



2024 State of Internal Developer Portals

December 2023

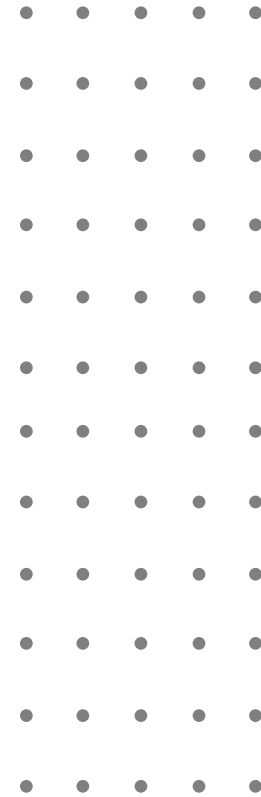
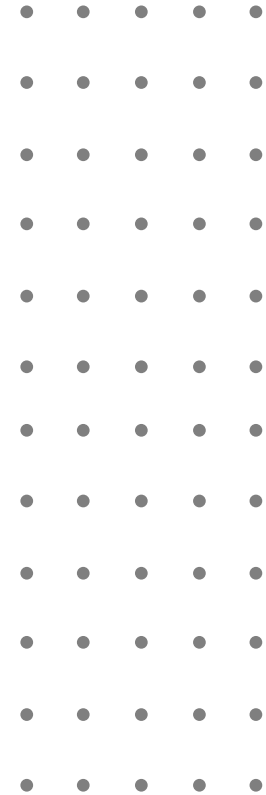


Table of Contents

Introduction and Key Findings	3
Internal Developer Portal Insights	7
Are you using an internal developer portal? 35% plan to do so in next 12 months	8
What type of internal developer portal is it?	9
What is the Main Measure of the Success for the Internal Developer Portal?	10
Time to deploy to production and non-core work as developer productivity indicators.....	11
How do you measure developer productivity? Surveys and reports, not frameworks	12
What was the main motivation that led to the decision to implement a portal?	13
Standards, productivity and engineering velocity are driving platform engineering	14
Only 34% use the portal to drive engineering quality and compliance	15
Platform Engineering Adoption Insights	16
Almost 99% of respondents report some level of platform engineering adoption.....	17
83% report implementing GitOps into their current DevOps practices	18
Demographics	19
About Port	22

Introduction and Key Findings



Introduction & Methodology

This report plants the first flag in understanding the state of internal developer portals in engineering organizations.

2023 was, in many ways, the year of platform engineering, with internal developer portals recognized in Gartner's *Hype Cycle for Software Engineering 2023* and acknowledged as the most mature tool to drive platform engineering in Gartner's *Market Guide for Internal Developer Portals*. Companies began defining and hiring developer experience roles tasked with setting up a portal to drive developer productivity and even increase happiness.

Despite this massive wave of adoption, the market seems to be in its initial stages, seeking to define the value of platforms and portals and determine what they're made of and what they can offer. That's why we launched this report, using a survey to assess how the market sees internal developer portals and possibly show ways in which the market can evolve, as well as understand critical pains and drivers for change in a world where developer productivity is a prominent issue.

This report surveyed 100 engineering leaders to answer questions such as:

- Have you adopted platform engineering and when
- Why did you want a portal and what type of portal did you use
- How do you think about engineering standards, and more

Methodology

We surveyed 100 full-time employees from the US and Western Europe (Germany, France, UK, Netherlands) working in companies with 150 or more developers. Respondents included platform engineers, developer experience professionals, platform product managers, and site reliability engineers. Our sample represented companies utilizing a microservice architecture in production within the engineering department.

Geographically, 50% of respondents were from the US, and the remaining 50% from Western Europe. The survey was conducted in October 2023 in collaboration with Global Surveyz, an independent survey company.

Key Findings

1 An overwhelming majority of companies see the importance of internal developer portals and are using or implementing them

Almost 85% of respondents surveyed have either started implementing internal developer portals or are planning to do so in the next year. Virtually all (99%) report that they have begun using platform engineering in their organizations, with 53% reporting that they've begun in the past year. This implies that companies are dedicated to incorporating portals as a fundamental aspect of their commitment to platform engineering.

