# **DevOps Toolchain: Overview**



## **What is DevOps?**

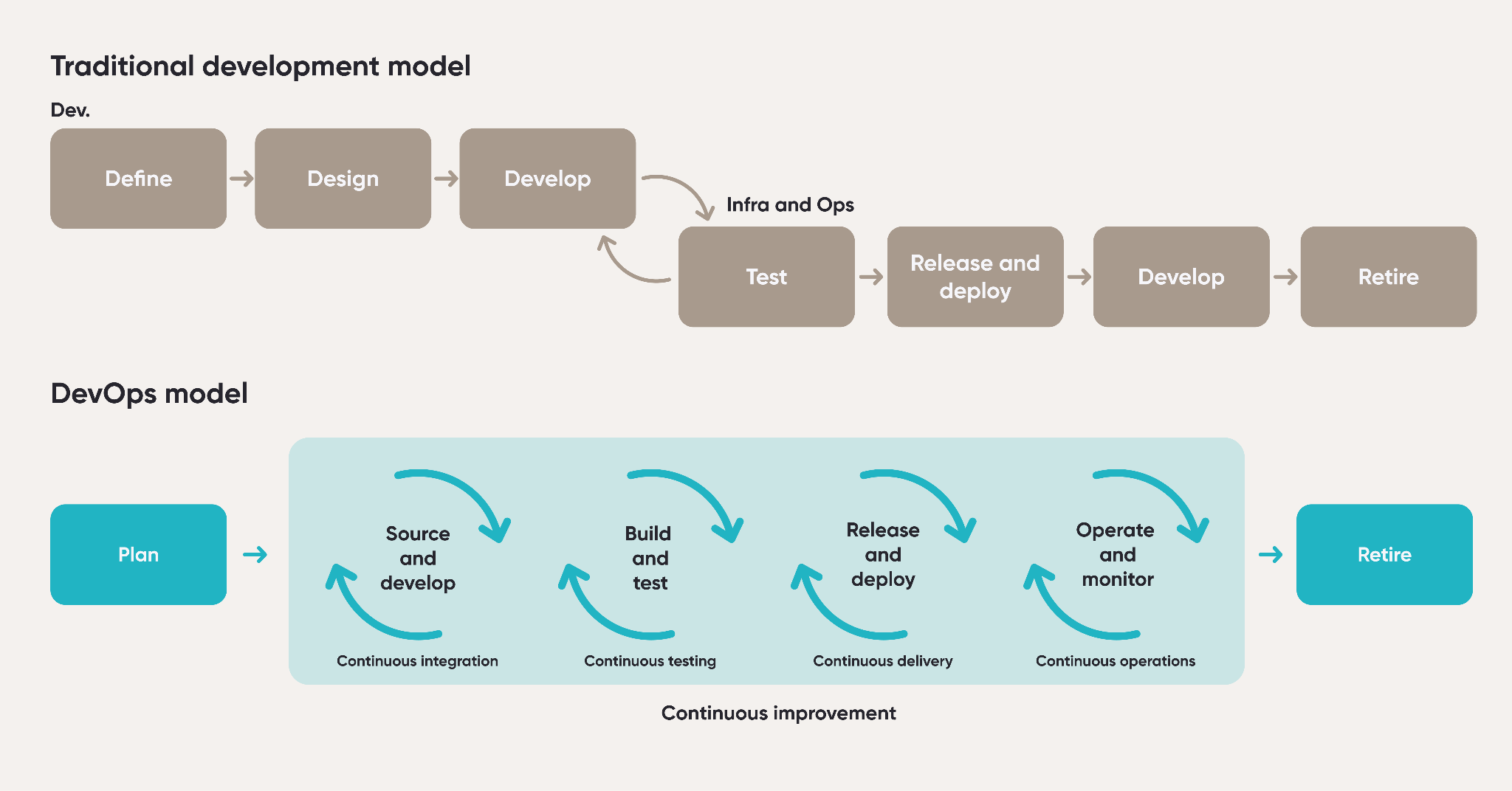
DevOps is a software development methodology that involves collaboration between development (Dev) and operations (Ops) teams via frequent releases, with incremental code updates throughout the development life cycle. It allows companies to promptly identify and rectify failed deployments and to swiftly introduce new features to customers, minimizing the risks associated with deployments.

There has typically been a lack of communication and collaboration between software development and operations teams. The development team would write code, followed by validation by the testing team. The code was then sent to the operations team for deployment. Developers usually worked in a controlled environment, enabling them to address bugs and refine requirements without compromising system stability.

However, difficulties arise when deploying the code into the live production environment due to discrepancies between the developer environment and the dynamic production environment. These disparities result in significant setbacks and amplify the divide between development and operations teams.

