FORRESTER®

Fostering High-Quality Embedded Software Development

Enhance Your Organization's Platform Engineering Strategy

A FORRESTER CONSULTING THOUGHT LEADERSHIP PAPER COMMISSIONED BY QT GROUP, AUGUST 2024

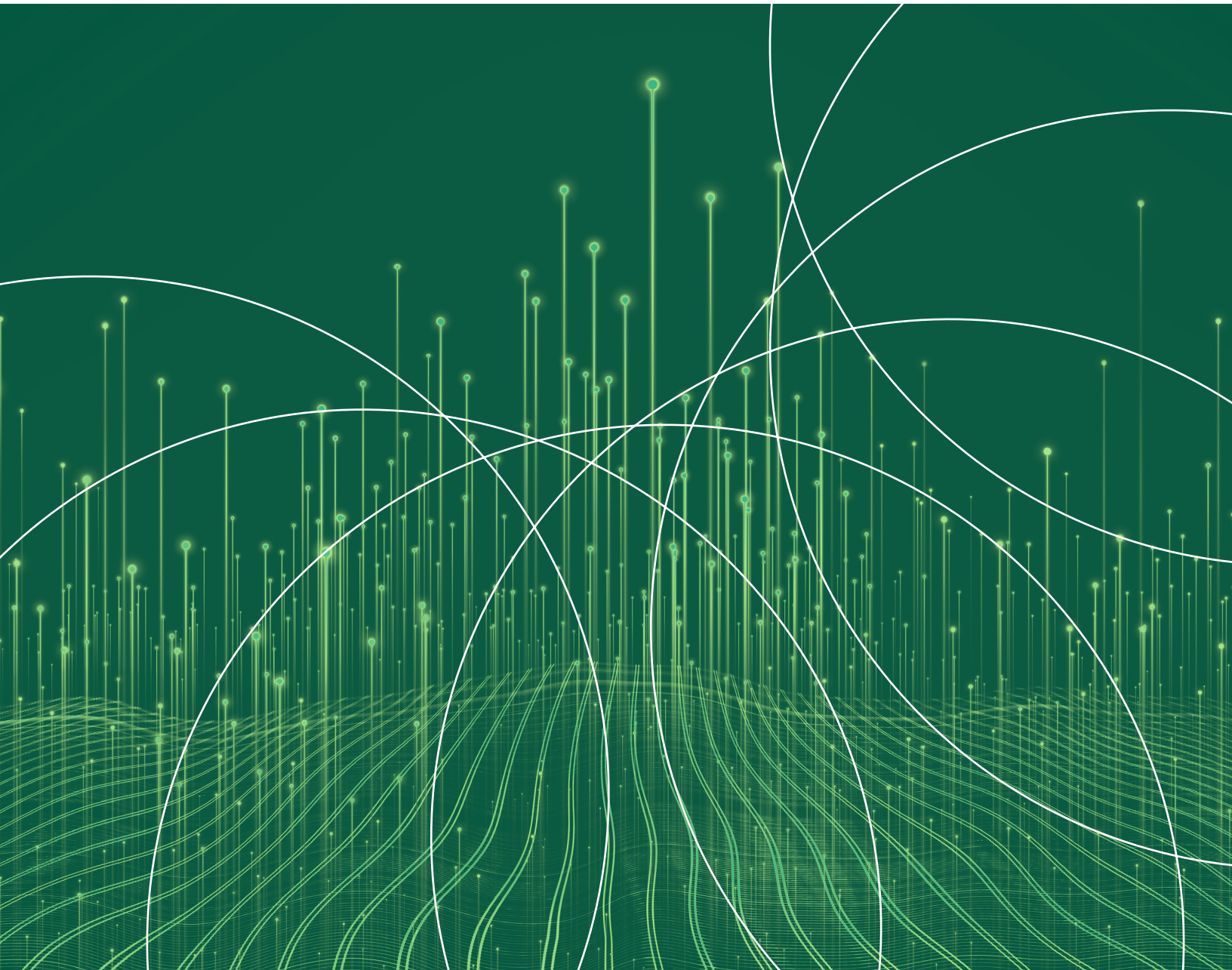


Table Of Contents

Project Team:

Diane Deng, Senior Market Impact Consultant

Alexis Ouelhadj, Market Impact Consultant

Tarun Avasthy, Senior Market Impact Consultant

Contributing Research:

Forrester's Technology Architecture & Delivery research group

Executive Summary

Traditional techniques of deploying and managing technology do not work for today's fast-paced business development and delivery tempo. The business shift requires modern technology teams to become more adaptive, creative, and resilient. Platform engineering is one of the building blocks of such modern development architecture. It promises to revolutionize software development by enhancing efficiency and cultivating innovation. However, many organizations struggle to fully realize these benefits due to the complexities of adapting software across diverse devices and operating systems and because embedded development still requires a high degree of custom-made solutions.

In April 2024, Qt Group commissioned Forrester Consulting to evaluate the adoption of engineering platforms for embedded software development. Forrester conducted an online survey with 317 decision-makers and influencers responsible for the platform engineering strategy at their organizations to explore the current state of platform engineering in the embedded software sector — the type of software that monitors or controls machines, such as consumer appliances, medical devices, industrial machinery, or vehicles — identifying key challenges and opportunities for improvement.

Key challenges include navigating diverse device environments, facilitating cross-functional collaboration, and managing computing resource constraints. Current platform strategy often fails to meet critical needs, particularly in diverse and highly regulated environments. Despite these hurdles, its potential to streamline development processes and enhance software quality at scale remains vast.

By prioritizing quality, safety, and regulation compliance, enhancing cross-functional collaboration, and fostering a thriving ecosystem and talent community, companies can unlock the full potential of platform engineering.

Key Findings

Embedded teams leverage platform engineering for an efficient, high-quality development workflow. In the highly regulated embedded software domain, prioritizing quality, security, safety, and compliance often reduces efficiency. Over half of respondents saw platform engineering as a way to meet high-quality standards more efficiently. By leveraging high-quality components and streamlined workflows, product teams can deliver better-quality software more rapidly and at a lower cost.

Existing platforms are not yet meeting expectations. Most respondents mentioned their organization's embedded software is still created with custom, project-based development. The need to create bespoke solutions underlines the low level of standardization and automation achieved so far, exposing developers to several key challenges.

Embedded development teams need to address talent shortages and tool integration challenges. Talent shortages are a challenge in advancing and promoting an effective platform strategy for most respondents. Yet 52% saw integration with specialized tools as a key capability for successful platform engineering. These preferences underscore the need for specialized talent and the importance of dedicated tools in transitioning to, maintaining, and fully realizing benefits of platform engineering strategies.

