarily work that well in reality.

While R&D and QA teams are quite effective in their daily tasks, ITOps teams usually have to acquire new skills in automation. The other approach is to have the R&D teams perform Ops-related tasks.

A side effect here is that the skills of the QA team members tend to focus more on scripting, integrating, and coding. Therefore, their traditional role in testing migrates to test automation, or even further, to process automation.

## Benefits of DevOps

A well-thought-out balance between the efforts and responsibilities of the DevOps participants pays off, but the benefits aren't immediately observable. You may be wondering: why is that the case?

There's no simple and easy answer. R&D and QA teams take good care of:

- automated development,
- building,
- testing,
- deployment of new software releases.

However, Ops teams still manage their previous responsibilities:

- infrastructure (hardware and network),
- operating systems,
- policy compliance,
- incidents.

In smaller environments, some of the tasks usually handled by Ops teams now need to be done by software or test engineers—and the complications don't end there.

## Shifting the responsibilities and blurring the lines with DevOps

The responsibilities of R&D and QA bleed into one another and the boundaries between them become increasingly blurry, since automation is required not only in testing, but also in the development pipeline.

As a result, developers become more involved in Ops-like tasks (provisioning the environment and tools), while test engineers need to use programming skills to:

- automate tests,
- test or stage environment provisioning,
- help enable continuous integration and continuous delivery (CI/CD).

The growing experience of R&D and QA teams within DevOps organizations helps mitigate infrastructure-related issues, offloading the Ops teams. However, in large companies, such a low-level collaboration isn't always possible because of their scale.

In that case, Ops teams are still involved in hardware, network, or systems management—particularly production environments, where optimal resource allocation (virtualization and containerization) and business continuity are key.

## Best practices and tools for DevOps

DevOps isn't just the teams and their responsibilities; it's also **a collection of tools and best practices that cover automation, monitoring, and optimization of tasks and processes.**

A well-composed toolset should be as small as possible to reduce maintenance costs, while the tools should integrate easily with little to no effort. The larger the organization, the better and tighter the collaboration between your teams should be in order to choose the tools and best practices for you.

Your budget is another aspect to consider. When the investments in DevOps resources are considered too high, some companies decide on going with **a rather extreme mode of DevOps implementation called NoOps.**

## What is NoOps?

“NoOps,” which stands for “No IT Operations,” is an approach that can greatly benefit companies with high technological maturity or play a significant role in software development and maintenance under certain circumstances.

## Goals of NoOps

The main assumption behind NoOps is that **developers (or, more generally, DevOps practitioners) are no longer required to take care of operations and may concentrate on software development.**

Under NoOps, all operations are automated, minimizing or even eliminating the risk of human error, and automated processes are faster and less error-prone. With this approach, ITOps teams no longer need to be engaged in technology-related tasks on a daily basis, because **everything that could be automated is already automated.**

## Benefits of NoOps

Sounds like wishful thinking? Granted, the proposal seems controversial and unrealistic today, but its potential benefits are quite clear:

- 1) the risk of human error is considerably reduced or removed;
- 2) DevOps substantially accelerates daily operations and collaboration;
- 3) ITOps becomes a strategic team working on technological advancements, instead of dealing with dull tasks, such as hardware and network maintenance or system backups.

However radical it may appear, certain companies or groups in larger organizations may consider taking the NoOps approach.

For instance, if a company has all their IT infrastructure in the cloud and their application development is mainly based on serverless computing, NoOps might be worth looking into for them.

### What is serverless computing?

Serverless computing is very appealing from two perspectives:

- 1) infrastructure investments—none or almost none (in-house costs only);
- 2) maintenance or administration costs—none.

All of this is handled by a given organization's cloud service provider. The entire infrastructure is proportionally scaled and provisioned, according to the needs and contract levels. Everything is automated and essentially beyond your control, because the provider takes care of the environment.

### **Risks and drawbacks of NoOps**

From a budgeting point of view, **the NoOps approach isn't very efficient**. The costs may exceed the investments on your own architecture, systems, and personnel, while the performance of serverless computing environments may prove too low for production-grade applications.

The opponents of NoOps frequently argue that **the idea is extreme in many aspects, since it assumes highly specific circumstances and the use of serverless computing**. The latter means you have no control over scaling mechanisms and they remain unknown to you (it's a hidden cost of using serverless computing).

Taking these drawbacks into consideration, **tech-savvy and serious businesses usually conclude that NoOps is far too risky**. Choosing this approach may result in crashing the digital business or endangering the digital transformation.

It's common industry knowledge that R&D teams are best at development, but not necessarily at designing, provisioning, scaling, optimizing, or maintaining production environments that run business-critical applications and supporting services. What is often overlooked, or even forgotten, is determining who is going to maintain automation, integration, and administration of the existing systems and applications.