2 The market is still unclear about what constitutes an internal developer portal

When asked what type of internal developer portal companies use, an amazing 35% reported using spreadsheets with microservice data. However, using spreadsheets should not be considered a true internal developer portal given its highly manual nature and the absence of developer self-service capabilities. 53% utilize various forms of developer portals, including in-house, backstage, and commercial products. Notably, smaller companies tend to have self-service interfaces that aren't portals, while large companies are more likely to use in-house portals.

3 Surveys and custom reports trump developer productivity measurement frameworks

Most respondents (56%) define a successful portal as one that improves developer productivity. Yet 75% of respondents used surveys or custom reporting to measure developer productivity instead of recognized frameworks such as space or DORA.

4

Without platform engineering, developers spend significant time on non-core work. Regardless of platform engineering choice, they face prolonged software deployment timelines.

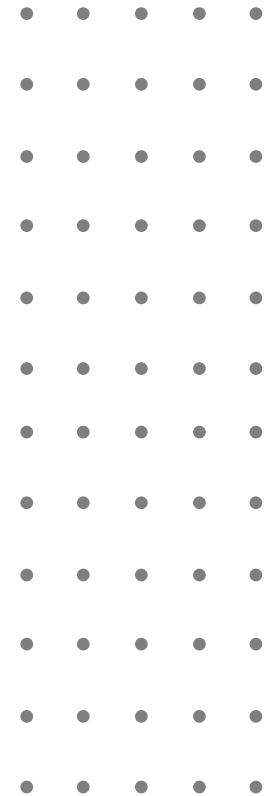
Virtually every organization (99%) has initiated some form of platform engineering, with 83% incorporating GitOps into their approach, while the remaining organizations are on the verge of adopting it. When asked about the daily time developers spend on non-core tasks due to the absence of platform engineering, a staggering 70% revealed that developers spend 3-4 hours each day on such activities. Additionally, for 68% of respondents, the average lead time to deploy software to production is several weeks or even months.

5

Only 34% of respondents use internal developer portals to drive engineering standards, the primary objective of platform engineering

According to survey respondents, the primary objective in implementing platform engineering is to enhance software quality and engineering standards, closely followed by goals of improving developer productivity and engineering velocity. However, the survey reveals that only 34% of respondents utilize internal developer portals for driving engineering standards. This discrepancy underscores the potential for broader adoption and exploration of internal developer portals as powerful tools in driving and maintaining engineering standards.

Internal Developer Portal Insights



Are you using an internal developer portal? 35% plan to do so in next 12 months

50% of respondents already use an internal developer portal, with 35% planning to use one in the next 12 months.

In smaller companies (<300 developers), there is less reported usage of portals (43%), with 45% planning on adding a portal within the next 12 months. Larger companies report greater use of portals (57%), signifying a higher rate of adoption.

This makes sense given that platform engineering and internal developer portals are in greater need for larger engineering teams, where greater microservice complexity, onboarding needs, and DevOps-developer divides usually appear.

