iTMethods.

itmethods.com

WHITEPAPER:

# Focus On Your Core Business, Not Your DevOps Tools

A managed DevOps platform provides organizations a modern suite of tools to support application development and operations without distracting the team from core business goals.

# Executive summary

Software is now absolutely critical to organizations outside of the traditional “tech” space. Software development in organizations of all kinds is also becoming more complex. A modern software development organization depends on dozens of different tools to keep their software factory running, their team members productive and their software high-quality and secure. DevOps tools make modern software development possible, but they also create their own overhead, as engineers have to devote time and resources to updates, integrations and building tool-specific skills. In this white paper, we’ll address how a managed DevOps SaaS Platform can provide state-of-the-art software development tools while freeing up engineering resources to focus more on their core business.

# Introduction

What does the digital transformation look like, in practice, at companies for whom technology is not their core industry? In most cases, it means an increased prioritization of **moving the IT operations from being a cost center to making it a profit center**, by using software to create amazing digital experiences for both customers and employees. This is a major shift in organizational priorities as it relates to software development and requires a corresponding change in how organizations manage their software development processes.

As software development and the larger IT organization become profit centers, **developing high-quality software is simply table stakes**, a core part of how organizations in any industry build a competitive advantage. Speed is also critical — there is a never-ending arms race of good customer experiences. The ability to understand how customers interact with your software, what features delight them and which ones fall flat, and to make rapid changes to the production software based on that information **is the only way that organizations can stay ahead of the software curve.**

Lastly, **software security is more important than ever** while also being increasingly complicated. Securing modern software is so challenging that few companies do it well, and the ever-changing security landscape means that even experienced infosec professionals are constantly struggling to stay current on cloud security best practices.

So how does this relate to DevOps tools? DevOps tools are what keep the software development process humming as quickly as possible, keeping the feedback loop from end user to developer as short as possible. Tools are also what allow infosec professionals to manage the dizzying array of security configuration and scanning, as well as to automate as much of the security process as possible. **The problem is the tools can also require a lot of care themselves.**

# The tooling challenge

Tools are great — and modern software development would simply not be possible without them. But they also create their own sets of problems for enterprises. Here's the most common challenges we see organizations struggling with in terms of managing their tools.

## 1 Software currency

In an organization with 20-30 DevOps tools — a relatively standard number for an average IT organization — **simply keeping all of those tools up-to-date is a non-trivial task**. Each tool will have its own release schedule, and there will also likely be unscheduled patches to handle. In most organizations, there's also a mix of commercial tools and open source platforms, which come with their own operational challenges.

**Neglecting software updates is also not advisable**, but something too many organizations do simply because they are overwhelmed. Many updates have important security fixes, and if updates are neglected for too long an **organization leaves themselves exposed**. In addition, many of the tools have to integrate with each other, and neglecting updates can lead to problems with backwards compatibility and integration with other tools.

## 2 Security

How secure are your DevOps tools? How confident do you feel in their security?

DevOps tools are business-critical and have access to all of the code an organization runs, both in internal and external-facing applications. A compromised DevOps tool could be devastating to the organization. Yet with dozens of different tools, each with a different update schedule and different security best practices, **most organizations can't confidently say that their DevOps tools are secure**.

At the same time, infosec professionals are already struggling to keep up with all the new attack surfaces related to operating in the cloud, using containers and Kubernetes and relying on open source that they simply can't focus on everything at once.

## 3 Expertise Gaps

**Every time you add a new tool, your team has to learn how to use it.** They have to figure out exactly how it integrates with other tools, how it fits into their normal development workflow and what the best practices are for getting the most out of the tool.

Things are already moving fast. Your team is probably still learning how to use containers, what the best practices are for cloud native applications and how to get the most out of microservices. **The DevOps tools are also continually changing**, and team members have to keep up with a never-ending onslaught of updates and UI changes.

As a result, **they never feel like complete experts in all of the DevOps tools** they use as part of their workflow. In all likelihood, this expertise gap also means they aren't taking full advantage of most tools and aren't even aware of how they could use existing tools to increase productivity.