Advancing a platform strategy requires balancing standardization and flexibility. Platforms need to harmonize processes without hindering productivity. Nearly half of the respondents highlighted the difficulty of maintaining the balance between standardization and flexibility — between established golden paths and the ability to customize for unique cases.

The Evolution And Importance Of Platform Engineering

Approaches to software development have evolved significantly over the past few decades. Traditionally, software development has been primarily focused on custom project-based solutions tailored to specific use cases. This approach, though it has served its purpose, has been time-consuming and resource-intensive, often leading to fragmented and inefficient development processes.

With the advent of a more connected world and the proliferation of connected devices, software developers now juggle multiple types of tasks, such as feature developments, development standards, compliances, and safety. The need for more streamlined and efficient development processes has grown, and platform engineering has emerged as a response to these challenges. Platform engineering provides a structured approach to software development that significantly reduces the need to duplicate the effort for nondifferentiating but fundamental tasks, such as security, compliance, and deployment.

This study defines a software platform engineering strategy (platform strategy or platform engineering, for short) as a cohesive approach to software development with standardized tools, templates, APIs, and best practices. The goal is to enable rapid and effective innovation by providing a reliable, consistent development environment. This strategy focuses on streamlining the development process, reducing complexity, and enhancing productivity through a well-designed and maintained platform.

In the current technological landscape, a platform strategy is more relevant than ever due to several key factors:

- **Rapid technological advancements.** The pace of technological change has accelerated, necessitating quicker development cycles and more adaptable solutions. A platform strategy enables organizations to respond swiftly to these changes by providing a flexible and scalable development framework.
- **Increasing complexity of embedded systems.** The high complexity of modern embedded systems stems from the wide variety of integrated functionalities and the need to interact with diverse hardware and software environments. Platforms help reduce toil by taking care of infrastructure work — like security, compliance, and deployment — in a consistent, systematic, and (semi-) automated way.
- **Demand for interoperability.** As the internet of things (IoT) continues to grow, the need for interoperability between devices from different manufacturers has become critical. Platform engineering facilitates this by promoting the use of standard protocols, enabling seamless communication and integration.
- **Stringent regulatory requirements.** Industries such as automotive, healthcare, and consumer electronics face strict requirements related to safety, security, and regulatory compliance. A platform strategy supports adherence to these regulations through processes, tools, and code with best practices baked in. It ensures that compliance is active from the ground up.

> A platform strategy supports adherence to regulations through processes, tools, and code with best practices baked in.

Benefits Of Platform Engineering Become Appealing To Embedded Developers

Embedded software operates essential machines like vehicles, medical devices, and industrial machinery. Because of this, embedded development faces unique challenges with respect to quality and safety compared to other types of software. These challenges include stringent regulatory standards, high usability requirements, integration with other devices, interoperability, and connectivity. Errors and poor usability in embedded software can have severe safety consequences, unlike software bugs in typical office software that might only cause inconveniences.

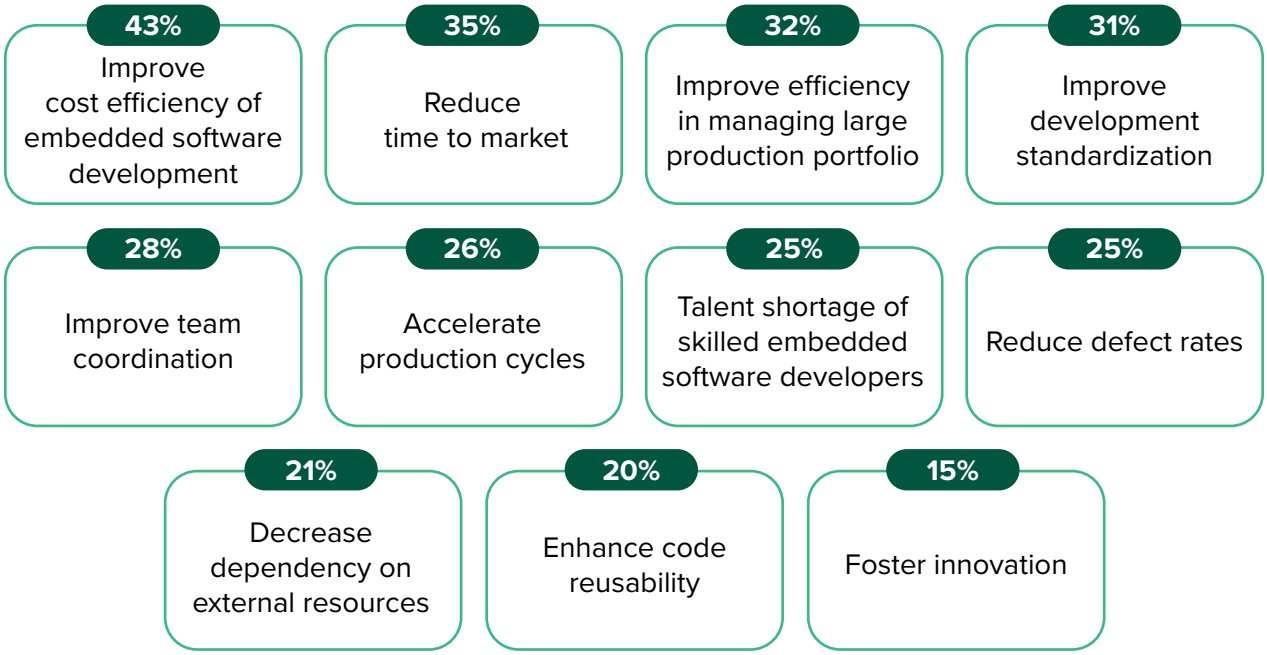
Survey respondents revealed their drivers for pursuing a platform strategy for embedded software included (see Figure 1):

- **Improving cost efficiency.** Forty-three percent of respondents cited improving cost efficiency as the main driver towards platform engineering. By leveraging a platform of reusable components that offers self-service and automation capabilities, enterprises reduce duplicate efforts and foster consistency. Cost efficiency is a strong driver towards platform engineering especially in the highly regulated medical device industry.
- **Improving time to market.** Standardized and reusable components streamline development cycles and boost efficiency, leading to quicker market entry and enhanced security. The availability of high-quality, ready-made components helps speed up the process. In this survey, 35% of respondents identified reduced time to market as the main driver for adopting platform engineering, along with reduced defect rates (25%) and enhanced code reusability (20%).
- **Streamlining portfolio management and team coordination.** By providing ready-made libraries, efficient interfaces, and streamlined workflows across teams, platform engineering boosts efficiency in managing large production portfolios (32%) and enhances team coordination (28%), ensuring seamless project execution. For example, streamlined communication channels and unified data formats reduce integration issues and enhance overall system performance.

Safety and security are persistent concerns for respondents in our study. Platform engineering helps mitigate risks in this area by standardizing and automating processes, thereby establishing a secure foundation for product teams. Platform teams can also advocate security measures, encouraging other teams to follow suit.