#### 

#### **DevOps vs. the traditional development model**

****

Source: Deloitte, recreated by SPEEDA Edge

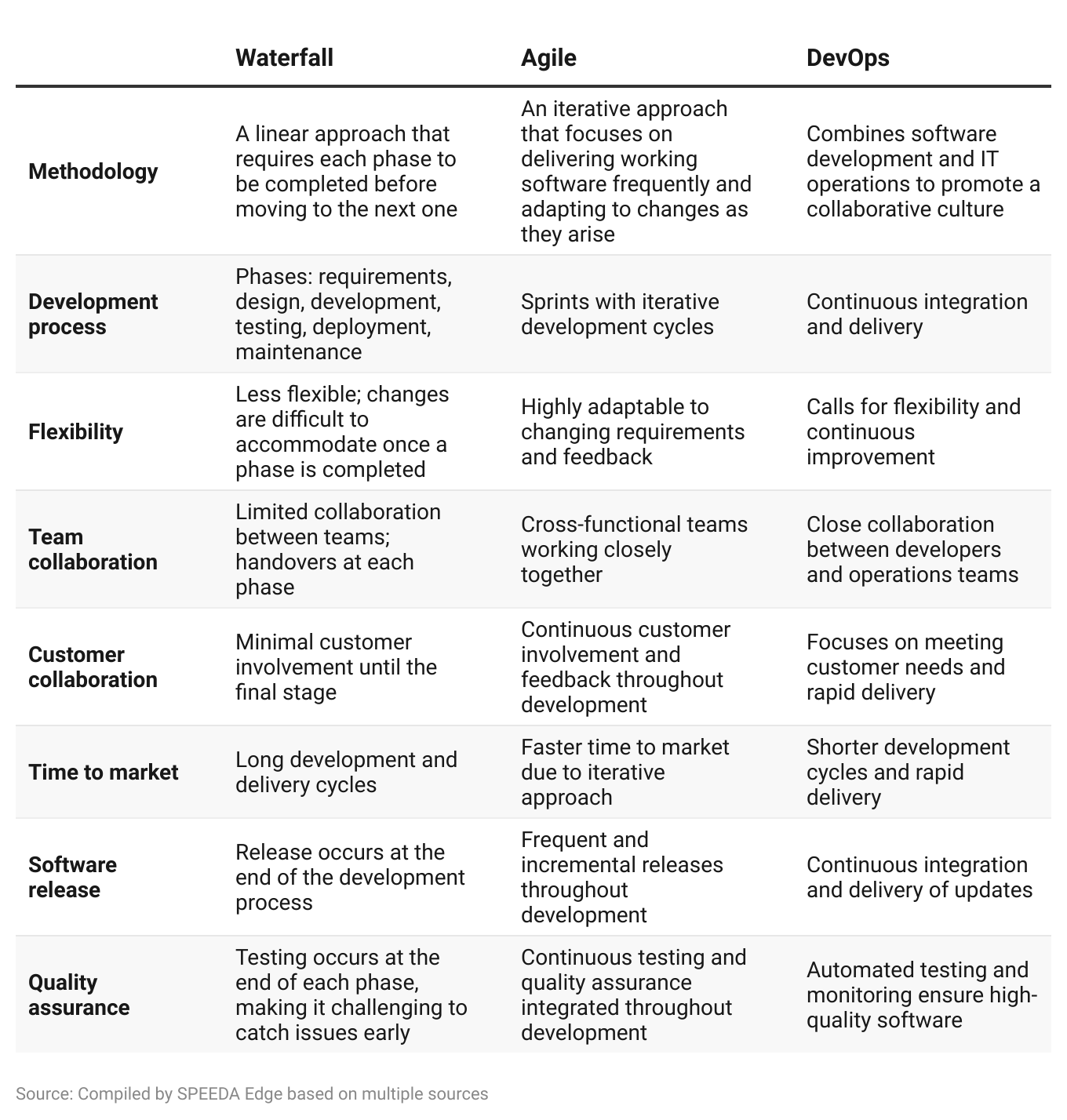
## **How is DevOps different from other software development approaches?**

Before DevOps came into play, the Waterfall and Agile methods were two common software development approaches. Waterfall followed a sequential and rigid process, characterized by lengthy development cycles and limited customer involvement until the final stage. In contrast, Agile prioritized iterative development, close collaboration, and frequent customer feedback, resulting in quicker time to market and improved risk management.

DevOps is an evolution of other development approaches and expands its scope beyond development to encompass the entire service life cycle. It promotes shorter development cycles and demands collaboration by incorporating operations engineers into development teams to help solve bottlenecks during software releases. Operations engineers are given the additional responsibility of handling engineering tasks in addition to deployment and monitoring.