One last downside of NoOps is that it requires the application performance monitoring and management tools to be carefully selected and properly implemented.

## What is BizDevOps?

DevOps, the way it is typically understood, doesn't account for business.

While this might be acceptable in certain cases, agile organizations that aim to respond to market needs quickly have to adopt a **new operational model called BizDevOps**.

### Goals of BizDevOps

**BizDevOps is nothing more than DevOps that takes business into account, using responsive business to give direction to DevOps teams.**

The combination of DevOps with business also serves to deliver new releases to the market even faster, compared to standard DevOps.

Among other factors, this is why **BizDevOps is sometimes referred to as DevOps 2.0**.

### Benefits of BizDevOps

The BizDevOps model makes DevOps teams [collaborate with business in an agile way](#). **The synergy of the teams accelerates the development pace**, requiring the business to own and actively contribute to the product/service backlog.

At the same time, DevOps teams become more responsible for the business side of what they deliver. This is a significant change to the existing *business request*  $\rightleftharpoons$  *development response* model.

### Best practices and tools for BizDevOps

Much like other approaches, BizDevOps also requires special tooling to be successful.

Apart from the standard toolset for DevOps, **user experience analytics solutions ([UX/CX real-time analytics](#)) are crucial for market success**. They allow you to monitor, analyze,

and respond to your customers' needs. Ease of use and integration with the existing tools for monitoring and management are pivotal factors while choosing your tech stack.

Market analysts agree that BizDevOps and digital performance management are **the most important technological drivers of your digital transformation and business development**. Relying on the visualization of key performance metrics and dependencies is mandatory to make informed and educated business decisions while analyzing the business efficiency.

In today's digital market, **intuition isn't a business strategy that pays off**. Your vantage point has to be built around information collected from different sources and angles. Due to the number of data sources and the volume of data, you should consider **using big data solutions**. Simple tools, such as spreadsheets or relational databases, are no longer effective.

You'd be wise not to miss artificial intelligence in this picture, either. Hence, **a new term has been coined: AIOps**.

## What is AIOps?

The term "AIOps" stands for "Artificial Intelligence for IT Operations." It was originally [defined by Gartner](#) in 2017.

AIOps refers to the way IT organizations manage data and information in their environments using [artificial intelligence \(AI\)](#).

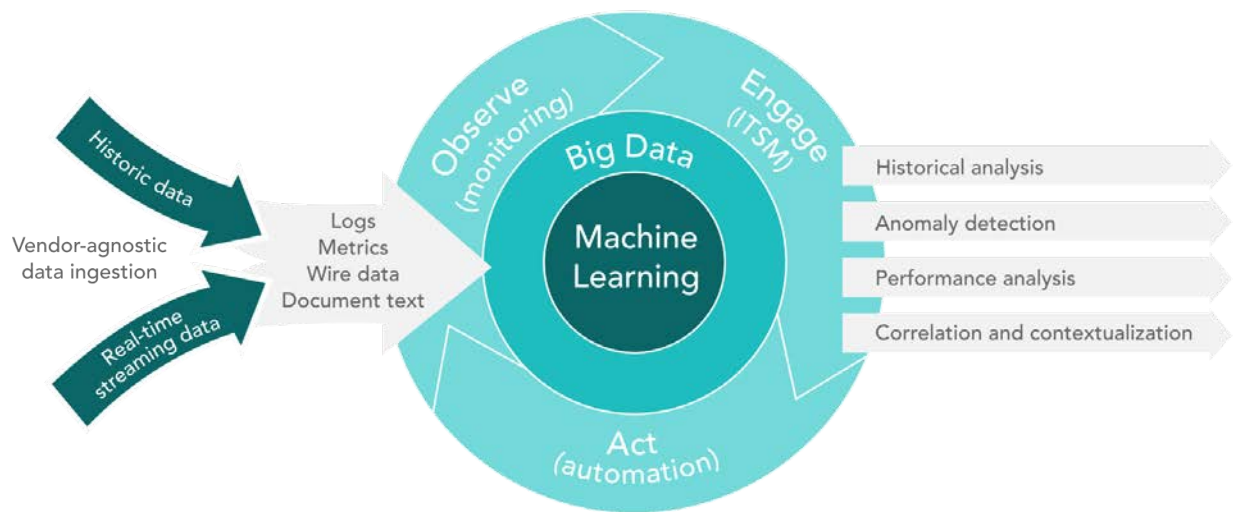
The definition is very broad and non-specific, leaving plenty of room for interpretation. In their efforts to implement more intelligent IT operations, AIOps platform vendors are not in any way limited in how they acquire and utilize:

- big data,
- [machine learning \(ML\)](#),
- pattern recognition,
- other analytics techniques.