FIGURE 1

"What are the top drivers that led you to pursue a platform engineering strategy for your embedded software development?"



Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

EFFICIENCY IS THE KEY OPERATIONAL BENEFIT OF PLATFORM STRATEGY

As the embedded software industry navigates the complexities of modern development, executives are increasingly acknowledging the value of building and maintaining a platform strategy. Notably, 93% of

respondents agreed that their organization's platform engineering strategy is backed by leadership and deemed critical to the business's future. Additionally, 92% confirmed that their platform is designed for continual improvement with a dedicated budget, ensuring ongoing advancements and high-quality outcomes.

By standardizing development processes through the use of libraries, templates, APIs, and best practices, platform engineering minimizes variability and reduces complexity in development. Developers can better direct their expertise to value-added tasks and avoid using disparate tools. This not only ensures wider consistency and higher software quality, but it also allows businesses to scale without a proportional increase in software engineers, addressing the persistent issue of talent shortages.

The survey respondents revealed their organizations have experienced or expect to experience several operational benefits (see Figure 2):

- **Improved development efficiency.** This was cited by 59% of respondents as the most significant benefit. By streamlining processes and providing out-of-the-box, high-quality components, platform engineering reduces development time and increases overall productivity.
- **Reduced maintenance cost.** Nearly half of the respondents (49%) saw a reduction in maintenance costs via platform use. More specifically, the top three benefits related to software maintenance include simplifying updates across products (45%), predictive maintenance and fault detection (42%), and performance monitoring (39%).
- **Improved DevOps practices.** Just under half (45%) of respondents reported that platform engineering contributes to improved DevOps practices. This improvement fosters better integration, continuous delivery, and more efficient workflows, enhancing the overall development lifecycle.

> The dual benefits of security and compliance ensure that software not only meets regulatory requirements but also maintains robust security measures, protecting both the company and its customers.

FIGURE 2

"What are the top benefits your embedded software development teams experienced/expect to experience with the adoption of a platform engineering strategy?"

59% Improved development efficiency

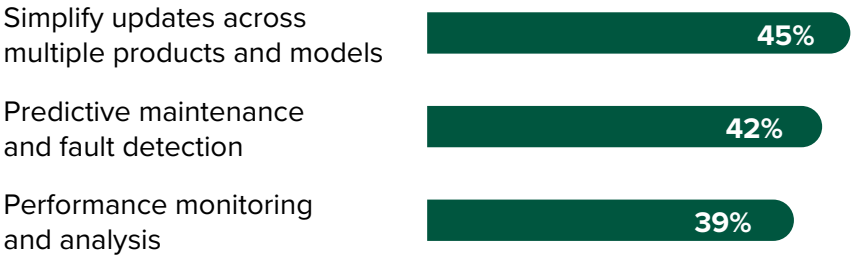
49% Reduced maintenance cost

45% Improved DevOps practices

40% Improved embedded software ownership

39% Improved resource utilization

"How does the engineering platform help with software maintenance?"



Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Note: Showing top responses
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

QUALITY UNDERPINS ALL BUSINESS BENEFITS

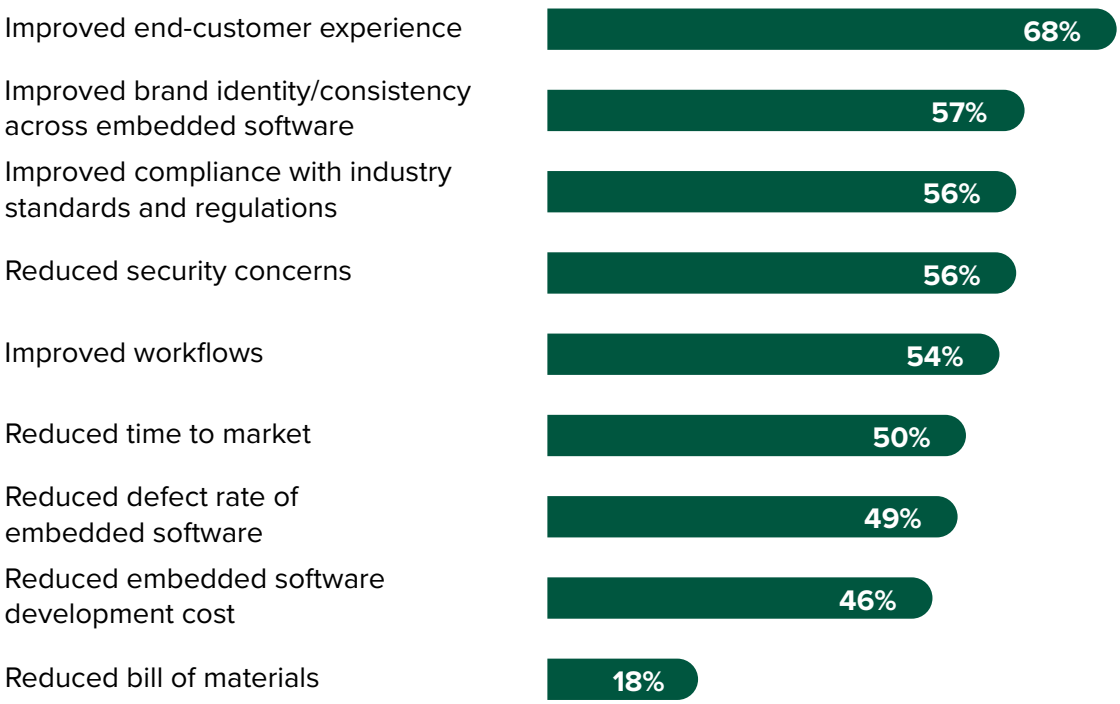
Businesses adopting a platform engineering strategy experience several key benefits, which significantly impact both operational processes and customer satisfaction. The survey data highlighted the following top business benefits (see Figure 3):

- **Improved end-customer experience.** Platform engineering encapsulates best practices and provides high-quality components. Respondents reported that their platform strategy provides greater customer satisfaction to the end customers (68%) as developers can deliver higher-quality, more reliable software.

- **Improved brand identity and workflow efficiency.** Respondents agreed that a standardized platform approach ensures consistent quality and branding (56%). This led to streamlined workflows that enhanced their organizations' ability to maintain a consistent look and feel across models and products and reduced the time and effort spent on customization (54%).
- **Enhanced compliance and security.** More than half (56%) of respondents reported that platform engineering reduces security concerns and improves compliance with industry standards and regulations. This dual benefit ensures that software not only meets regulatory requirements but also maintains robust security measures, protecting both the company and its customers.

FIGURE 3

"What are the top benefits your business experienced/expect to experience with the adoption of a platform engineering strategy?"



Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

Some respondents were able to quantify the business benefits resulting from adopting a platform strategy in the last 12 months. These benefits included:

- **Shorter time to market.** Platform engineering improves the availability of a streamlined process, self-service, and automation capabilities. Out of the 50% respondents who have managed to shorten their time to market with platform engineering, 93% experienced a reduction of at least 6% in their time to market, while over 55% saw a reduction of over 11%.
- **Reduced defect rate.** Platform engineering strongly enhances software quality by providing ready-made, reusable components and well-defined integration, testing, and deployment processes. Out of the 49% of respondents who have experienced a reduction in defect rate, over 38% saw a high defect reduction (11% or more). The higher quality is particularly noticed in the medical and automotive sectors.
- **Reduced embedded software development cost.** While a smaller team of highly skilled engineers maintains the platform, the platform's self-service capabilities for end products make it easier for less experienced (and less expensive) developers to build and ship high-quality products with greater efficiency. Out of the 46% who have managed to reduce embedded software development cost, 47% experienced a high cost savings (11% or more).

By adopting a platform strategy, embedded software teams not only anticipate substantial improvements but also experience tangible benefits. This strategic approach drives efficiency, brand consistency, customer satisfaction, and quantified cost reduction, positioning companies for long-term success in the competitive modern market.

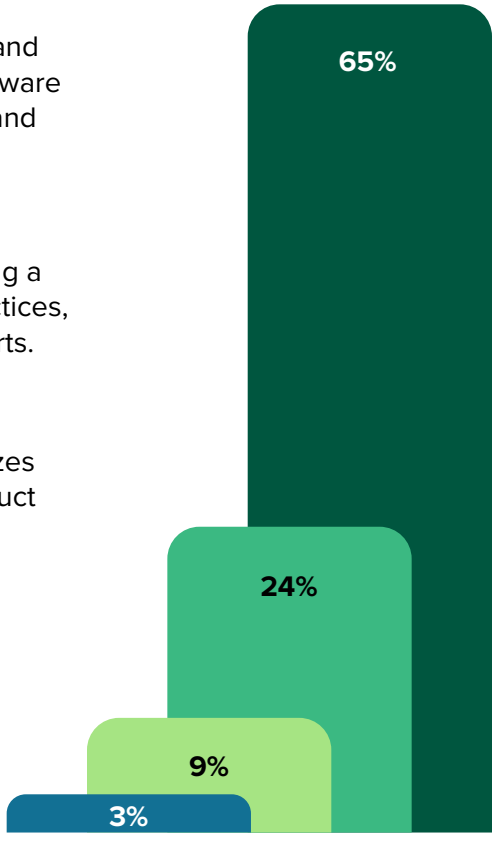
Existing Platform Strategies Fall Short Of Expectations

Many organizations believe their platform strategies are mature. Two-thirds of respondents (65%) saw their platform as the foundation upon which embedded software is built, with ongoing efforts to enhance its capabilities and increase automation and use case coverage (see Figure 4).

FIGURE 4

"Which of the following best describes the current status of your organization's platform engineering strategy?"

- Operational — Our engineering platform is operational and serves as the foundation upon which all embedded software is built, with ongoing efforts to enhance its capabilities and reach maturity.
- Resource hub creation — We are focused on establishing a central repository of technical documentation, best practices, and reusable components to support development efforts.
- Foundational development — Our organization recognizes platform engineering as the cornerstone for future product development, and we are in the process of building or planning this foundation.
- Exploratory phase — We're in the early stages of understanding what a platform engineering strategy could look like for our organization.



Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Note: Showing top responses; percentages may not total due to rounding.
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

However, our study also indicates a significant gap between perception and reality (see Figure 5). These platforms often fall short of two critical areas:

- **Supporting configuration-based development.** Despite the perceived maturity, respondents noted 63% of embedded software on average is still created via custom, project-based development rather than by configuring predefined building blocks. The need to create bespoke solutions for varying products and customer needs indicates that the level of standardization achieved so far is still low and the variety of use cases still poses a major challenge.

> On average, **63%** of embedded software still requires custom development rather than configuration of predefined components.

- **Fostering a platform-as-a-product mindset.** Over 90% of respondents agreed their platform is a strategic factor for the success of their business, supported by leadership and maintained as a product with a dedicated team, owner, backlog, etc. However, while 68% track platform feature usage and 65% track developer productivity metrics, only 38% of respondents track developer experience metrics. Since developers are the platform customers, such a low score indicates that most companies' platform-as-a-product culture is still far from matching that of typical end-customer products — where feedback is of the highest importance.

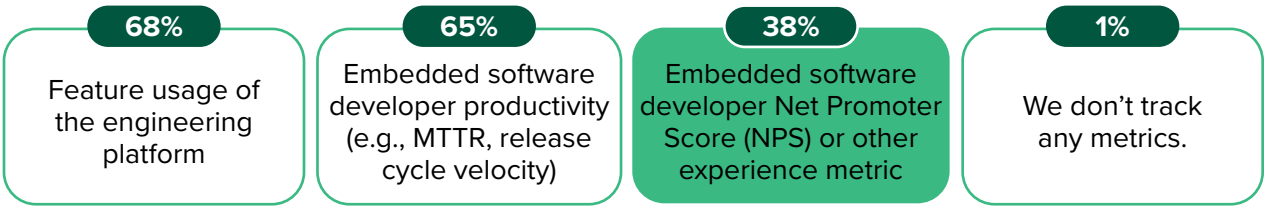
FIGURE 5

"What percentage of your embedded software is developed through the following approaches?"

Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

FIGURE 5 (CONT.)

"Which of the following platform user (e.g., embedded software developer) metrics does the platform engineering team track?"



Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

CHALLENGES PERSIST ACROSS THE DEVELOPMENT LIFECYCLE WITH CURRENT PLATFORM STRATEGIES

Despite the promise of transformative benefits, current platform strategies are struggling to meet some critical needs of embedded software development. Our study uncovered several persistent challenges that these platforms have yet to overcome, including:

- **Navigating diverse device environments.** Over half the respondents (51%) noted the top challenge of their current embedded software development approach is working across devices, operating systems, hardware, and form factors. Hardware variations in terms of processing power, memory, connectivity, and even screen sizes make it challenging to ensure consistent performance across platforms. Additionally, maintaining security and compliance remains a major hurdle, with 51% of respondents citing it as a significant challenge. Standardizing platform components while providing the necessary flexibility for embedded software developers is also problematic for 49% of respondents, which underscores the complexities involved in managing diverse use cases.
- **Facilitating cross-functional collaboration.** Collaborating across design, development, testing, and deployment is challenging for 41% of respondents. Promoting cross-functional collaboration is a bigger challenge for larger organizations and for those respondents who plan to implement a platform strategy (57%) vs. those who have already implemented it (40%) or are expanding the implementation (38%). Additionally, 43% reported difficulties in maintaining developer self-service capabilities for the majority of use cases.

