Gartner

Gartner for I&O and Software Leaders

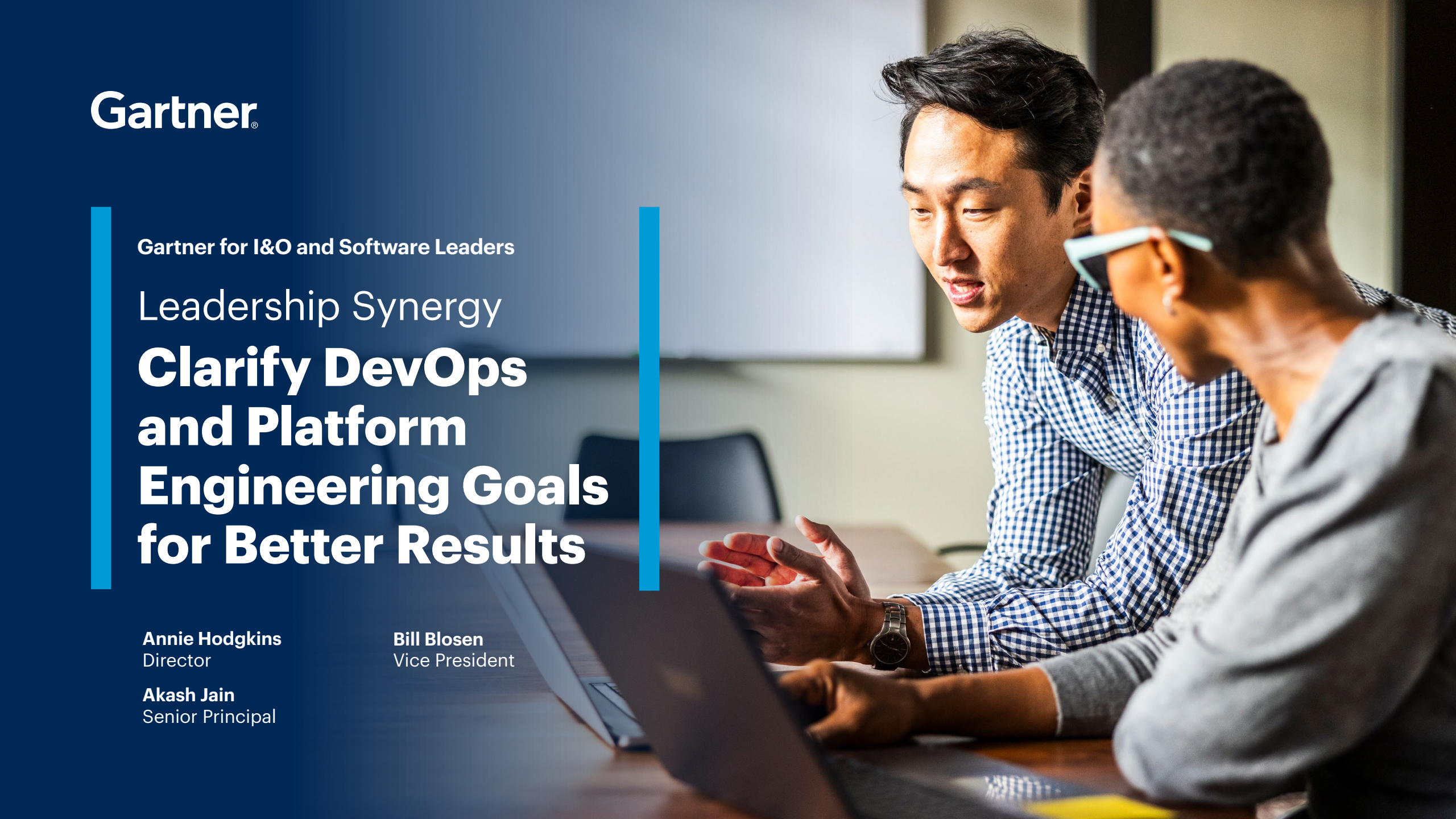
Leadership Synergy

Clarify DevOps and Platform Engineering Goals for Better Results

Annie Hodgkins
Director

Akash Jain
Senior Principal

Bill Blosen
Vice President



The purpose of DevOps and platform engineering in software engineering is frequently misunderstood, resulting in inefficient delivery of business outcomes, developer frustration and wasted investment. Software engineering leaders need to clarify the goals of each approach to maximize their impact.