## 4 Diluted focus from business priorities

The reason you have so many tools is to improve your ability to deliver high-quality software as quickly as possible. But when the team is focused on managing tools instead of using those tools to deliver software, they can end up getting bogged down in upgrades and security configurations, **hampering the end goal of using software to deliver business results**.

DevOps tools exist so that engineering teams can increase their deployment frequency, reduce the lead time to change and reduce the change failure rate without having to roll their own tools to handle tasks like continuous integration or container image scanning that all companies need to do, regardless of their core business. There are commercial and open source tools to handle these tasks because organizations don't want to spend time building internal tools from scratch.

DevOps tools really should just work, with as little management overhead as possible. Because **the end goal is to focus as much energy on the truly differentiated work** — the process of using software to meet your customers' desires in unique and profitable ways — that only your business knows how to do. Very few businesses consider their ability to manage DevOps tools a competitive advantage. But how can they get that focus back?

# Approaches to the DevOps tool conundrum

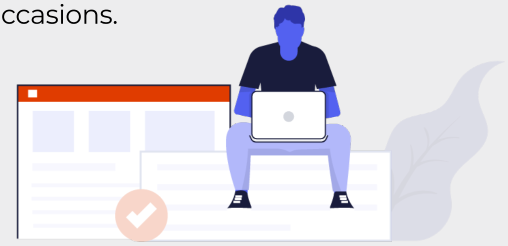
So how should companies manage their DevOps tools, so that they're spending more time on their core business and less time managing repetitive, manual tasks?

## 01 DIY

At every organization there is going to be a build versus buy discussion, one that will likely involve selecting DevOps tools as well as other software infrastructure tools. **The core questions to ask when you consider whether or not you should build a tool internally are:**

- Do we get a competitive advantage from building this tool?
- Is our team uniquely suited to build this tool?
- Is our team uniquely suited to operate this tool throughout its entire lifecycle?
- Is building this tool part of our team's / company's core competency?

There may be situations where it makes sense to build a tool internally, but if the answer is no to those questions, **building a tool or tools internally is likely a distraction from your core business.** Even in the most technically sophisticated organizations, building a tool internally will only make sense in rare occasions.



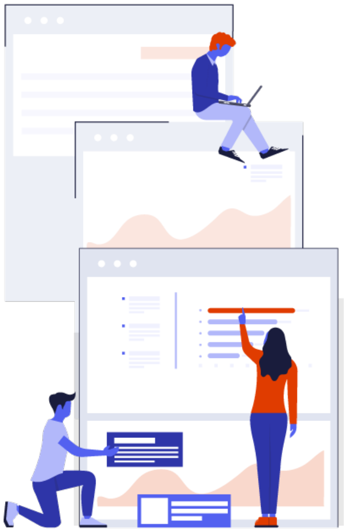
## 02 Stringing together point solutions

A more common approach to implementing DevOps within an organization is to buy a series of point solutions, then to DIY the integrations, sometimes also integrating some completely homegrown tools into the mix. This approach is certainly better than a complete homebrew scenario, but **it also leaves the engineering team managing integration points, upgrades and security on their own.**

## 03 Fully managed platform

The third alternative is to consume all of your DevOps tools as part of a fully managed, integrated platform. This approach moves the responsibility for management, integrations, upgrades and security away from your internal engineering team to the platform provider. **A fully managed platform maximizes the internal team's ability to focus only on the core business**, and in most cases also results in a more secure DevOps toolchain, because responsibility for security is with the platform provider, not an internal IT team trying to do hundreds of other things.

# What do you get with a managed DevOps tool platform?



The bottom line for most organizations is that there's no competitive advantage from managing your entire suite of DevOps tools internally. Managing the tools, keeping them secure and up-to-date, is the foundation of everyone's software factories and critical to get right. And yet these tasks, which are so critical to the health of the software that the business depends on, do not provide any competitive advantage. All businesses have to do it but no one has any reason to be the best at it.