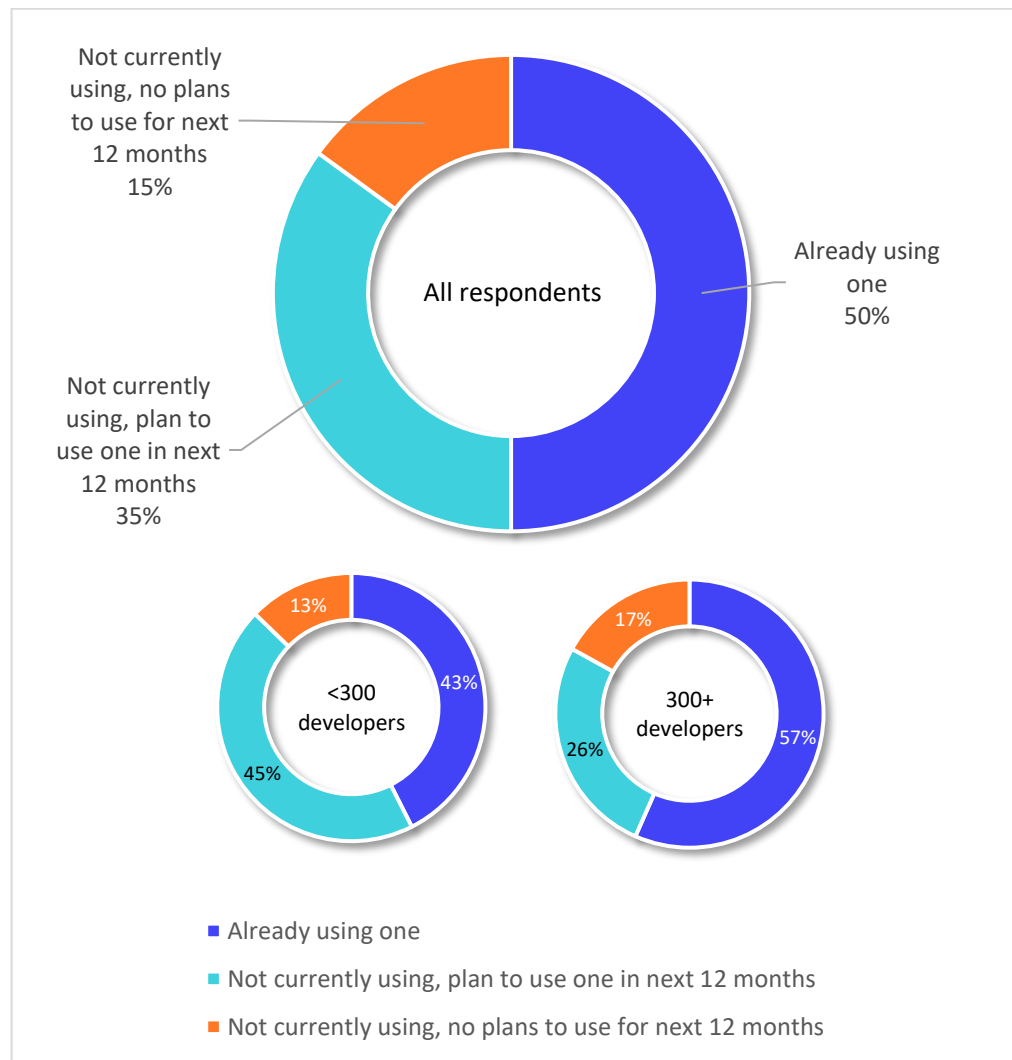


Figure 1: Use of Internal Developer Portal

What type of internal developer portal is it?

85% of respondents reported using an internal developer portal or planning to add one. However, only 53% use what we define as a portal, while 35% use spreadsheets with microservice data and 12% use a self-service platform that isn't a portal (e.g. CI), both of which we do not define as an internal developer portal.

The data suggests that the market may be wrongly assuming that software catalog capability in a spreadsheet or a self-service CI constitute an internal developer portal. This is a sign that the pillars of the internal developer portal are still not completely understood by respondents.

Notably, 18% of larger engineering group size companies have in-house developer portals, compared to only 7% for the smaller companies. Larger engineering teams are also more likely to have a developer portal based on a commercial product (22% vs 12%) and less likely to use a self-service platform that isn't a portal (7% vs 17%). This makes sense, as self-service platforms don't scale well, and the sheer size of the engineering organization may justify investment in an in-house developer portal.

Finally, we note that backstage use is almost the same for smaller and larger engineering organizations.