Overview

Key findings

- DevOps, including DevSecOps, is a way of working and culture to achieve business goals and objectives. DevOps aligns the objectives of people and teams across a software product's life cycle, but many organizations treat DevOps as a label for a developer-toolchain-oriented team.
- Platform engineering is the discipline of building and operating self-service developer platforms for software development and delivery. It has gained a lot of momentum, but some organizations mistakenly launch platform engineering by relabeling an existing platform, infrastructure or developer tools team as a "platform engineering team."

Recommendations

- Create a common understanding of DevOps and platform engineering by aligning the goals of each approach to clear, shared business objectives.
- Improve developer experience and productivity by treating internal developer platforms as products and by strengthening product management skills and DevOps mindsets on platform engineering teams.

I&O leaders are partnering with software engineering leaders to break down siloes and improve knowledge sharing of modern software engineering skills to more effectively support business outcomes. Applying software engineering best practices to infrastructure platform teams leads to greater agility and faster product delivery through robust, scalable and secure platforms.

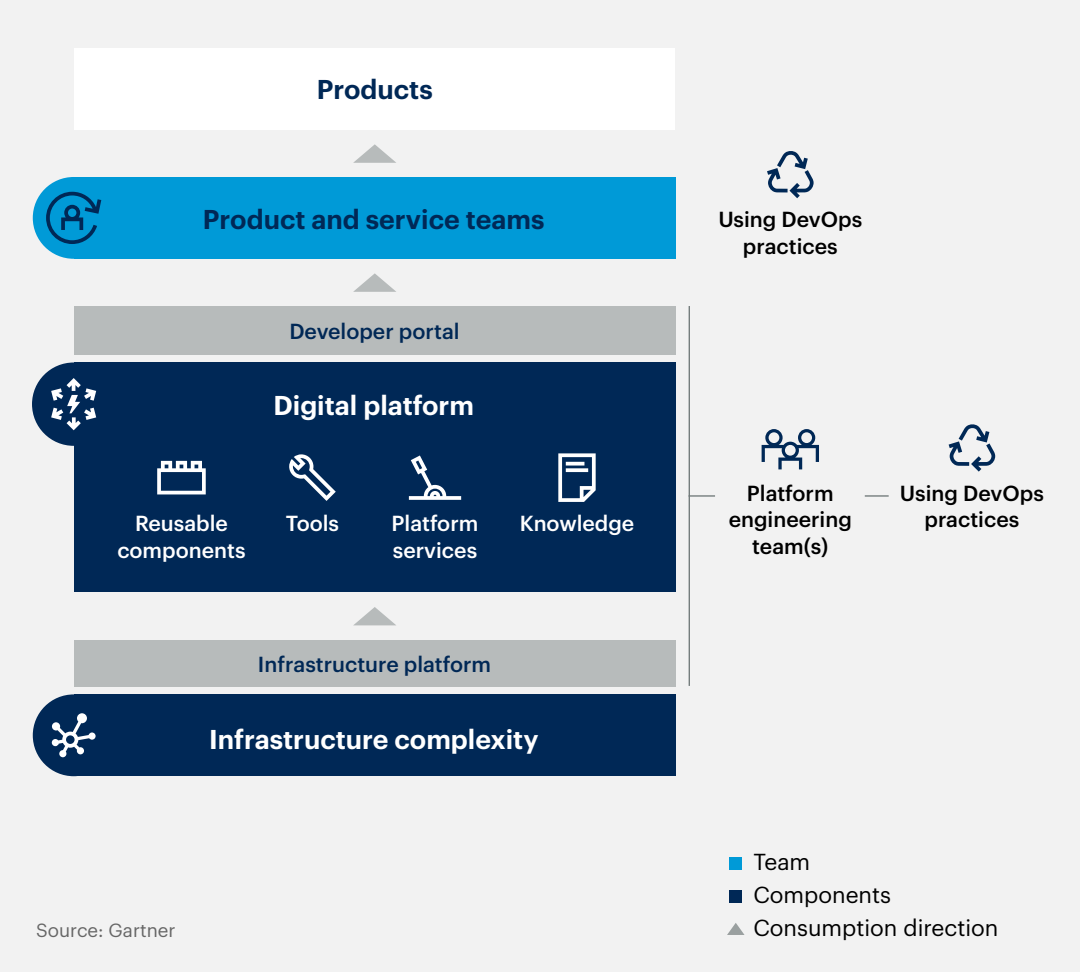
Introduction

Platform engineering has gained momentum as a discipline to improve the flow of business outcomes and reduce the cognitive load of software product teams working in a complex technology ecosystem. Many organizations, however, see platform engineering as another label for any platform or DevOps tools team.