## Goals of ITOps

The main goal of AIOps is to enhance your IT operations by providing you with data from multiple sources for analytics and automation.

AIOps is meant to optimize the use of highly skilled engineers for tasks that cannot be automated or where extensive domain knowledge is absolutely essential.



*AIOps platforms enabling IT operations management (ITOM)*

The AIOps paradigm assumes the data collected from all available sources can be shared across all teams. That way, analytics is simplified and experts are no longer required to troubleshoot most of the typical cases.

## Benefits of AIOps

More data from more sources enables multidimensional analysis, but at the same time storing and processing big data requires more resources. Consequently, most companies face common issues with big data solution implementation.

The easiest way to help yourself with this is to simplify your analytics. Nobody wants to waste literal years slicing and dicing data sets, traversing data lakes, and the like. Time is money, and **if your AIOps platform gives you answers with supporting data, rather than raw or processed data without an answer, it's a big win for your business.**

The need for data supporting the answers is especially true for software developers, since they often don't trust artificial intelligence. They usually understand how AI works and whether it can give predictable results. Whatever your AIOps platform of choice is, remember that it should provide you with the supporting data whenever necessary.

## Best practices and tools for AIOps

The goals drafted by Gartner may be realized by AIOps platforms in many different ways.

You can use a commercial or open-source platform that can:

- consume structured and unstructured data from different feeds;
- integrate the data with management platforms already used at your company;
- develop the analytics logic to provide outcomes, as described above.

Alternatively, you may consider a commercial platform that does most of the data consumption and analytics out of the box. However, going down this road means integration with other platforms will require some development on your side.

Regardless of the choice you make, software development is bound to be involved to make sure your investment in the tools bring you the expected returns. [Letting an external service provider handle the development and integration is a smart play.](#)

## Last notes on IT operations

The successful implementation of an AIOps platform in a modern enterprise will always mean that such a platform is going to become a part of your business ecosystem. This is usually accomplished through integrating the business-critical operational systems into a “[single pane of glass](#).”

It's quite natural that the whole process of implementing an AIOps platform must be aligned with the reorganization of your IT operations and should follow an operational model specifically fitted for your business.

One of the tried-and-tested ways of doing this is the Site Reliability Engineering model developed by Google. We'll discuss it at length in a little while.

First things first, though. Before you start moving in any concrete direction, let's make sure you know exactly where your organization stands on the IT maturity scale.



# The It Maturity Of Your Company

---

At this point, we'd like you to ask yourself: **"Where am I? Where am I going?"** Naturally, we mean that in the business context of your company.

Answering the first question is neither obvious nor easy, especially for larger businesses. Even a small team may have doubts whether it fits a specific category. The answer gets more difficult still if your company happens to be large and multi-location, operating on the global market.

However, asking the second question potentially holds even more value. It's much easier to answer, despite the fact that more than one road can lead you to a clearly defined goal for digital transformation.

**In the following section, we'll explain which factors you should consider first and foremost while mapping out a strategy for your digital business management.**

Alternatively, those factors may be important to you simply because you're already in the process of preparing your business for digital transformation and need tech support for your business operations.

## What is IT maturity?

**As IT becomes the core of your business, the performance of your business depends on IT excellence. This, in turn, relies on IT maturity.**

But what is "IT maturity," exactly? After all, if your business is mature enough to operate on the market, it means your IT capabilities are at least as mature as your business, right? Well, not quite.

If we define IT maturity as “a range of capabilities meant to deliver certain outcomes,” the outcomes should be based on the following items:

- organizational structures,
- processes and practices,
- skills and knowledge,
- tools and policies,
- systems and data,
- documents and agreements.