#### 

#### **Key differences between DevOps and common software development approaches**



## **What does the DevOps toolchain look like?**

DevOps requires a collection of tools for each phase of the DevOps lifecycle, which makes up the DevOps toolchain. The toolchain can be clustered into five segments:

1. **Planning and analysis**: Platforms that help companies with software development workflows such as analyzing requirements, planning development schedules, and building roadmaps.
2. **Code editors and hosting platforms**: Desktop and web applications for writing code, which also offer features for intelligent auto-complete and code generation capabilities using AI. This segment also includes platforms that provide tools for version control and hosting code.
3. **Test and review**: Platforms to test and debug software, which also feature AI-based tools to automate testing workflows.
4. **Product deployment and feature management**: Platforms to automate software deployment into production environments hosted either on-premise or on the cloud, along with feature management capabilities that help developers control the availability of features in an application.
5. **Software monitoring and observability**: Platforms for developers to monitor running software and track processes, helping to identify crashes and errors and to resolve them.

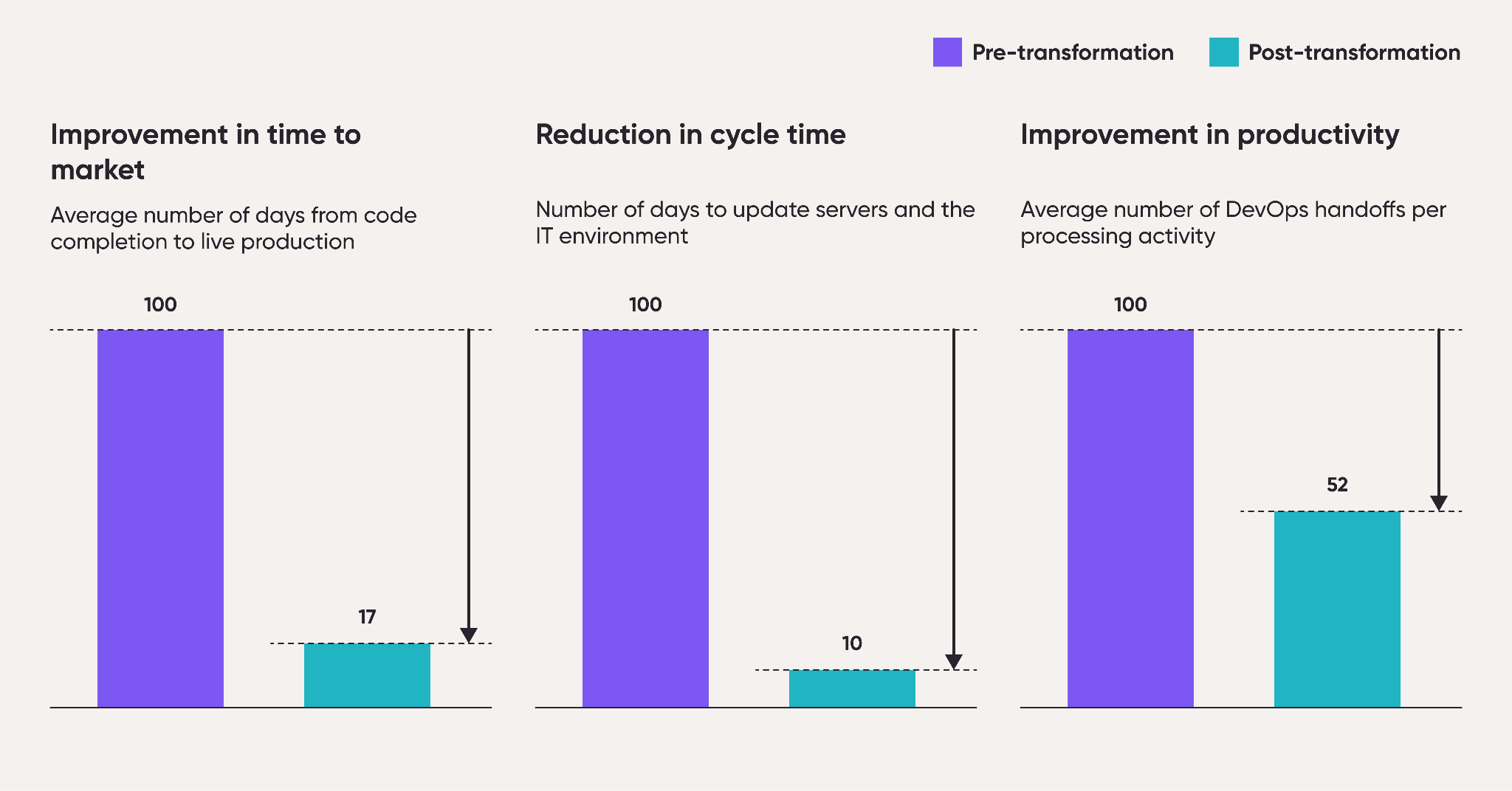
**However, we exclude the following areas when selecting companies for this industry:**