Without a clear understanding of both DevOps and platform engineering, as well as defined goals for each, organizations will fail to achieve better product outcomes and improved developer experience. Instead, organizations will experience wasted investment in platforms that developers dislike, or worse, ignore, and increased developer frustration.

Software engineering leaders seeking to modernize practices and improve the value created by their product teams are asking for clarity in the definitions of DevOps and platform engineering and in how to bring this clarity to their organizations. Figure 1 illustrates that software product or service teams and platform engineering teams use DevOps practices to deliver customer value. These insights provide recommendations to overcome common misconceptions by clarifying the distinct business goals that each approach enables.

Figure 1: Clarify the distinction between DevOps and platform engineering



Source: Gartner

Analysis

Align the goals of DevOps and platform engineering to business objectives

There is a great misconception in the IT press and industry on the definition of DevOps. DevOps is not the team that manages and operates the toolchain, or a specific role, nor is it an objective or a goal.

Convey the business impact of DevOps principles

DevOps is a way of working to achieve business goals and objectives. It is a highly collaborative approach that focuses on the optimization of value delivery, cost and risk on behalf of a defined customer (see Keys to DevOps Success).

In addition, inquiry with Gartner clients has uncovered that many organizations are operating with a misguided understanding and implementation of DevOps.[1] This includes:

- Equating DevOps with tools. This is often a signal that there is a deep misunderstanding of the DevOps values in the first place, or a cultural resistance to change.
- Creating DevOps job titles and teams oriented around continuous integration/continuous delivery (CI/CD) or toolchain maintenance and support, much like a center of excellence (COE). Superficial changes to teams will not embrace the paradigm shift required for successful DevOps ways of working.
- Measuring product team success using output metrics such as lines of code or bugs fixed, rather than focusing on delivering customer outcomes and business value. Developers with a user- or customer-centered mindset are more productive and deliver better software product outcomes.[2] DevOps requires a focus on delivering business value, not technology-centric outcomes.

To overcome these misconceptions, software engineering leaders should anchor in a common definition and understanding of why DevOps is needed to achieve specific business outcomes. Will improving software development processes generate better customer outcomes? Do your initiative leaders value fast feedback loops and the ability to experiment? Software engineering leaders should only consider pursuing tools and technologies after defining how DevOps can improve business needs and outcomes. See the Gartner Magic Quadrant™ for DevOps Platforms for an overview of some of the tools used to support DevOps principles and common needs, like version control and CI/CD.

Reiterate the value of DevOps principles in software engineering townhalls and leadership communications to avoid falling back into the common misconceptions. Partner with HR or learning and development (L&D) teams to incorporate DevOps training in software engineering team onboarding. Convey how DevOps delivers on specific business needs by using examples and success stories, such as how:

- **Automation** can improve delivery speed and efficiency and reduce incidents and manual efforts for higher quality outcomes (see Automate the Application Delivery Value Stream).

- **Continuous delivery** allows for more rapid delivery of reliable releases.
- **Multidisciplinary teams** reduce time-consuming handoffs between siloes to speed up value delivery and solve complex problems faster.
- **Continuous integration** ensures the production readiness of an application for faster time to market and minimizes large-scale risk of failure.

Evangelize the overall goal of platform engineering

Platform engineering is the discipline of building and operating self-service developer platforms for software development and delivery. Platform engineering teams can and should apply DevOps principles to their own work and mindset.

Gartner analysts have heard various misconceptions of platform engineering adoption, including:[3]

- Retitling an existing DevOps toolchain team to become the new platform engineering team. This involves only superficial changes to the team's way of working. As a result, the team often does not have the right mindset or mix of skills to manage and evolve the platform as a product.
- Operating platform engineering teams with a "build it and they will come" mentality and not focusing on the usability of the platform capabilities. Overengineering the platform, without continuous platform consumer feedback, can lead to wasted efforts and misaligned platform capabilities.
- Making the platform usage mandatory with no options to use other tools. This common pattern undermines the need for the platform engineering team to make their platforms compelling and easy to use.