**Here are some of the primary benefits organizations get from using a managed DevOps tool platform.**

## Easy Migrations

In most organizations, the first step to getting value out of cloud-native development practices is planning and executing a migration and modernization project. In a best-case scenario, migrations take months, and require entire teams to learn new skills while they make decisions that can have lasting technical consequences.

According to Gartner, the percentage of enterprise workloads in the cloud will double between 2020 and 2023, from 20% to 40%. That means that many engineering teams are currently grappling with how to make the migration as painless as possible in the short term while also putting all the right foundational elements in place for long-term success. Using a managed DevOps platform is a key way to meet that objective.

> **Modern software development depends on tools, and one of the first steps in any migration process will be deciding which tools the organization should use, configuring those tools and ensuring that they integrate with each other.**

Migrations are already very complex. They involve specialists from a variety of disciplines, from network engineers and storage admins to infosec teams and even business leaders. In most organizations, **none of the specialists involved in the migration to a cloud-first software development model are experts in cloud technology** — they are all learning as they go, while trying to make good decisions and collaborate with interdisciplinary colleagues.

**Managing integrations in the DevOps toolchain is a big part of putting the groundwork in place** for a successful migration and modernization effort. When you have dozens of tools, all of which are new to the team, and need to understand how those tools interact, how the dependencies work and where the failure points are, **just setting up the DevOps toolchain can be very time consuming**. It can also be a source of invisible risk: Integrations have to be set up securely, and if they aren't set up according to best practices at the outset, this security risk can easily slip under the radar once the tools are in regular use and the integration points aren't regularly evaluated.

**Lastly, in an enterprise setting there are internal and external compliance frameworks that can't be ignored just because the team is moving to cloud native development. Keeping track of compliance across dozens of tools quickly becomes overwhelming, and can make proving compliance during an audit nearly impossible. It also becomes challenging to control the entire toolchain centrally, or to put into place guardrails on how the individual tools can be used or configured. With a managed DevOps platform, central compliance teams can easily ensure compliance across the entire tool suite, prove that compliance, and control the tool settings and configurations centrally themselves.**

## Operations

Even if the migration and modernization process is expertly managed, operating cloud workloads is fundamentally different from running workloads on-prem, and requires a different approach, one that is intimately connected to your DevOps tools.

**Operations teams are ultimately responsible for the same metrics in a cloud environment as in on-prem — ensuring high availability, setting up disaster recovery, avoiding single failure domains. They are also responsible for ensuring observability into the entire stack, from infrastructure layer to application layer, and having a way to store, process and access relevant metrics when needed.**

The differences between a cloud native and on-prem environment, as well as the major differences in application architecture, mean that operations teams and the tools they rely on are still playing catch-up to common development practices. **Successfully operating a cloud-native application is very different from operating an on-prem application**. Everything from the tooling involved, both third-party and from cloud providers, to the amount of log data, to the ephemeral nature of containers means the operations teams need a fundamentally different approach.

This means ops teams are on an even steeper learning curve than application developers as workloads are moved into the cloud and refactored to be more in line with cloud native best practices. As organizations prepare to put their migrated applications into production, **the ops team needs tooling that will help them get up to speed as quickly as possible**, without having to think about how to get visibility into the cloud environment or how to deal with the exponential growth in log data.

A managed DevOps tool platform helps ops teams stay productive and successful in keeping applications running in the cloud highly available and secure.

**A managed DevOps platform:**

- Ensures ops teams have the data they need when there is an incident;
- Prevents ops teams from making costly mistakes simply due to lack of knowledge;