1. **Collaboration tools** that allow software development teams to work together on projects (covered under [Remote Work Tools](https://sp-edge.com/industry/10)).
2. Platforms that allow individuals to **create programs and applications** using functionality such as drag-and-drop **without requiring prior coding knowledge** (covered under [No-code Software](https://sp-edge.com/industry/14)).
3. Platforms that enable IT professionals as well as those with limited technical know-how to **create software programs with very little traditional computer programming** (covered under [Low-code Platforms](https://sp-edge.com/industry/95)).
4. Platforms that provide services such as automation, creation, deployment, integration, and scaling **for containers and containerized applications** (covered under [Next-gen Cloud Computing](https://sp-edge.com/industry/170))
5. Platforms that provide tools such as SDKs, blockchain node infrastructure, and bridges for developers to **create, host, debug, and add extra functionality to decentralized apps (dApps) on the blockchain** (covered under [Web3 Ecosystem](https://sp-edge.com/industry/144)).

## **What’s driving interest in DevOps?**

### **1. Desire to shorten software development cycles and accelerate delivery**

Organizations are increasingly prioritizing speed and the ability to deliver value quickly to their customers. DevOps is attractive because it enables developers to write software in small increments that can be integrated, tested, monitored, and deployed within hours. This allows development teams to be more responsive to the evolving needs of businesses while minimizing the risk of major disruptions caused by large releases with multiple changes. The ability to swiftly roll back a small change in case of any issues in production adds to the appeal of DevOps. A [McKinsey](https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/beyond-agile-reorganizing-it-for-faster-software-delivery#/) study revealed that companies using DevOps can significantly reduce the average time required for code development and deployment to 15 days from 89 days—a reduction of more than 80%.