To overcome these misconceptions, proactively educate engineering teams and key stakeholders across business and technology groups about the overall goal of platform engineering: To improve developer experience and productivity, while reducing cognitive load and attrition. Use this message to clarify the difference between DevOps principles and platform engineering team goals and skill set needs (see How to Start and Scale Your Platform Engineering Team). Leverage a variety of channels and forums to reiterate why the goal of platform engineering matters to your organization by using examples such as how:

- **Paved roads** reduce the cognitive load and improve the experience of developers on software product teams.
- **Dedicated platform teams** combine multidisciplinary skill sets to build, curate, measure and evolve internal platforms to continuously improve product team pain points across the organization.
- **Embedded guardrails** can reduce the effort to deliver security, architecture, compliance and quality outcomes at scale.
- **Self-service** access to information, assets and capabilities can reduce tedious dependencies and improve the speed and predictability of software delivery.

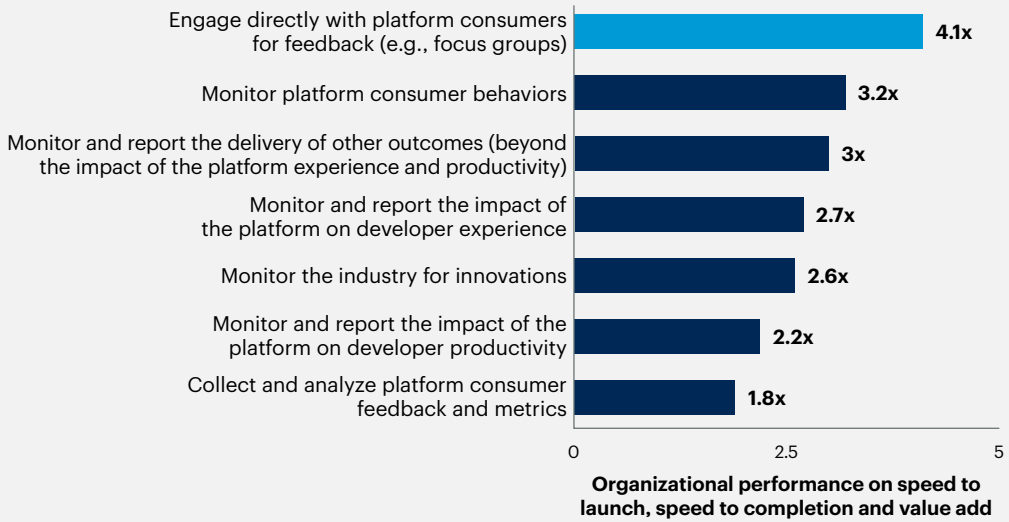
Treat internal development platforms as products

Internal development platforms focus on improving developer and business outcomes over producing technology inputs. The platform engineering teams that support these internal development platforms can and should apply DevOps principles to their own work to understand the needs and challenges of their active and potential platform consumers. To embrace the DevOps mindset of collaboration and customer-centricity, platform engineering teams should strengthen product management skills.

Software engineering leaders should first emphasize that what sets the high-performing platform engineering teams apart from the relatively lower performing teams is not the role composition of the team — but the activities for which they are accountable. In particular, platform engineering teams that are accountable for activities such as engaging directly with platform consumers for feedback on the platform capabilities, ease of use and satisfaction see 4.1 times better delivery outcomes.[3] (See How to Equip the Product Owner to Drive Platform Engineering Success.) This includes delivery outcomes such as speed to launch, speed to completion and value generation (see Figure 2).

Software engineering leaders should use this data to steer platform engineering teams to the product management activities that drive the highest value and demonstrate tangible ways of embracing the DevOps mindset and ways of working.

Figure 2: Platform engineering team activities to drive higher performance of delivery outcomes



n = 238; platform consumers and producers, excluding unsure
Q. Dependent variable: How would you characterize your organization's overall performance on the following activities relative to its peers or competitors? Ahead/far ahead of average score on speed to launch, speed to completion and getting value.
Q. Independent variable: To the best of your knowledge, are the individuals or teams that provide platforms typically responsible for any of these activities? Always responsible.
Source: 2023 Gartner Platform Teams and Platform Engineering Survey
Note: Three activities that were found not to increase organization performance are not shown.