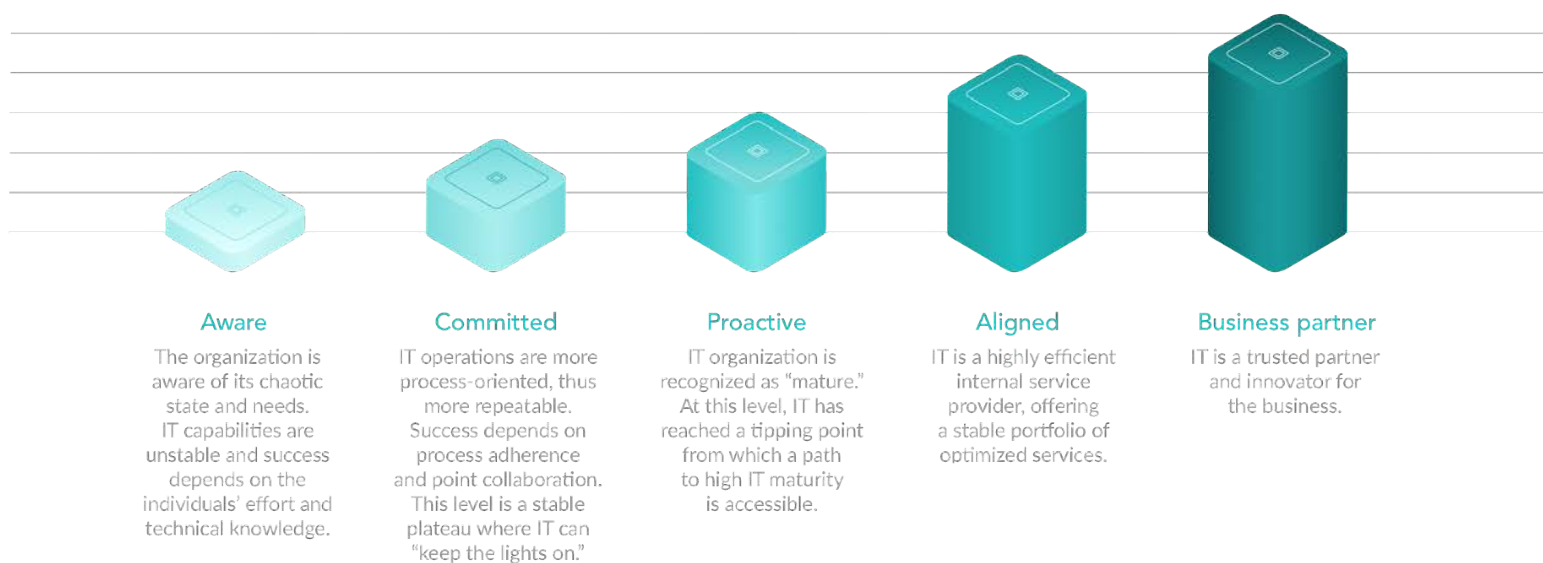
From the IT maturity perspective, these criteria are precisely the foundation for enabling IT excellence, which then brings about business maturity, innovation, and productivity. This whole cause-and-effect is best illustrated in the following sequence:



Generally speaking, IT maturity is how the IT capabilities fit and enable business performance that can be considered “excellent.” Since no two businesses are exactly the same, the capabilities and the use thereof are always company-specific.

**What matters here is relevance.** Even if your business and the IT aspect of it are relatively small, they can still be mature and excellent.

According to IT market analysts, the evaluation of IT maturity can be described as a ladder climbed by a company to achieve high maturity.



There are many maturity models your organization can follow—such as the [Capability Maturity Model \(CMM\)](#)—but it's important to remember that **moving up on the maturity model isn't a goal in itself.**

Achieving higher levels on your model of choice is nothing more than just a movement from chaos toward well-organized processes. To prevent your business from becoming too process-heavy, we recommend you follow this simple piece of advice:

**Your IT should be as high on the maturity ladder as it makes sense for your business.**

This doesn't mean you should stall in your improvement process, though. Just like with every other initiative in an organization, you need resources and a plan. Having a roadmap is extremely important, since it helps you structure the execution of your plan.

[Culture change](#) also cannot be ignored when it comes to IT maturity. Quite the opposite, it should become a part of your company's mindset.

Interestingly enough, only ~10% of all IT businesses self-evaluate at maturity level 3 or above, according to Gartner.



# How To Build Reliable IT Operations

---

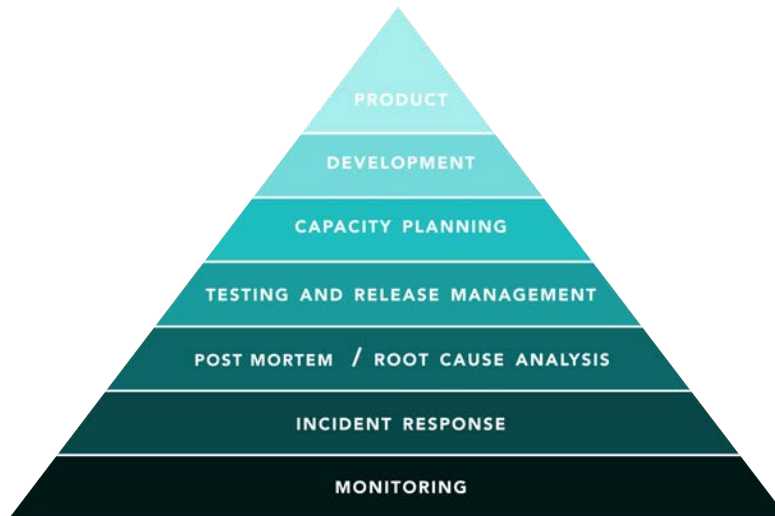
IT maturity and IT management technology are like communicating vessels—**the more mature your IT is, the more sophisticated IT management technologies it uses.** And yet, some businesses still hold out hope that using modern tools will elevate their maturity.