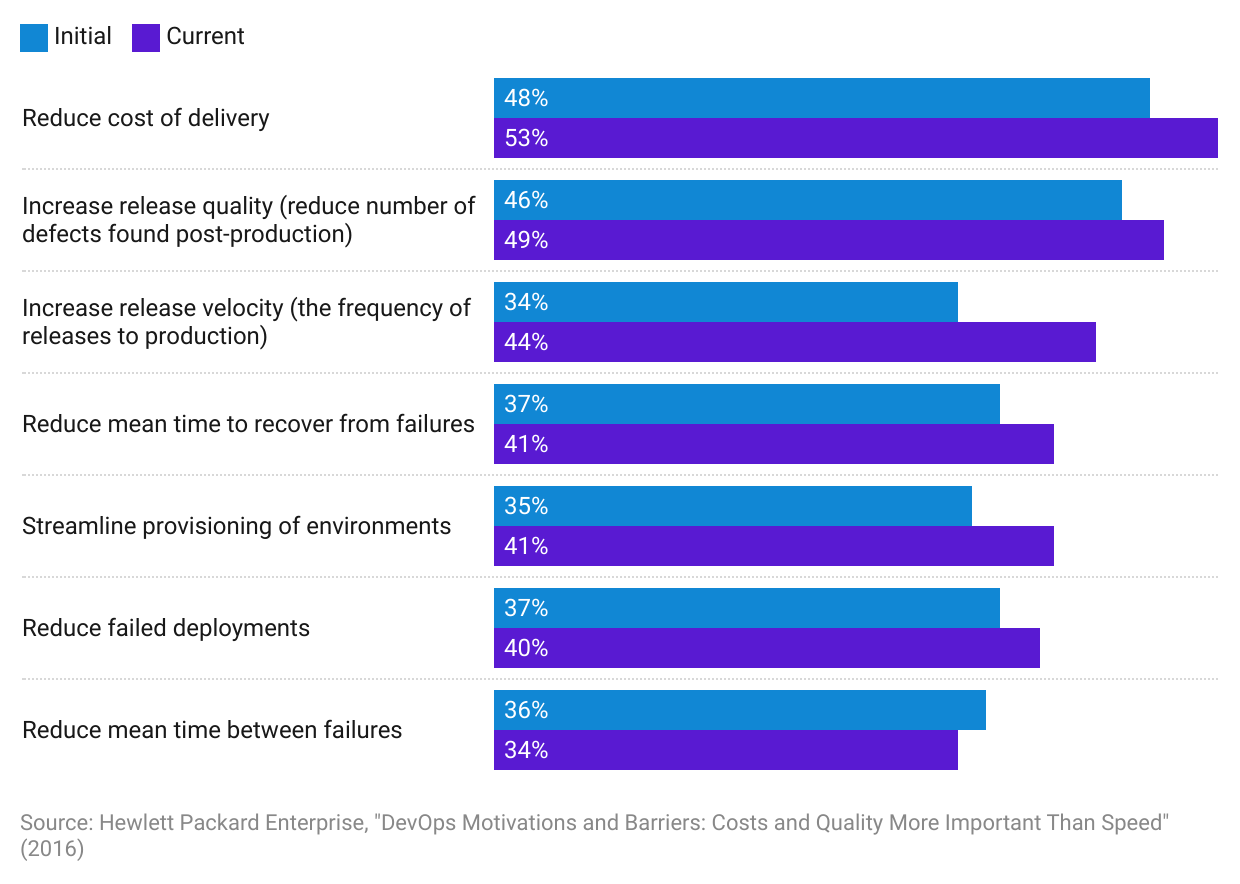
*Charts are indexed to 100*

Source: McKinsey, recreated by SPEEDA Edge

### **2. Potential cost savings and benefits from DevOps implementation are significant**

DevOps generates cost savings primarily through accelerated delivery and improved resource use. A [Hewlett Packard study](https://techbeacon.com/sites/default/files/gated_asset/devops-motivations-barriers.pdf) reported that 53% of enterprises cited cost reduction as the most significant factor for implementing DevOps. A drop in the number of defects in post-production and an increase in the frequency of releases to production were the next highly ranked factors.

#### **“Reducing cost of delivery” among top motivations behind implementation of DevOps**

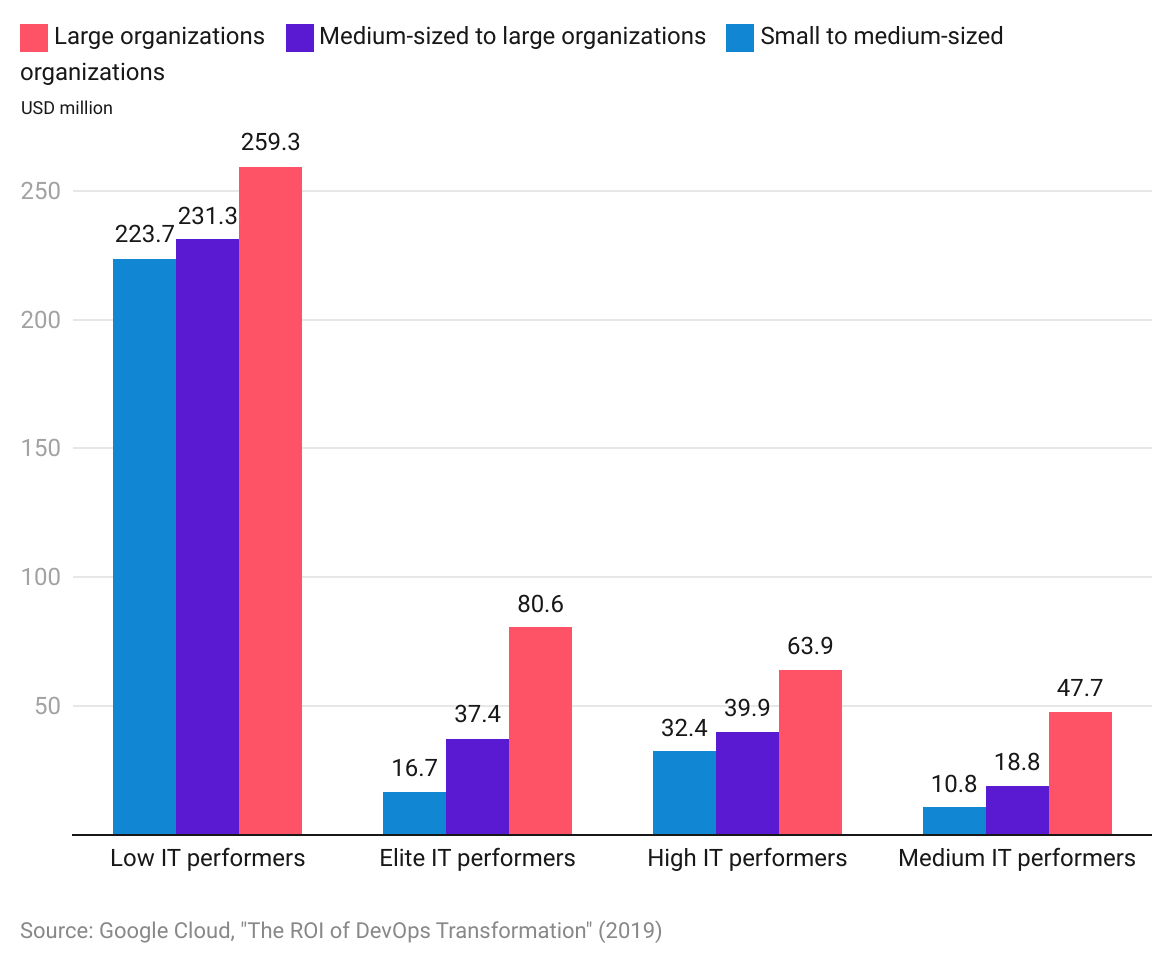


Moreover, a [Google Cloud whitepaper](https://services.google.com/fh/files/misc/whitepaper_roi_of_devops_transformation_2020_google_cloud.pdf) explored the benefits of DevOps through 1) improved efficiency via the reduction of unnecessary rework, 2) potential revenue gained by reinvesting the time saved in new offer capabilities, and 3) cost savings from downtime avoided. The study classified organizations as large (technical staff of 8,500), medium-sized to large (2,000), and small to medium-sized (250).