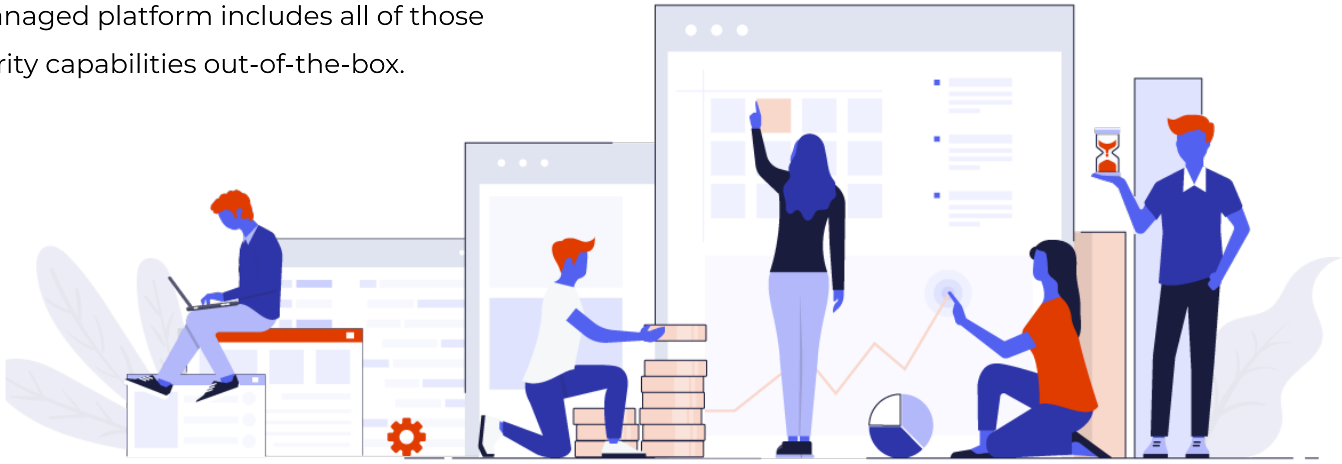
A managed DevOps platform also has major advantages when it comes to security and compliance. Continuous security and compliance are critical in the cloud, and most organizations find that their **legacy tools can't handle the compliance and security challenges unique to cloud environments**. A managed DevOps platform has built-in security and compliance controls that are built for the cloud. Even if these controls are carefully managed during the initial development or migration process, **ops teams need a way to continuously review and adjust security policies**, as well as understand the application's actual behavior. A managed platform includes all of those security capabilities out-of-the-box.

## Tool Optimization

The software development process in a modern organization is never 'finished.' Even once the application has been successfully deployed, it is far from "done." Instead, it has simply reached the production phase of its lifecycle, and will continue to be updated, iterated on and changed for months or years to come.

The migration to the cloud may or may not at some point be 'over' — for many organizations, some applications stay on-prem indefinitely, even when most of the workloads are in the cloud.

Regardless, **the DevOps tools should not be any more static than the rest of the software engineering organization**. Organizations who self-manage their DevOps tools generally have a set-and-forget mindset about their tools, only making changes when something goes wrong, a tool needs to be replaced or there is a critical upgrade to manage.



That would be a poor mindset to bring to software development, and it’s no more appropriate when applied to the DevOps tools. With a managed DevOps platform, teams have access to tools that are being continually optimized, updated, and tuned. This leads to:

- **Better availability and application performance.** Tools are optimized to keep the software development pipelines up and running and notify teams of any potential issues as soon as possible.
- **Better security.** Security is an ever-evolving target, and having a service provider constantly on the lookout for potential security vulnerabilities and proactively addressing them leads to a better overall security posture.
- **Cost optimization.** Running a DevOps toolchain can become expensive, especially because the tool costs can influence cloud costs. Using a managed DevOps platform gives organizations access to continued cost optimizations, both as related to individual tools in the platform as well as how those tools drive cloud costs.
- **Increased innovation.** The tools your team uses are what allows them to innovate by freeing up time and mental resources to think creatively about how to use software to meet customer desires. There are constantly new features being released as part of the many DevOps tools available, but most companies who self-manage their tooling don’t take advantage of them, often because they require infrastructure changes or integrations. With a managed DevOps tool platform, all the engineers automatically have access to the latest tooling to help them innovate as quickly as possible.

# Using the iTMethods DevOps SaaS Platform

What can you expect with a managed DevOps Saas Platform like iTMethods'? Here's what organizations get after moving to a fully managed option for their DevOps tooling.

## More productive developers

Using a managed DevOps platform frees up more developer time to focus on application development than stringing together point solutions on your own, and much more time than attempting to create DevOps tools in-house. With a managed DevOps platform, the initial time to value is much shorter than with point solutions because there is minimal set-up, integration or configuration work to do. At the same time, there is much lower continued operational complexity, because updates, security patches, integrations and any new feature activation is all taken care of by iTMethods.