Software engineering leaders should guide platform engineering teams to understand and pursue these activities to surface software product team pain points, preferences and needs. The 2023 Gartner Platform Teams and Platform Engineering Survey found that platform teams that delivered an easy-to-use, self-service platform saw significant improvements in developer experience and productivity.[4]

Work with platform engineering teams to build continuous feedback loops and uncover instances of high cognitive load that software engineers experience in their development tools and processes. Highlight teams that are successful in aligning iterative improvements to enhance those activities.

For example, 60% of survey respondents from the 2024 State of DevOps Report believe that the top benefit of platform engineering on software product outcomes is built-in security compliance.[5] This is important to note, as nearly two-thirds of software engineering leaders report that application security skills are an organizational pain point.[6]

Software engineering leaders can also use the Gartner Developer Experience Assessment to source comprehensive and deep insights into staff perceptions of their developer experience to identify your organization's strengths and high-impact improvement opportunities. These insights can help validate the priorities of your platform engineering team and demonstrate leadership support of the product management mindset.

Evidence

[1] Between 1 January 2024 and 4 December 2024, Gartner analysts had 4,490 interactions with clients on the topic of DevOps.

[2] Accelerate State of DevOps 2024, DORA.

[3] Between 1 January 2024 and 4 December 2024, Gartner analysts had 1,961 interactions with clients on the topic of platform engineering.

[4] **2023 Gartner Platform Teams and Platform Engineering Survey.** This survey was conducted online from 9 May through 27 June 2023 to gather benchmarking data on platform engineering, to analyze the effectiveness of platform engineering in organizations, and to explore the factors and practices that enable or inhibit platform effectiveness in organizations. In total, 238 respondents from organizations across the U.S. participated from a range of industries (excluding IT software and public sector). Qualified participants were at organizations with at least $500 million — with 83% over $1 billion in revenue. Respondents were very involved in self-service developer platforms in their organizations as either a producer, a consumer or both. Respondents included practitioner-level leaders in software engineering, IT, infrastructure and/or IT operations, data and analytics, and cybersecurity. Disclaimer: Results of this survey do not represent global findings or the market as a whole, but reflect the sentiments of the respondents and companies surveyed.

[5] 2024 State of DevOps Report: Platform Engineering Edition, Perforce Software, Puppet.