The findings from the study indicate substantial gains. For instance, for medium IT performers (those that are doing well in terms of stability, on par with the high performers, but fall behind in speed of delivery), the estimated total value of benefits was between USD 10.8 million (small to medium-sized organizations) and USD 47.7 million (large organizations). The returns are even greater for organizations with low IT performers (USD 223.7 million–259.3 million), as they have the most opportunities for improvement by addressing low-hanging fruit and setting measurable goals.

#### 

#### **Total value of benefits from DevOps ranges between USD 10.8 million and USD 259.3 million**

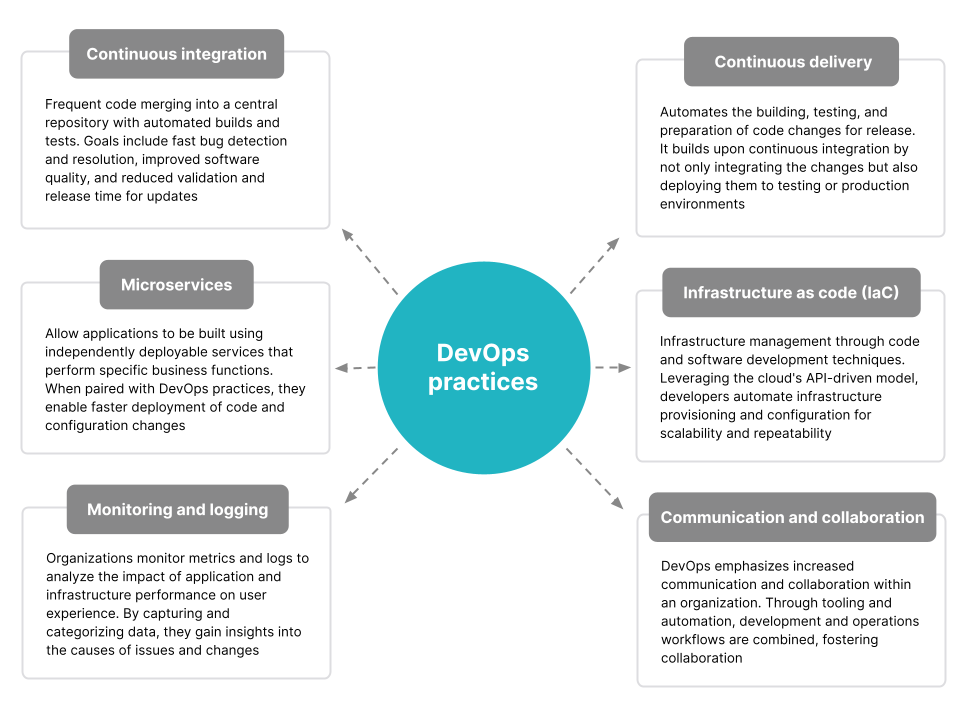


## **How does DevOps work?**

The DevOps approach is centered around processes that harness the power of microservices architecture by breaking down large, intricate systems into smaller, autonomous projects or components. Nevertheless, the mix of microservices and frequent releases can result in an increased number of deployments, potentially straining internal operating systems. To address this, DevOps employs specific practices such as continuous integration (CI) and continuous delivery (CD). CI involves the regular and automated integration of code changes into a shared repository, while CD focuses on automatically deploying approved code changes to the production environment once they have undergone building, testing, and approval.

The key difference between CI and CD lies in their specific areas of emphasis within the software development process. CI primarily focuses on the build and test stages, ensuring that code changes undergo thorough testing and integration into the codebase. On the other hand, CD places greater emphasis on the deployment stage by automating the process of deploying code changes to the production environment. The collaboration of CI and CD within the DevOps framework contributes to the efficient and dependable development and delivery of software.

To ensure the adaptability of computing resources, automation techniques such as infrastructure as code (IAC) and configuration management are also used. Also, monitoring and logging play critical roles in tracking the performance of applications and infrastructure, enabling the swift resolution of any issues.