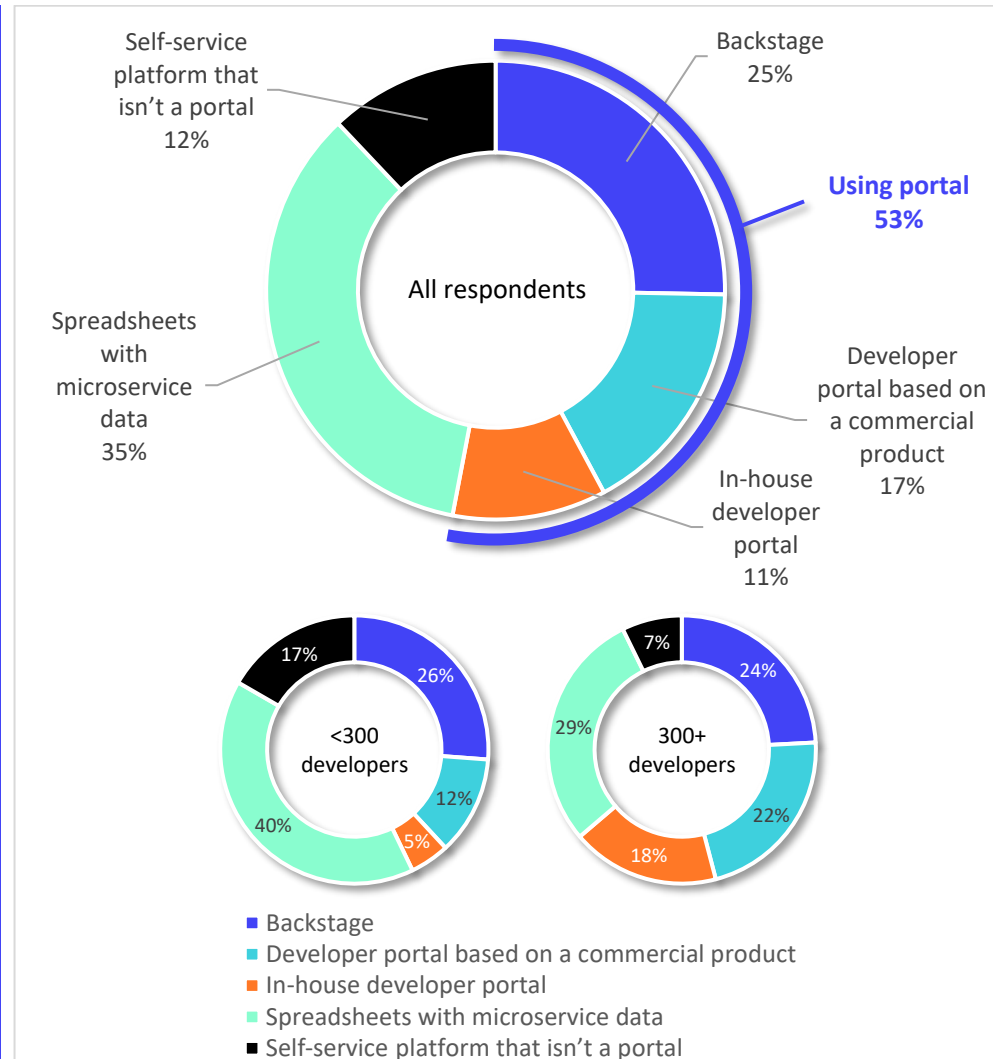


Figure 2: Type of Internal Developer Portal

What is the main measure of success for the internal developer portal?

Of the 85% of respondents using or planning to use a portal, the main measure of success was better developer productivity (56%) followed by reduced time to deployment (25%). This shows that respondents understand that portals are meant to improve productivity and create a tangible result by allowing for faster deployment.

Only 11% viewed the mere use of a portal as a metric of success, and 8% chose reducing tickets as the goal of the portal.