You may be familiar with the phrase “Hope is not a strategy.” It’s very much true, especially for production environments.

In the world of IT, it’s common sense that systems and services don’t deploy and maintain themselves. That’s why we still need people in IT operations and a roadmap that is critical for your business.

## Site Reliability Engineering by Google

Google has defined a model called Site Reliability Engineering (SRE) as **a collection of best practices for any digital business** that takes contemporary circumstances on the highly competitive market as seriously as they should.



The SRE recommends that mature businesses follow this model or a similar one, otherwise the success of their digital transformation is more than questionable.

Business maturity doesn't mean your company has to have a 30-year history on the market. It means that **your approach to running a digital business is mature.**

Service or application delivery and maintenance on a large scale are much more demanding and difficult than automated provisioning of test environments, application deployments, or performance testing.

Such a cycle, while typical for DevOps or BizDevOps teams, is only one of many other cycles that large IT organizations have in place and need to manage every day.

With that in mind, it's no surprise that an enterprise like Google has built such a model. Otherwise, they wouldn't be able to run their business successfully. Therefore, even if your organization isn't likely to reach a scale comparable to Google's, it still makes a lot of sense to follow the SRE model.

Your business maturity should mean you take into account all the layers of the SRE model and build them accordingly, paying particular attention to four attributes:

- people,
- processes,
- technology,
- business practices.

The reasoning behind such an approach is that each layer represents clearly defined business needs that have to be addressed.

## 1) Monitoring

You start with the monitoring layer, because you need to know what happens in your digital business. **The more measurement points you consider, the better you understand how your business works.**

For example, you should measure, among others, the performance of:

- the systems that deliver your applications;
- your application's code on the server side;
- the client side, which translates into the users' experience of your application;
- your line of business (or your small digital business as a whole).

Monitoring is the foundation of the entire reliability model for good reason, so pay attention to how you implement your monitoring.

You'd be wise to engage your trusted partner here, because while you don't have to be an expert in monitoring, you're definitely an expert in running your business.

## 2) Incident management

Managing and responding to incidents in a structured manner helps you build trust among your clients (and employees) in situations when bad things happen. And make no mistake, they *will* happen. So *when*, not *if*, they do—better be prepared.

What your organization needs to do is **define an incident management process to handle the incidents in a repeatable way**. Incident management is an area of the IT Service Management (ITSM) process. Its primary goal is to **restore normal service operation and minimize the impact on business operation**.

Ideally, the response to incidents should be based on and closely integrated with the monitoring layer. It's great if it's automated, as this saves you a lot of time.

## 3) *Post mortem* / root cause analysis

**To improve your business, it's crucial that you know the root cause of the issue.** Otherwise, you won't be able to take corrective action based on facts.

This corrective action is meant to serve a dual purpose:

- 1) help you understand when, why, and where an incident happened;
- 2) allow you to strengthen your business for the future.

In both cases, the goal is the same: **to make sure the issue doesn't happen again, and if it does, to minimize its business impact.**

Again, in a perfect world, root cause analysis should be automated. After all, who has the time to spend countless hours troubleshooting an issue, then analyze multiple application or system logs, application load, and so on? The analysis should also be based on real monitoring data from different measurement points.

#### 4) Testing and release management

**Regardless of the type of product, quality matters.** Before you release any version of a product, make sure its features are carefully composed and thoroughly tested, otherwise you're putting your reputation at risk.

For digital products, there are plenty of testing and release management platforms you can use. Ideally, the test tooling should be seamlessly integrated with the monitoring and incident management platforms. This will enable a controlled approach to solving application performance issues.

**Release management is nothing more than building and delivering software**, meaning:

- source code management,
- configuration management,
- build automation,
- test automation and integration,
- package automation and management...

...and likely a couple more areas of software product delivery.

If it sounds complicated, that's because release engineers really are crucial to providing an **automated, reproducible, and reliable way of building binaries and configurations.**

#### 5) Capacity planning

Planning for capacity may sound like the wave of the future, but it's essential for a business to be successful.

We've all heard dozens of stories about organizations falling victim to their own success. All the failures in those stories have the same underlying cause: those companies weren't prepared for their business to grow to a scale they hadn't planned for.

**This is why capacity planning plays such a significant role in digital transformation.** Traditionally, you can use monitoring data from different measurement points collected over a longer period of time to understand how your business behaves at a load. Then you can use approximation or some other trend line technique to predict your business needs at a larger scale.