Source: Compiled by SPEEDA Edge based on multiple sources

## **What is AI’s impact on DevOps?**

The growing interest in integrating AI/ML into DevOps practices arises from their ability to automate various tasks including code analysis, testing, deployment, monitoring, and maintenance. By leveraging data analysis from diverse sources like code repositories, logs, and user behavior patterns, AI/ML techniques enable proactive issue prediction and prevention, allowing teams to address potential problems before they arise. Furthermore, these technologies provide valuable insights into system performance, allowing teams to continuously refine their processes and achieve improved outcomes, and automate bug and vulnerability detection. Additionally, ML algorithms can analyze performance data to identify opportunities for optimization.

According to GitLab’s [2022 Global DevSecOps Survey](https://about.gitlab.com/blog/2022/09/15/why-ai-in-devops-is-here-to-stay/), there has been a significant rise in the adoption of AI/ML in DevOps practices. The survey findings indicate that 24% of respondents reported incorporating AI/ML into their DevOps processes, a doubling from 2021. To be specific, 37% of teams are using AI/ML for software testing, while 31% are applying it for code reviews.

#### **How AI impacts DevOps**



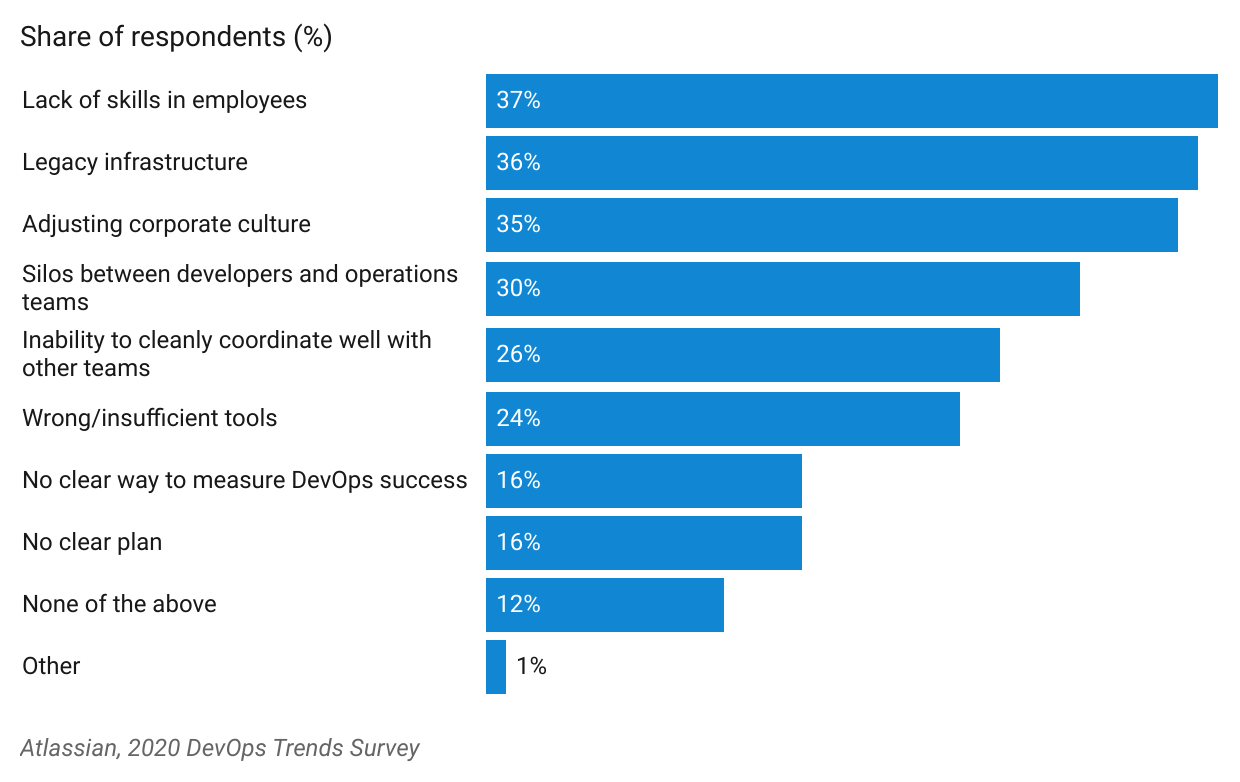
[Generative AI](https://sp-edge.com/industry/163), a subset of AI, has gained significant attention since the successful launch of [Open AI](https://sp-edge.com/insights/16079)’s [ChatGPT](https://sp-edge.com/insights/16084) in November 2022. There have been multiple generative AI-based product launches recently with regard to DevOps, and the most notable among them was Google’s announcement of [Duet AI for Google Cloud](https://cloud.google.com/blog/products/application-modernization/introducing-duet-ai-for-google-cloud) in May 2023. The product offers real-time code recommendations, generates complete functions and code blocks, and suggests fixes for vulnerabilities and errors. Google's sophisticated generative AI models, which have been trained using data from Google Cloud, including sample code and documentation, enable this capability by enhancing and refining behaviors and patterns. Startups such as [OpsVerse](https://www.businesswire.com/news/home/20230425006050/en/OpsVerse-Announces-Aiden-a-Generative-AI-Based-and-Secure-DevOps-Copilot) and [Kubiya](https://www.businesswire.com/news/home/20230417005008/en/ChatGPT-for-DevOps-Kubiya-introduces-Generative-AI-engine-for-DevOps-and-Platform-Engineering) also leveraged generative AI technology to launch DevOps assistants in April 2023, while [GitLab](https://sp-edge.com/companies/169199) and [Tabnine](https://sp-edge.com/companies/878414) also offer assistant tools based on the large language models (LLMs) created by OpenAI.