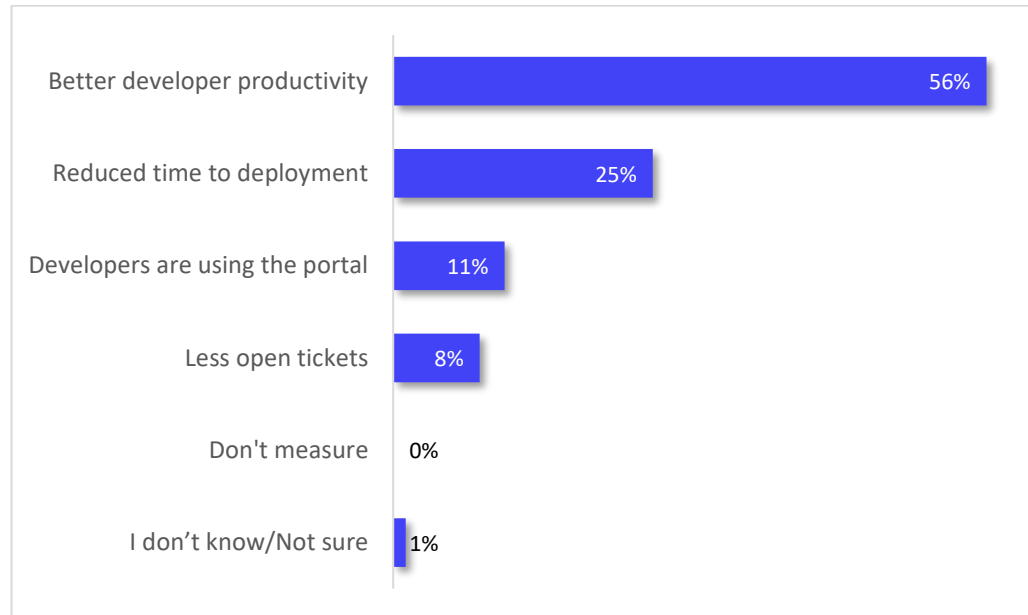


Figure 3: Main Measure of the Success of Internal Developer Portal

Time to deploy to production and non-core work as developer productivity indicators

We asked how much time developers spend each day on non-core work because of not having platform engineering in place (Figure 4).

78% of respondents report three or more hours of developers' time spent daily on non-core work.

We also inquired about the average lead time to deploy software to production (Figure 5). We found that 68% of respondents have an average lead time of several weeks or months.

These results explain why the industry is focusing on developer productivity, platform engineering, and the reduction of cognitive load.

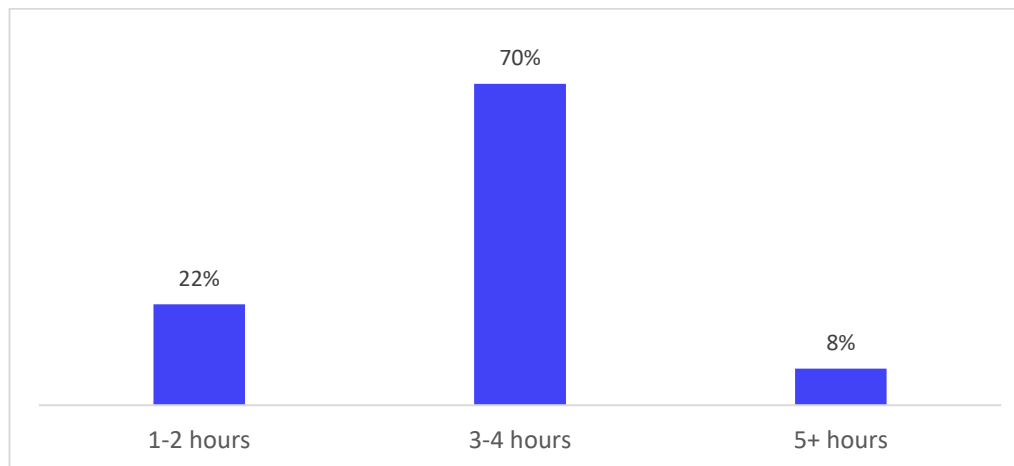


Figure 4: Developers Time Spent on Non-Core Work Due to Lack of Platform Engineering

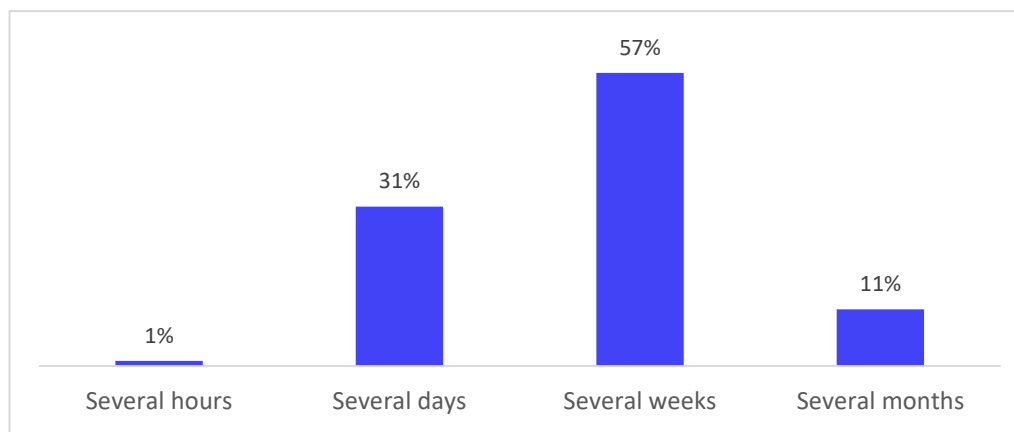


Figure 5: Average Lead Time to Deploy Software to Production

*Question allowed more than one answer and as a result, percentages will add up to more than 100%

How do you measure developer productivity? Surveys and reports, not frameworks

Surveys (43%) and custom reports (31%) are the most common ways of measuring productivity, signifying a focus on either self-reported productivity or custom measures of productivity instead of frameworks such as the DevEx framework (15%), DORA metrics (5%) and SPACE (5%).