- **Managing computing resource constraints.** Computing resource constraints, such as memory, energy consumption, and CPU performance, affected 38% of respondents. Moreover, 40% of respondents reported their organization struggles with managing updates and iterations of the platform while minimizing the impact on embedded software development. These constraints impede the full realization of platform engineering benefits by forcing teams to allocate resources to manage these limitations instead of focusing on forward-looking development.
- **Leveraging and integrating third-party APIs.** Third-party APIs and a software ecosystem can improve the functionality and time to market of embedded systems. However, different APIs have varying agreements and latency times and expose the systems to more security risks and privacy regulations as developers need to ensure third-party software handles sensitive data appropriately. The security and integration risks made integrating with third-party APIs and open-source software one of the most common challenges across the development lifecycle (44%). By monitoring technical debt and minimizing the number of third-party elements, platform teams can mitigate these challenges.
- **Incorporating accessibility and inclusivity in the UX/UI design process.** Many of these challenges contribute to the difficulty of incorporating accessibility and inclusivity. For instance, computing resource constraints make it difficult to implement accessibility features without compromising performance, and testing embedded systems for accessibility adds on to the complexity as simulating real-world scenarios can be resource intensive. Moreover, embedded systems also lack a unified accessibility standard, unlike web or mobile applications. Forty-four percent of respondents noted incorporating accessibility and inclusivity in the UX/UI design process is a top challenge across the embedded development cycle.

Addressing these challenges is crucial for platform engineering strategies to meet the evolving needs of embedded software developers.

The Path To Advance Your Organization's Platform Strategy

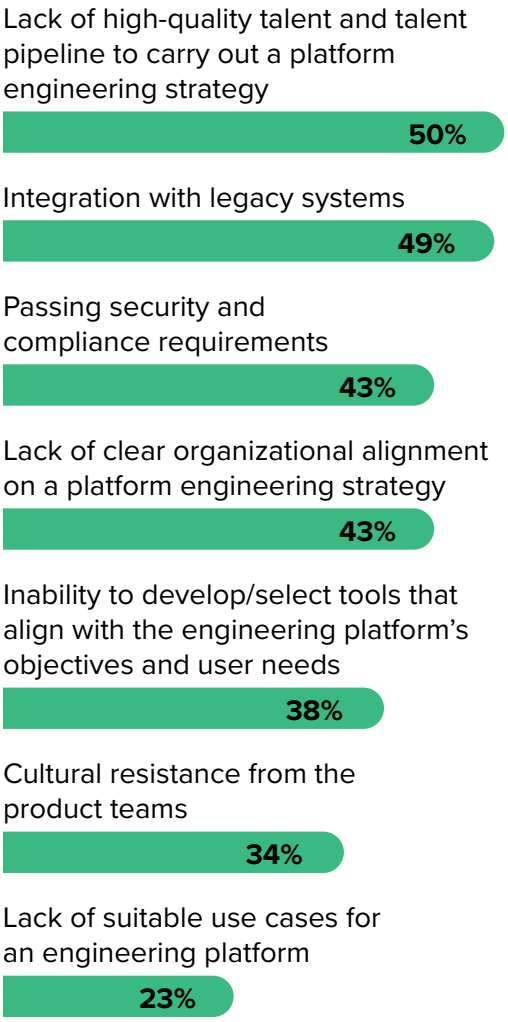
The path to implementing an effective platform strategy can be difficult, but there are tools that can accelerate progress. Survey respondents acknowledged several challenges during the transition phase, including (see Figure 6):

- **Talent shortage and legacy integration difficulties.** Given the specialized, complex, and diverse skill sets in demand for embedded development, half of the respondents (50%) identified the lack of high-quality talent as the top obstacle faced during the transition to a platform strategy. This issue is compounded by the complexity of working across different device environments, operating systems, and hardware, which often requires further knowledge specialty. Integrating legacy platforms also poses a significant challenge, with 49% of respondents struggling in this area. Finally, 34% of respondents reported cultural resistance from product teams, highlighting the importance of addressing internal pushback and promoting a culture that embraces platform teams.

> The talent shortage issue is compounded by the complexity of working across different devices, often requiring further knowledge specialty.

FIGURE 6

"What are the primary barriers you faced/ are facing when transitioning to a platform engineering strategy?"



Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

- **Security and compliance.** Ensuring security and compliance is a persistent challenge in both the transition and operational phases. The compliance requirements are distinct depending on the industry. ISO 26262:2018 defines a framework for identifying hazards, assessing risks, and implementing safety measures throughout the development process for automotive companies; medical device software development needs to follow IEC 62304:2006, which ensures software development lifecycle follows a rigorous and documented process to protect patient safety.

  During the transition phase, 43% of respondents cited difficulties in passing security and compliance requirements, given the stringent standards and regulations in this space. During the operational phase, the security challenge intensifies, since embedded teams hold full lifecycle responsibility from the first line of code until a device is decommissioned for mission-critical and safety-critical applications. These findings underscore the need for a reliable, standard-compliant, and quality-assured framework to fully harness the potential of their platforms.

- **The difficulty of maintaining a balance between standardization and flexibility.** During the operating phase, nearly half (49%) of respondents highlighted the difficulty of maintaining a balance between standardization and flexibility. On the one hand, platforms should allow developers to use standardized templates and modules while enabling customization to address unique project requirements. On the other hand, there should be awareness that requirements may change. Platforms and platform teams should thus be adaptable to such changes.

Nearly half of respondents highlighted the difficulty of maintaining a balance between standardization and flexibility.

Addressing these transitional and operational challenges is crucial for organizations to realize the full benefits of platform engineering. By investing in talent development, fostering a culture of collaboration, implementing robust security frameworks, and achieving a balance between standardization and flexibility, companies can enhance their development efficiency, ensuring compliance and driving innovation.

Spotlight: Talent Challenges In Embedded Development

Talent shortages are a common challenge in the embedded development space. As embedded engineers and developers try to adapt:

1. **Embedded engineers need to understand the end-user experience.** Experienced embedded developers are accustomed to designing electronic control units (ECUs), programmable logic controllers (PLCs), or firmware for microcontrollers. Such types of software operate machines, but embedded device manufacturers need their developers to also have a solid understanding of the end-user experience and human-machine interaction. In the embedded world, the end products' functionality is tightly intertwined with the way users access and interface with the devices. Hard-core embedded developers often lack such understanding.
2. **Web developers face a move into the embedded world.** Web developers, on the other hand, have humans and usability as their first targets. The user interface is the first access point to a service that should be delivered promptly and efficiently. However, web technologies do not necessarily scale for embedded platforms. There is a steep learning curve to understand the intricacies of embedded development, let alone the complexity of C/C++ plug-ins, cross-platform development, etc., and there may not be immediate solutions on the web.
3. **The need for unifying standards is stronger than ever.** While embedded developers need to handle increasingly complex hardware and software systems, the pervasiveness of digital controls in machines and devices calls for a clear strategy regarding the safety, quality, and reliability as part of the user experience. The need for technologies that can handle both the front-end and back-end components under a unique, easy-to-understand framework to enable efficient collaboration and high quality out of the box is stronger than ever.