## **What are the risks to growth?**

### **1. Implementation and scalability issues**

While organizations acknowledge the advantages of implementing DevOps, there are certain apprehensions surrounding adoption. In an Atlassian [study](https://www.atlassian.com/whitepapers/devops-survey-2020), 84% of respondents revealed that they had faced barriers to their DevOps implementation, indicating that a lack of employee skills, legacy infrastructure, and corporate culture were the biggest hurdles.

#### **Lack of employee skills is the top-ranking barrier to DevOps implementation**



Moreover, in a [study](https://www.bain.com/insights/devops-tech-report-2021/) conducted by Bain & Company, approximately 90% of companies encountered challenges when attempting to scale DevOps practices. In addition to the aforementioned barriers to implementation, the complexity of managing various tools contributes significantly to these difficulties.

### **2. Lack of interoperability standards**

The absence of interoperability in DevOps arises from the freedom service providers have to choose diverse products and solutions from various open-source communities and vendors. Each company employs specific tools, technologies, and processes for their CI/CD pipelines, creating fragmentation and complicating the selection process for the DevOps automation team. This complexity requires a deep understanding of tools to choose the appropriate ones for specific use cases, often leading to workarounds for compatibility.

However, these workarounds can exacerbate interoperability issues when repeated across multiple companies. The lack of standardization presents challenges for users aiming to build and oversee comprehensive CI/CD pipelines. Establishing CI/CD pipelines that bridge the gap between open-source communities, vendors, and service providers becomes crucial for achieving efficient and secure software delivery.

### **3. Security concerns in cloud infrastructure**

A McKinsey study referenced in the [Azure Annual DevOps Report](https://www.arrow.com/ecs-media/13726/ms-azure-devops-report.pdf) found that only 17% of organizations conduct security testing specifically for major releases or when deploying to production, while only 46% of enterprises integrate security tools into their DevOps pipelines. Given that the DevOps toolchain is implemented in cloud environments, it presents unique security challenges alongside its numerous benefits. Unlike traditional on-premises software deployments, the cloud exposes a wider range of potential vulnerabilities. A minor misconfiguration or human error in the cloud environment can expose critical resources to public networks.

The emergence of [DevSecOps](https://www.ibm.com/topics/devsecops#:~:text=DevSecOps%E2%80%94short%20for%20development%2C%20security,%2C%20deployment%2C%20and%20software%20delivery.), which involves integrating security measures at every stage of the DevOps model, could prove helpful in addressing these concerns. DevSecOps promotes a "Shift Left" approach, focusing on ensuring application security from the earliest stages of the development life cycle.

## **What’s next?**

The next evolution of DevOps has already begun through the integration of AI. The use of AI to automate coding tasks is populating the industry, and this can be expected to spread to other segments, such as monitoring and observability, given the importance of real-time vulnerability and performance checks. Furthermore, the "Shift Left" approach is expected to bolster confidence in DevOps implementation through the addition of security across the development cycle. As adoption picks up, the market for DevOps is projected to grow at a CAGR of 19.7%, reaching [USD 25.5 billion](https://www.marketsandmarkets.com/Market-Reports/devops-824.html#:~:text=The%20size%20of%20the%20global,USD%2025.5%20billion%20by%202028.) by 2028, up from USD 10.4 billion in 2022.

Organizations weighing the perceived benefits and growth barriers of DevOps should understand that its implementation is not just establishing a toolchain and making gains overnight. Efforts to implement DevOps in one step are likely to result in disappointment, especially in large organizations that require more time and effort to initiate change. It is crucial to understand that the framework is unlikely to succeed without robust infrastructure and, most importantly, a collaborative environment—which often poses the greatest challenge in scaling DevOps. Therefore, as organizations search for speed and agility in software development, collaboration remains paramount.

©2024 Uzabase, Inc. All Rights Reserved. The information contained herein: (1) is proprietary to Uzabase Inc. and/or its content providers; (2) may not be copied or distributed; and (3) is not warranted to be accurate, complete or timely. Neither Uzabase Inc. nor its content providers are responsible for any damages or losses arising from any use of this information.