This may suggest that these frameworks are difficult to implement or that surveys and custom reports provide valuable insights that aren't necessarily captured by the frameworks.

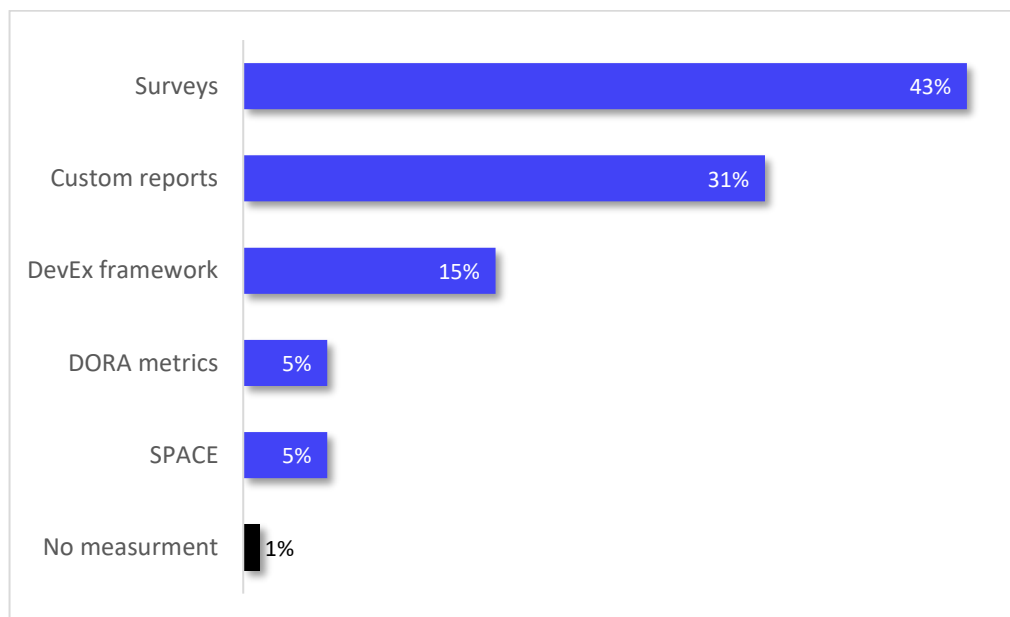


Figure 6: Developer Productivity Measurement Practices

What was the main motivation that led to the decision to implement a portal?

The core driver of the platform engineering movement is to give developers a higher degree of agency and free up the time of DevOps and platform teams. This is done by exposing managed and reusable self-service actions, which developers can consume on their own, thereby reducing the need for redundant work by DevOps teams and minimizing repeated requests and repetitive work.

This aligns with our finding that 44% of respondents cited “preventing redundant work” as their top reason for adopting a portal, followed by faster deployment cycles (43%), easing developer cognitive load (38%) and easing DevOps fatigue (36%).

Surprisingly, lack of Kubernetes knowledge and dealing with AppSec, which are often cited as main drivers of portal adoption, were listed last.