**As your company and product grow, it's easier to scale when you invest in their growth properly.** Infrastructure and [sales force](#) are a couple of examples where capacity planning makes a clear difference.

Alternatively, you can use the so-called "intent-based capacity planning" promoted by Google. In essence, it's based on rather abstract requirements: performance metrics, domain knowledge, and some specific algorithms. Of course, what works for Google doesn't have to work for everyone, but taking a simpler approach based on approximation will work in most cases of a lower scale.

## 6) Development

Having all of the underlying tiers built, you can now focus on providing a "playground" for your product developers. And not just software developers, but also UI/UX designers and other contributors who have a say in what gets developed and how.

Large companies like Google usually do their [software engineering in-house](#), which might not be completely applicable in your case. Fortunately, that doesn't mean the SRE model isn't used in cases where software development is handled by an external company.

**Whether you call it "outsourcing" or "team extension," using development teams from a service provider aligns with the SRE model perfectly. External teams:**

- **fit into your company's culture and tech stack,**
- **use the same toolchain as you,**
- **follow the same principles as the other teams within your organization.**

When you develop a software product in an environment that has been carefully designed and prepared according to the SRE, you can be sure that all the moving elements of

product development and launch on the market support your initiative and organization in successful operations.

## 7) Product

Last but not least, the star of the show: your product.

You are now ready to launch your working product and make it usable from the very first day on the market. You have a number of more or less systematic approaches to choose from, all of which have a common denominator: **the launch checklist**.

The SRE provides a wide selection of examples and guidelines on how to develop your own launch checklist, as well as the best practices for production services.

We suggest that you study the SRE closely for details to get a full picture of how these practices can be implemented in your product launch.

## Communication and collaboration

Neither communication nor collaboration are included in the SRE model, but they are essential for successful operations.

**It's a challenge for every organization to build effective and reliable communication.** Whether they follow the SRE or not, they still face the same challenges as the rest. However, the model helps you address and solve many of those problems by implementing communication around the core of the operations: the production meetings.

Production meetings are service-oriented and held by the SRE team. **The composition of your SRE team should be cross-organizational**; the more diversity it has, the better off it is, since such teams pay more attention to communication.

The goals of the production meetings are two-fold, clear, and simple:

- 1) make sure that every participant leaving the meeting knows what's going on;
- 2) improve the services by sharing insights into how the services work with other participants.

The organization of the meeting, the roles, the responsibilities, and the outcomes can either be specific to your organization or follow Google's recipe.

**Collaboration is instrumental because the SRE model has many layers and expertise from each layer should bleed into the adjacent layers.** This helps bring people from all layers together, giving every participant a greater understanding of the operations.

Active participation in the meetings and better communication skills will improve the entire collaboration and prevent your team members from falling into the trap of feeling akin to a recipient on an email distribution list.

## Automation

It's always an option to run your processes manually. People have done it this way since the beginning. But is it even reasonable to continue doing so these days?

Imagine manually collecting and processing data in the following processes:

- discovery of infrastructure components;
- real-time monitoring of hardware, systems, applications, and users;
- management of complex process flows;
- automation of complex service delivery processes;
- aggregation and visualization of collected metrics and trends consisting of myriads of time series.

**Without the automation of IT operations, you wouldn't be able to run a digital business.** Whatever you want to accomplish on the maturity ladder, however you want to follow your roadmap, think of automating your core areas and processes as a must.

## Should you use a reliability model for your business?

**You might think the SRE model is only good for large enterprises, but it actually scales well both ways.**

Whether you're a startup or a well-established, full-grown business, the SRE is like a baseball hat—one size fits all. It's just a matter of what you choose and how you decide to implement it.

Even during the stage of [building an MVP to validate your business idea](#), you have to account for all the mechanisms that will let you know:

- what and how your clients do (monitoring),
- when your MVP has issues (incident management),
- how it behaves under load (capacity).

You will also need to collect feedback for further development. Doesn't that sound deceptively similar to the DevOps cycle of continuous improvement we've described above?

**If you plan on making your digital business more innovative, competitive, and successful, your digital transformation strategy should include measuring key performance indicators (KPIs) on every level of Google's Site Reliability Engineering pyramid.**

Therefore, you need to select the tools that will enable you to deliver on all those layers and continuously monitor their efficiency or performance. It's also entirely sensible to use the services of an [expert third-party consultancy](#) or get an external professional to [handle your integration/implementation needs](#).



## Summary

---

Long story short, here are the most important takeaways from this introduction:

**1) Determine your ITOps model, the direction you want to move in, and a roadmap.**

It's a necessary step before you proceed. If you don't know your location on the map, there's no way your journey can be successful. Sooner or later, you'll hit the rocks.