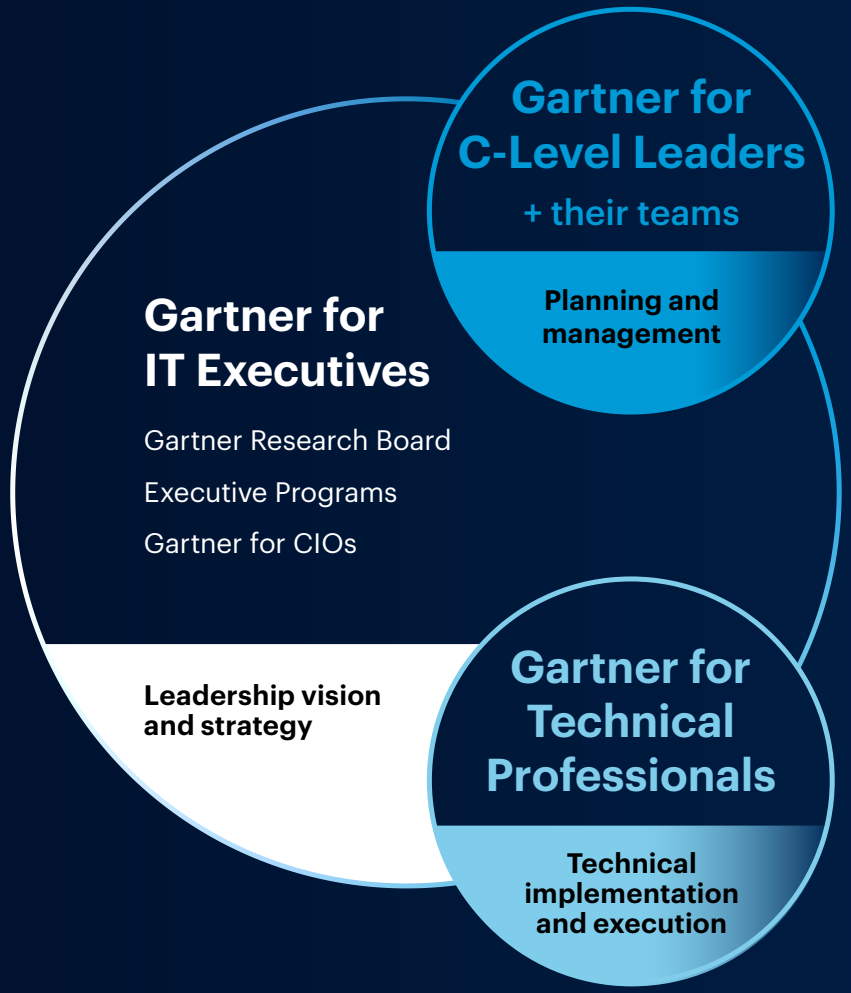
[6] **Gartner Software Engineering Survey for 2024.** This survey was conducted to identify the most important roles and skills for software engineering leaders and the change in their demand and importance since last year, understand how talent is sourced generally and for acquiring necessary artificial intelligence (AI)/machine learning (ML) skills, and what tools are seen to increase developer productivity and the metrics used to measure them. It also examines how software engineering leaders anticipate change in their operating budgets and the cost management steps taken. It further aims to identify the quality and testing techniques and programming languages software engineering leaders currently use and/or plan to use; their frequency of usage of UX design, user research and AI in generating components of user experience; and its impact on user satisfaction, accessibility and usability. It also intends to understand the software engineering leaders' responsibilities they find most difficult, the career paths available for senior-level individual contributors and how they are set up, how organizations attract and retain top performers in those career paths, and what management training is offered to staff. The survey was conducted online from October through December 2023 among 300 respondents from the U.S. (n = 241) and U.K. (n = 59). Qualifying organizations operated in multiple industries (excluding the IT software industry and education sector) and reported enterprisewide revenue for fiscal year 2022 of at least $250 million or equivalent, with 63% over $1 billion in revenue. Qualified participants were highly involved in managing software engineering/application development teams and the activities they perform. Disclaimer: Results of this study do not represent global findings or the market as a whole but reflect sentiment of the respondents and companies surveyed.

360-degree coverage for CIOs and their teams

Expert guidance and tools to maximize success at every level

Validate your vision and empower teams to execute with confidence.

CIOs rely on Gartner for Technical Professionals (GTP) to kickstart their most crucial projects. Instead of starting from scratch, they tap into expert resources to help accelerate their teams' performance. GTP offers purpose-built tools, proven technical plans, assessments, models, step-by-step guides, reference architectures and roadmaps — all designed to help you achieve better business outcomes when you engage as a team.



CISO/Security & Risk Leaders
CDAO/Data & Analytics Leaders
Software Engineering Leaders
Head of Infrastructure & IT Operations
Enterprise IT Leaders

Data & AI Architects
Infrastructure & Cloud Architects
Software & Applications Architects
Security Architects

Get access to technical experts and role-specific peer communities with GTP.

Gartner

Actionable, objective insights

Position your organization for success. Explore these additional complimentary resources and tools for I&O and software leaders:

Tool

Gartner AI Use Case Insights for IT Leaders

Discover, evaluate and prioritize AI opportunities to accelerate IT transformation.

Access Now

eBook

Building and Implementing Modern Application Architecture and Technologies

Create a compelling business case to obtain funding and develop a migration plan.

Download Now

Roadmap

Design a Resilient IT Infrastructure Strategy

Avoid common challenges when modernizing your infrastructure.

Download Now

Insights

How Platform Engineering Teams Can Augment DevOps With AI

Improve developer experience and optimize the software delivery infrastructure.

Download Now

Already a client?
Get access to even more resources in your client portal. Log In

Connect with us

Get actionable, objective business and technology insights that drive smarter decisions and stronger performance on your mission-critical priorities. Contact us to become a client:

U.S.: 1 855 811 7593

International: +44 (0) 3330 607 044

Become a Client

Learn more about Gartner for I&O and Software Leaders
gartner.com/en/information-technology

Stay connected to the latest insights

Gartner