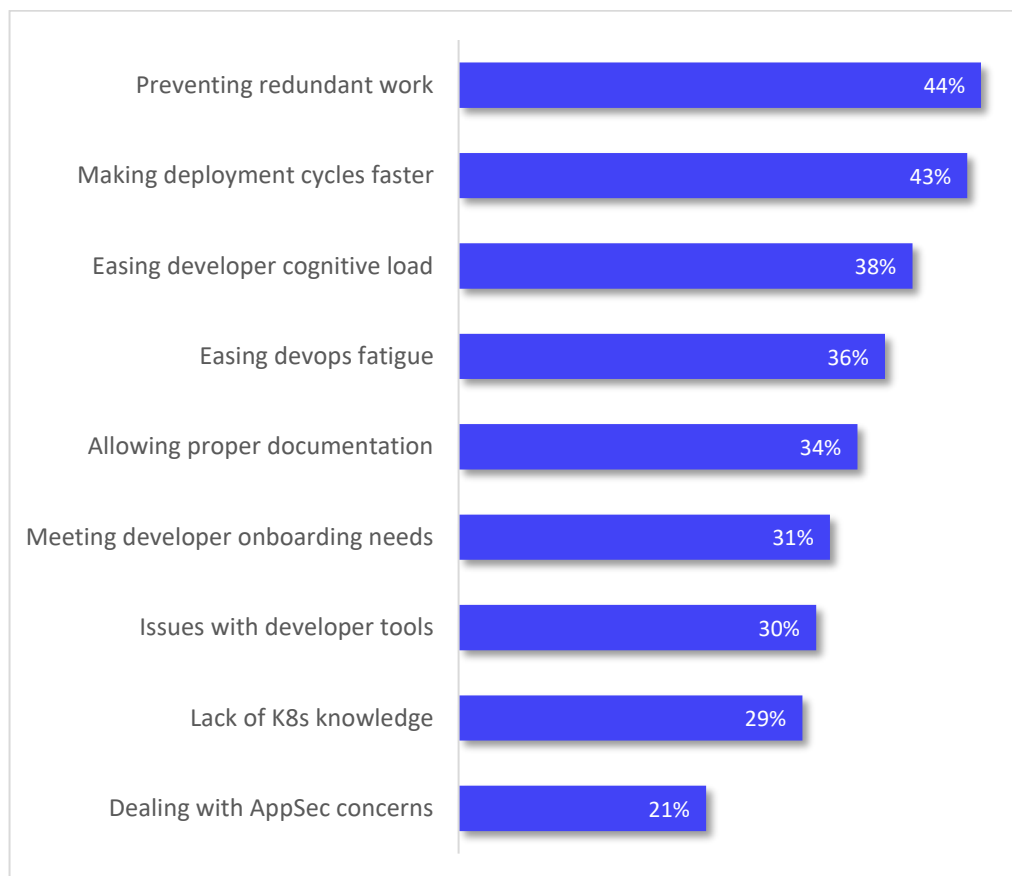


Figure 7: Main Motivations to Implement an Internal Developer Portal

*Question allowed more than one answer and as a result, percentages will add up to more than 100%

Standards, productivity and engineering velocity are driving platform engineering

Respondents were asked what they believe the top goals of platform engineering to be. The results were improving software quality or engineering standards (48%), developer productivity (46%), better engineering velocity (44%), driving engineering standards through initiatives (38%), and reducing developer cognitive load (34%).

All these goals are connected to meeting engineering standards and driving productivity. In contrast, the bottom four goals represent distinct use cases for platform engineering and portals, and none of them seem to be a single strong driver of platform engineering.



Figure 8: Top Goals of Platform Engineering

*Question allowed more than one answer and as a result, percentages will add up to more than 100%

Only 34% use the portal to drive engineering quality and compliance

We asked, “What is your main tool to ensure engineering quality and compliance in your organization?”.

The data shows that developer portals are not necessarily perceived as a core driver for standards and compliance, with only 34% using initiatives in an internal developer portal for this purpose. Documentation (46%) is the main tool, while 19% use manual checklists. Notably, only 1% of respondents reported not ensuring engineering quality and compliance at all.

Breaking down the data by engineering group size reveals that larger teams are more committed to both documentation and using the portal and are less reliant on manual checklists.