**2) Find a trusted partner to help you build your approach to a reliable digital business.**

Trust comes from successful, long-term relationships with partners you rely on. Don't be afraid to build your business with the support of experts in their respective domains.

It's fine to turn to more than one partner for help, but you should avoid creating each layer of your model with a different partner.

**3) Automation saves money and time.**

While your initial automation efforts may be significant cost-wise, the investment pays off over time. There should be room for automation in every aspect of your digital business.

**4) Estimate the project accordingly, based on your future needs.**

Be agile, change your organization's culture to support DevOps, and enable digital transformation. Start small and slow, but plan for the big and fast. Work closely with your partner, so that you're both aware of all the aspects of your business.

**5) Include integration of your toolchain.**

Disparate or loosely integrated tools can cause more harm than good. Choose your tools wisely (your partner can help you with that), so your integration doesn't become the development of another platform, adding unnecessary layers and complexity to the model.

**6) Don't try to be perfect.**

Start simple and iterate. You will add complexity over time, as your solution grows. [Always be agile](#), no matter what you do.

**7) Digital transformation is not about applications.**

Instead, it's all about your company's [culture change](#).



## Final Thoughts

---

Thank you for reading our guide to IT operations terminology and digital business transformation. Hopefully, you will no longer look at terms like DevOps, NoOps, or AIOps with confusion after reading this and your transformative efforts will be a breeze.

And now that you have all that operations-building knowledge, it's time to decide what comes next for your business in this regard.




We have a more detailed primer on DevOps specifically that gives particular attention to the benefits of this solution for business managers. [Go ahead and give it a read](#) if you feel you need a bit more persuading to try and introduce DevOps to your organization.

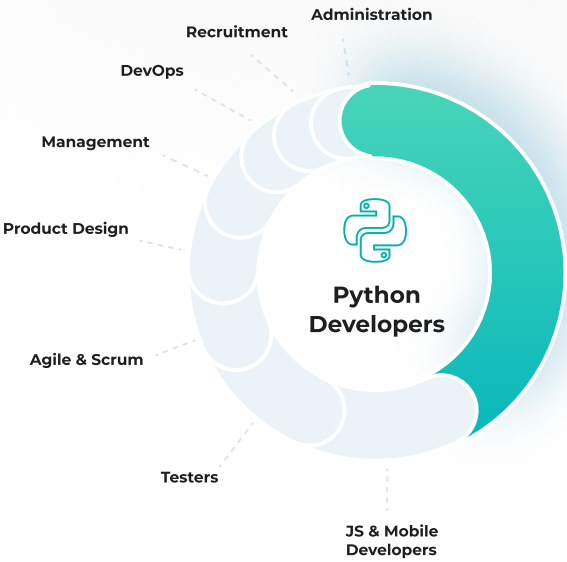
But if you're convinced enough and already thinking of transforming your business with DevOps, [we strongly encourage you to get familiar with the services we offer](#).

STX Next is willing and able to meet any and all DevOps needs you may have.

# Hire an exclusive Python development team







Accelerate your software project with Europe's largest Python software house.  
For companies with big projects and fast deadlines.

 <b>Team Extension</b>	 <b>End-to-End Development</b>	 <b>Consulting &amp; Expertise</b>
Additional developers or experts supporting your development efforts within 14 days	Full development team taking your project all the way from discovery to deployment	Solving your problems or improving your product with the help of subject matter experts



Over 400  
developers

Ready to empower any project with well-reviewed code and a results-driven Agile process

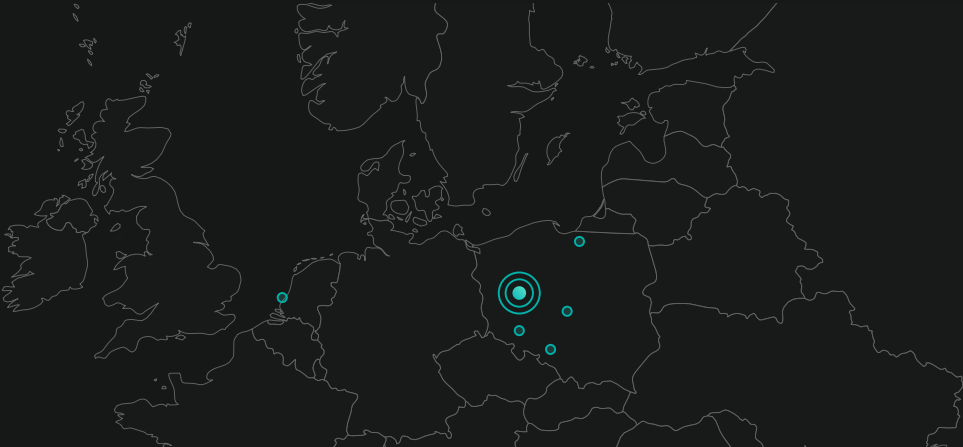
 <b>17+ years</b> market experience	 <b>750+</b> projects delivered	 <b>3.5+ years</b> average partnership
 <b>300+</b> clients served	 <b>550+</b> professionals on board	 <b>6.5+ years</b> average experience of our developers