Selecting The Right Tools For An Effective Platform Engineering Strategy

Choosing the right tools is crucial for the success of a platform engineering strategy, particularly in highly regulated markets with stringent quality, safety, and compliance requirements. Organizations prefer ready-made tools that act like building blocks and allow for customizations to suit their embedded platform needs.

Our study shows that organizations aiming to develop a successful platform strategy should prioritize (see Figure 7):

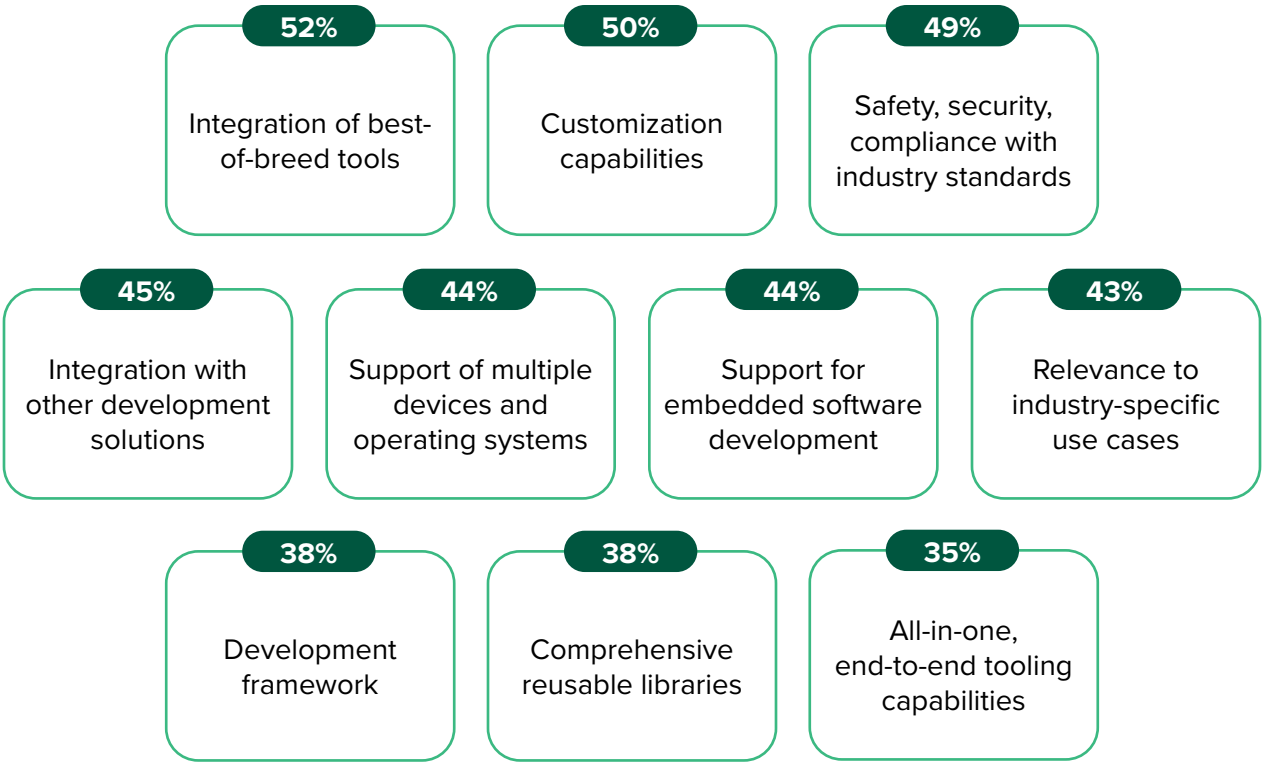
- **Best-of-breed tool integration.** Over half (52%) of respondents preferred dedicated, best-of-breed tools with integration capabilities, as opposed to only 35% who prioritized all-in-one, end-to-end tooling solutions. This preference underscores the importance of specialized tools that seamlessly work together, creating a cohesive and efficient platform ecosystem. For instance, integrating specialized security tools within the development frameworks ensures robust protection without compromising on performance.
- **Customization and flexibility.** Customization capabilities are essential for fostering a dynamic platform ecosystem and achieving strategic outcomes. Half of the respondents (50%) considered customization capabilities crucial when selecting tools for their platform. Additionally, 43% emphasized the importance of customizable APIs for third-party integrations. This flexibility allows developers to tailor solutions to specific project needs, enhancing efficiency and innovation.
- **Security, safety, and compliance.** Organizations prioritize tools with robust safety, security, and compliance features to meet the industry-specific standards required in embedded software. Nearly half (49%) of respondents saw these capabilities as essential. In the consumer electronics sector, this priority is even higher, with 53% of respondents emphasizing the importance of security and compliance in their platform engineering tool selection. With platform tools,

developers could focus on product functionality, while compliance with standards like IEC 60950 can be met and traced with minimal manual work.

By focusing on these critical areas, organizations can select tools that enhance their platform engineering strategy, ensuring the development of secure, compliant, and flexible embedded software solutions.

FIGURE 7

"When selecting platform engineering tools, which of the following are the most important capabilities?"



Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Note: Showing top 10 responses
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