DevOps tools should fade into the background for developers. They should just work as expected, every day, without causing deployment delays, security incidents or frustrating drains on productivity. The best way to have a complete solution that 'just works' is to go with a managed DevOps platform like iTMethods'.

## Lower overall costs

Dozens of individual contracts with point solutions adds up. So do the cloud costs and HR costs from operating open source projects.

Users of iTMethods' DevOps SaaS Platform find that their overall costs go down with a complete, managed DevOps platform compared to managing the entire toolchain in-house. Bundling access to all tools in one package reduces the cost of the tooling, outsourcing the management overhead reduces the need to salaries internally just to manage tools and the continual optimization efforts from iTMethods' team leads to progressive reduction in cloud costs related to the DevOps toolchain, from networking charges to storage.

### Better focus on the core business

The bottom line is that a managed DevOps platform like iTMethods' helps organizations rapidly modernize their development and operations workflows in a sustainable way, without incurring technical debt, and easily maintain the tools on which their modern workflows depend. This gives everyone in the engineering organizations the ability to focus more on solving engineering problems unique to their business and providing unique value to their customers.

## In conclusion

For most companies, the constraining factor in their software development process is developer time and productivity. Hiring new developers is slow, expensive and distracting, so making the current team as productive as possible is a top priority for engineering leaders. The iTMethods DevOps SaaS Platform helps organizations reach that goal both by ensuring engineers have access to the most up-to-date productivity-enhancing functionality in the tools they use, while also freeing engineers from managing those tools themselves.

The iTMethods DevOps SaaS Platform gives software engineers the maximum amount of freedom to focus on finding creative ways to use software to meet the unique requirements of their industry and the specific desire of customers.

# About iTMethods

iTMethods is helping Fortune 1000 and Digital leaders transform software development in the cloud by delivering 45+ leading DevOps tools, including those from Atlassian, AWS, CloudBees, GitHub, GitLab, Sonatype, and many more as a Managed/SaaS Service.

Our single-tenant Managed DevOps SaaS Platform is deployed in an AWS Region of your choice and integrates your tools into a seamless toolchain in the cloud and allows you to:

- Redirect internal resources to business priorities
- Gain insights and agility for your software teams
- Optimize operational performance
- Maintain software currency and high availability
- Achieve best practices in enterprise architecture/operations and more

### Here is what makes our Platform #1 in enabling Enterprise DevOps:

- **Secure, Compliant & Integrated.** SOC 2 Type 2 and AWS certified, supporting customers in highly regulated industries such as Banking/FinServ, Pharma/ Healthcare and with EMEA data residency and privacy legislation in place is our jam.
- **Reliable, Scalable & Resilient.** iTMethods' Platform features Hybrid Network Connectivity supporting both On-premise and Multi-Cloud (AWS, Azure, Google) deployments. And you have full control of your environment.
- **Cloud-Native Migrations and Integrations.** All your tools, secure, managed and integrated in one place. It's that simple.
- **Rapid deployment capabilities (1-3 months).** Leave lengthy deployments that slow you down in the past.

Successfully operating for 15+ years, we are a market leader in enabling Enterprise DevOps. Our proprietary Managed DevOps SaaS Platform enables companies to integrate, migrate, and modernize their multi-vendor, multi-cloud toolchain so they can build better software, faster and more securely. Our capabilities include supporting 45+ DevOps tools from leading vendors such as Atlassian, CloudBees/ Jenkins OSS, GitHub, Sonatype, GitLab, and many more. These tools are deployed to each customer's specific requirements, including security, scalability, and 24/7 customer support.

The result is our DevOps SaaS Platform securely integrates across multiple teams, multiple processes and workflows, and multiple tools / stacks to achieve excellence in Modern Software Delivery.

**Learn more at www.itmethods.com**

# Get in touch

☎ 877-533-8660

✉ innovate@itmethods.com

itmethods.com

iTMethods.