## Locations

Poznań (HQ) ●

Mostowa 38  
61-854 Poznań, Poland  
+48 61 610 01 92

- Wrocław ●
- Olsztyn ●
- Katowice ●
- Łódź ●
- Hague (Netherlands) ●

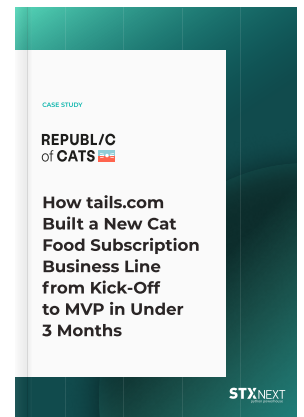
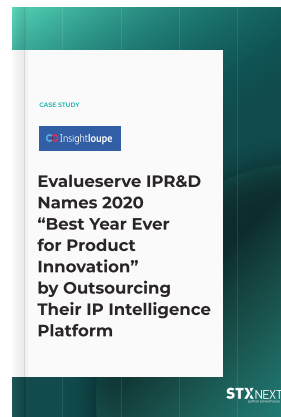
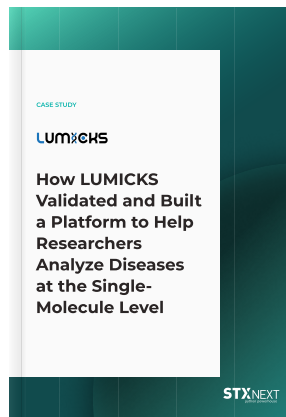


# Resources

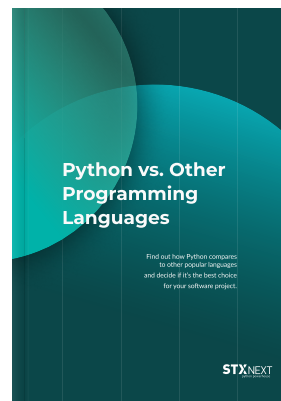
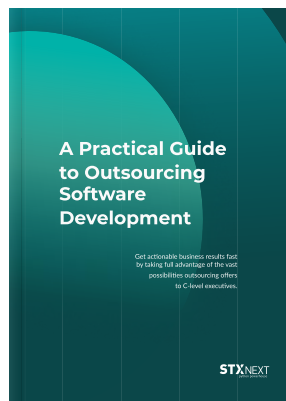
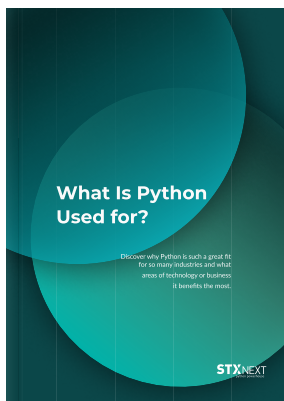
Arm yourself with the expert knowledge you need to successfully deliver software projects.

Get free in-depth resources, templates, and checklists—all based on 17+ years of software development experience.

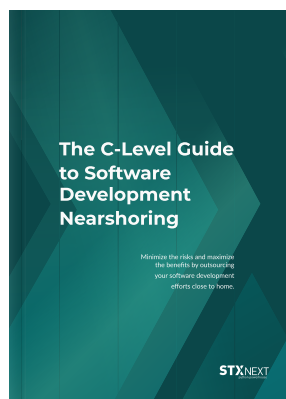
## Reports and case studies



## Guides



## Ebooks



DISCOVER MORE →

# Services

Your project is all that matters. We'll build it like it was our own. Whether it's team extension, end-to-end product development, or expert consulting you're after, we'll do everything in our power to meet your needs.



Python Development



JavaScript Development



.NET Development



Web Development



Software Testing & QA



Mobile Development



Django Development



Node.js Development



React Native Development



Fintech Development



Machine Learning



Data Engineering



DevOps



Product Design



Discovery Workshops

REVIEWED ON  
**Clutch** ★★★★★  
82 REVIEWS



## Tell us about your project

Speed up work on your software projects and outpace the competition.

HIRE US →



**Marta Błażejewska**

DIRECTOR OF SALES

marta@stxnext.com

+48 506 154 343



**Sebastian Resz**

HEAD OF SALES

sebastian@stxnext.com

+48 690 433 578

Follow us