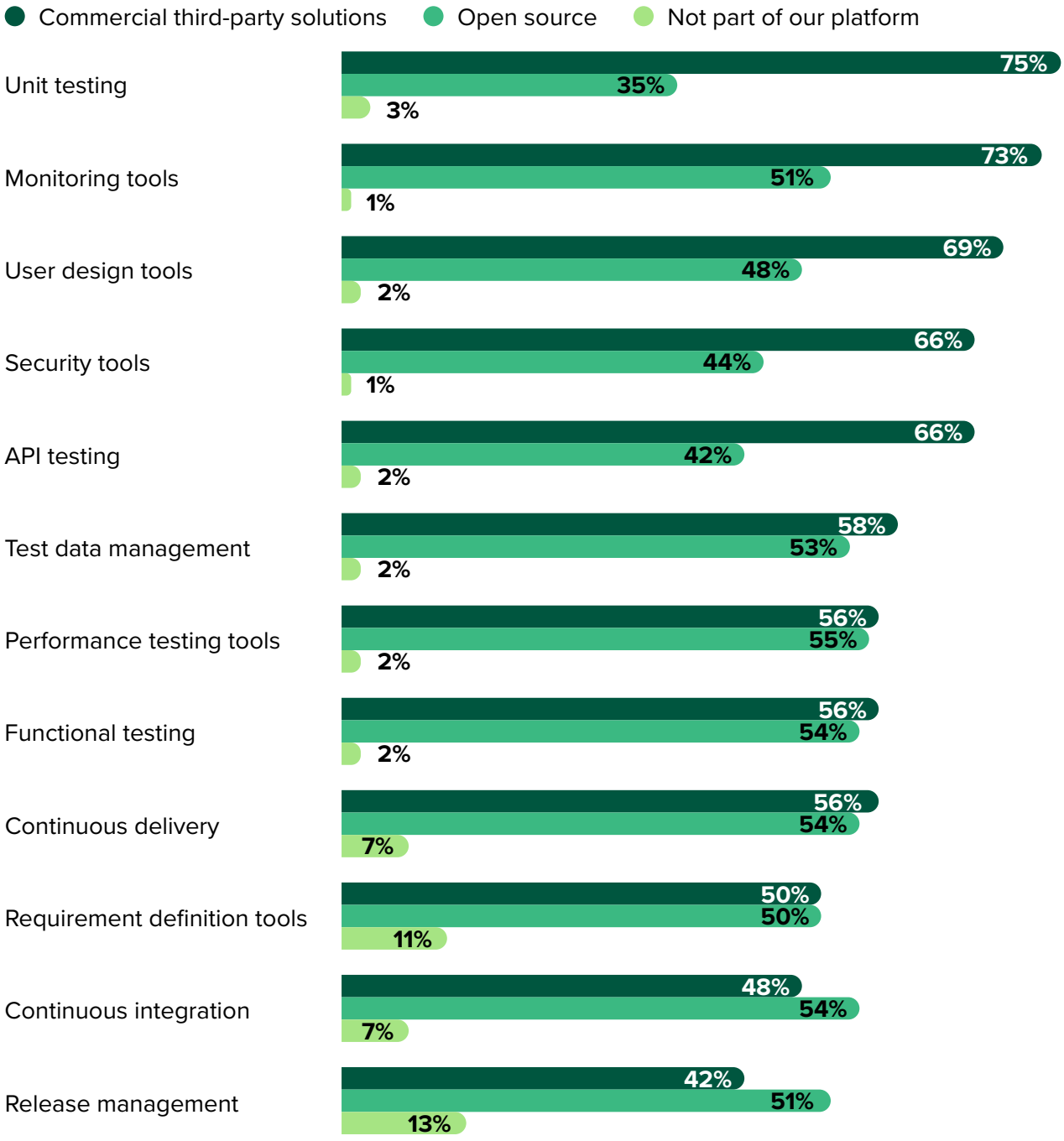
SELECT TOOLS AND READY-MADE COMPONENTS THAT COMPLEMENT YOUR PLATFORM STRATEGY

Software development tools assist developers in creating, testing, debugging, and deploying programs. They help develop the embedded software but are not part of the final product. For these tools, respondents prefer (see Figure 8):

- **Commercial unit testing, monitoring, and design tools.** Given their simplicity and security strengths, proven commercial third-party tools remain the preferred choice for engineering platforms. Commercial tools offer established frameworks and methodologies that provide consistency across projects. Commercial unit testing tools usually provide more advanced features, such as supporting more complex data structures.
- **Commercial security and API testing tools.** Commercial API tools offer more advanced features and enterprise-grade security compared to open-source options. Many also provide dedicated support to ease the talent shortage faced by embedded teams.
- **Open-source performance testing, functional testing, and continuous integration tools.** Embedded developers, especially in safety-critical applications, are typically reluctant to use open-source tools for their embedded software development due to legal exposure concerns, liability issues, and high-security standards. Yet, some well-maintained open-source tools are gaining popularity as certain companies believe their transparency allows the entire community to identify and address security risks. According to our survey, open-source solutions for performance testing (55%), functional testing (54%), and continuous integration (52%) compete today with commercial tools as the preferred solutions for platform engineering.

FIGURE 8

“For which of the following tools does your platform engineering team use open source vs. commercial solutions across the software development lifecycle?”



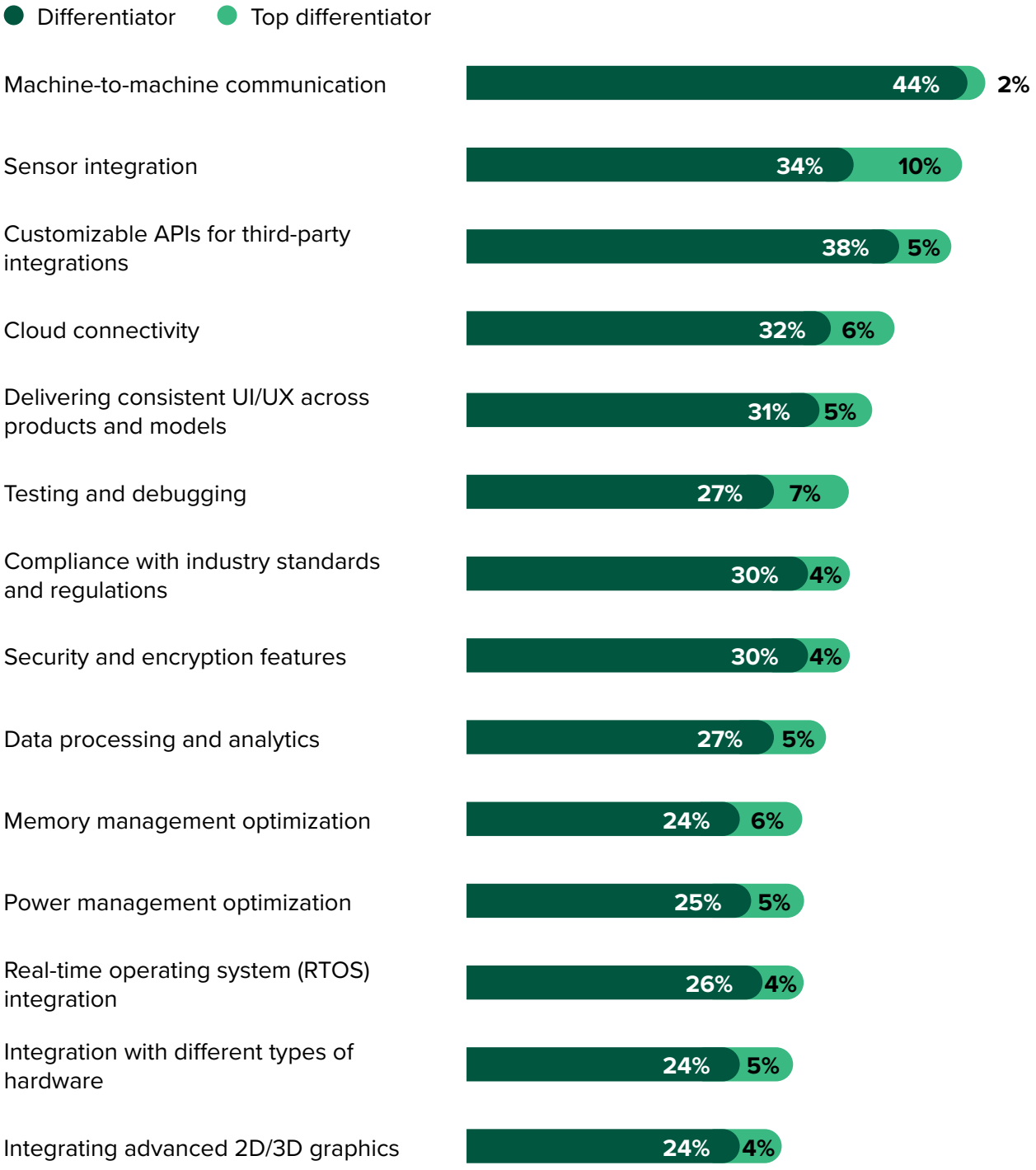
Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

In terms of ready-made components, top differentiators are those features or components organizations deem most critical in the choice of an engineering platform, including (see Figure 9):

- **Machine-to-machine communication and sensor integration.** With the overarching trend towards IoT, machine-to-machine communication empowers devices to become interconnected, making it a top differentiating component (46%). Sensor integration — enabling data collection — ranked as the second highest differentiator (44%). Given the strong device focus of embedded developers, cloud connectivity ranked lower than machine-to-machine and sensor, while still 38% considered it a differentiator.
- **Flexible development through customizable APIs.** Allowing developers to tailor third-party integration to the specific needs of the system, customizable APIs ranked as the third differentiator (44%). This also corresponds to the difficulty of integrating third-party APIs — one of the top challenges across the development cycle.
- **Consistent UXs.** Delivering a consistent UI/UX across products and models was also a significant differentiator (36%). This consistency ensures a uniform user experience, which is crucial for brand identity and for usability. As incorporating accessibility and inclusivity was a top UX challenge, developers may leverage libraries providing high-quality, ready-made components to alleviate this issue.
- **Prioritize security, safety, and compliance.** Compliance with industry standards and regulations and security and encryption features are critical, especially given the increased number of cyberthreats and new European legislation. Thirty-four percent of respondents considered them differentiators.

FIGURE 9

Most Differentiating Ready-Made Components



Base: 317 embedded software decision-makers at organizations in consumer electronics, industrial automation, medical devices, and motor vehicles industries
Source: A commissioned study conducted by Forrester Consulting on behalf of Qt Group, April 2024

Key Recommendations

While many respondents see their platform as mature, the level of standardization, reusability, and self-service capabilities remains low, and software is still custom-built for the specificity of the varying use cases. To bridge gaps and enhance the effectiveness of platform engineering, organizations must evaluate the current state of their platform strategies, document what they have, and determine how it helps reach their goals. A series of actions will help elevate platform engineering practices.

Leverage a flexible, scalable framework.

With rapid technological evolution, your platform should support change by design, be open to integration with ever-evolving technology, and be compatible with a variety of hardware and software platforms. Understand how many scenarios your platform can cover and leave some leeway for ad hoc code and platform updates to avoid hindering productivity in new or unexpected situations.

Use optimized cross-platform components.

Even with a platform in place, embedded developers write more custom code than expected and are challenged by the lack of self-service capabilities. Improve platform engineering practices by reusing cross-platform components that are easily deployable and embrace best practices, such as compliance with industry standards.

Enhance cross-functional collaboration.

Enabling cross-functional collaboration across the various stages of the product lifecycle is a priority for scaling the platform approach across departments, ensuring cohesiveness across processes and consistency across products. Enhance your agile practices and enforce a more effective division of labor by leveraging tools for each stage of the development cycle that communicate with each other efficiently and allow teams to work on a single source of truth.

Address quality and regulatory compliance.

Safety is a primary concern in embedded software development and a key selection criterion for platform tools. To maintain an effective platform and meet demanding industry standards, work with quality-assured partners providing compliance suites and static code analysis. Ensuring safety and regulatory compliance is the foundation of a robust platform in the embedded software domain.

Foster a thriving ecosystem and talent community.

Finding the right talent for platform engineering is a significant challenge. Cultivate a thriving ecosystem and talent community by coaching developers in platform engineering skills, benefits, and practices. Consider leveraging well-established solutions with robust ecosystems and active developer communities to support and enhance your platform strategy.

Appendix A: Methodology

In this study, Forrester conducted an online survey of 317 embedded software decision-makers at organizations in the United States, Canada, the United Kingdom, Germany, and France. Survey participants were asked the key drivers for platform engineering strategy, the business benefits, challenges developers face today, and the methods used to support platform strategies. Survey participants included key decision-makers, influencers, or users driving their organization's platform engineering strategy that have implemented and/or planning to implement a software platform engineering strategy in the next 12 months. Respondents were offered a small incentive as a thank you for time spent on the survey. The survey component was completed in April 2024 and finalized in June 2024.

Appendix B: Demographics

GEOGRAPHY	
Canada	**11%**
France	**19%**
Germany	**20%**
United Kingdom	**31%**
United States	**19%**

TITLE	
C-level executive	**6%**
Vice president	**15%**
Director	**38%**
Manager	**40%**

INDUSTRY	
Consumer electronics	**26%**
Motor vehicle manufacturing	**25%**
Industrial automation	**25%**
Medical devices	**24%**

PRIMARY TECHNOLOGY RESPONSIBILITY	
Software/application development	**100%**
Product management/product owner	**24%**
IT security and risk	**22%**
IT services management	**20%**
Program/project management	**20%**
Data management	**20%**
Network infrastructure	**15%**
Strategy/enterprise or technology architecture	**15%**
User device support and operations	**13%**
Hardware or data center operations	**11%**
Internal IT support	**9%**

Appendix B: Demographics, cont.

COMPANY SIZE	
20,000 or more employees	**9%**
5,000 to 19,999 employees	**43%**
1,000 to 4,999 employees	**48%**

COMPANY ANNUAL REVENUE	
>$5B	**34%**
$1B to $5B	**34%**
$500M to $999M	**32%**

DEPARTMENT	
IT	**62%**
Product development	**38%**

Note: Percentages may not total 100 due to rounding.

Appendix C: Supplemental Material

RELATED FORRESTER RESEARCH

The Best Tech Organizations Are Future Fit, Forrester Research, Inc., September 8, 2023.

Forrester's Guide To Great Developer Experience For IoT And Embedded Developers, Forrester Research, Inc., May 17, 2024.

Accelerate With Platform Engineering, Forrester Research, Inc., April 3, 2024.


FORRESTER®