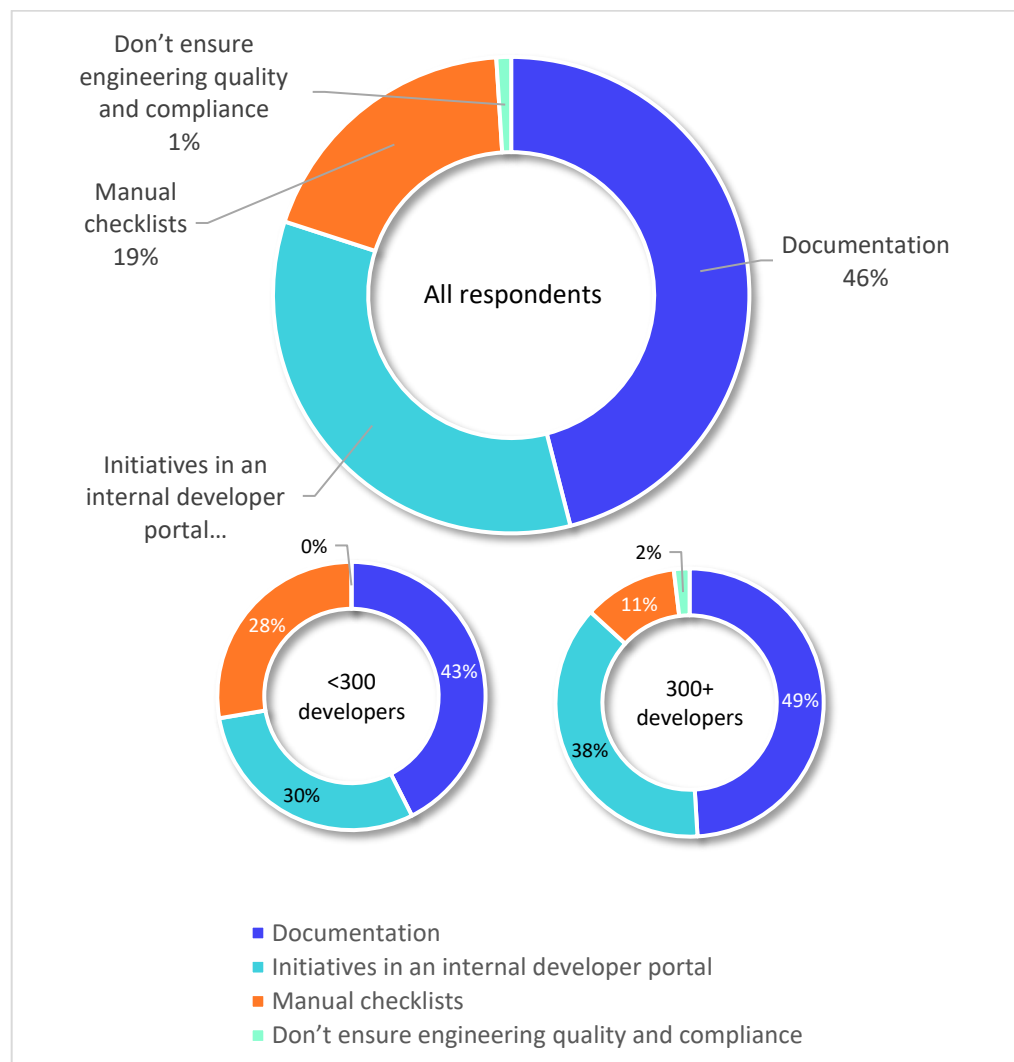
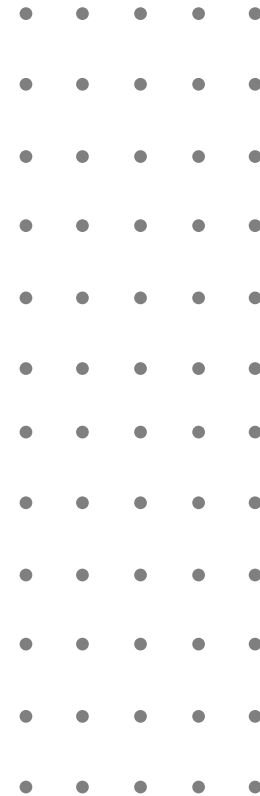


Figure 9: Tools to Ensure Engineering Quality and Compliance

Platform Engineering Adoption Insights



Almost 99% of respondents report some level of platform engineering adoption

We also asked survey respondents when their organizations began implementing platform engineering. 11% reported starting the process in the past 5 years, 35% within the past 3 years, 53% in the past year, and 1% are planning to implement in the upcoming year.

A regional breakdown reveals that North American organizations began implementing platform engineering earlier than European ones, with 62% implementing platform engineering in the past three and five years, compared to 30% for European organizations. However, 70% of European respondents reported a plan to begin implementing platform engineering in the coming year.

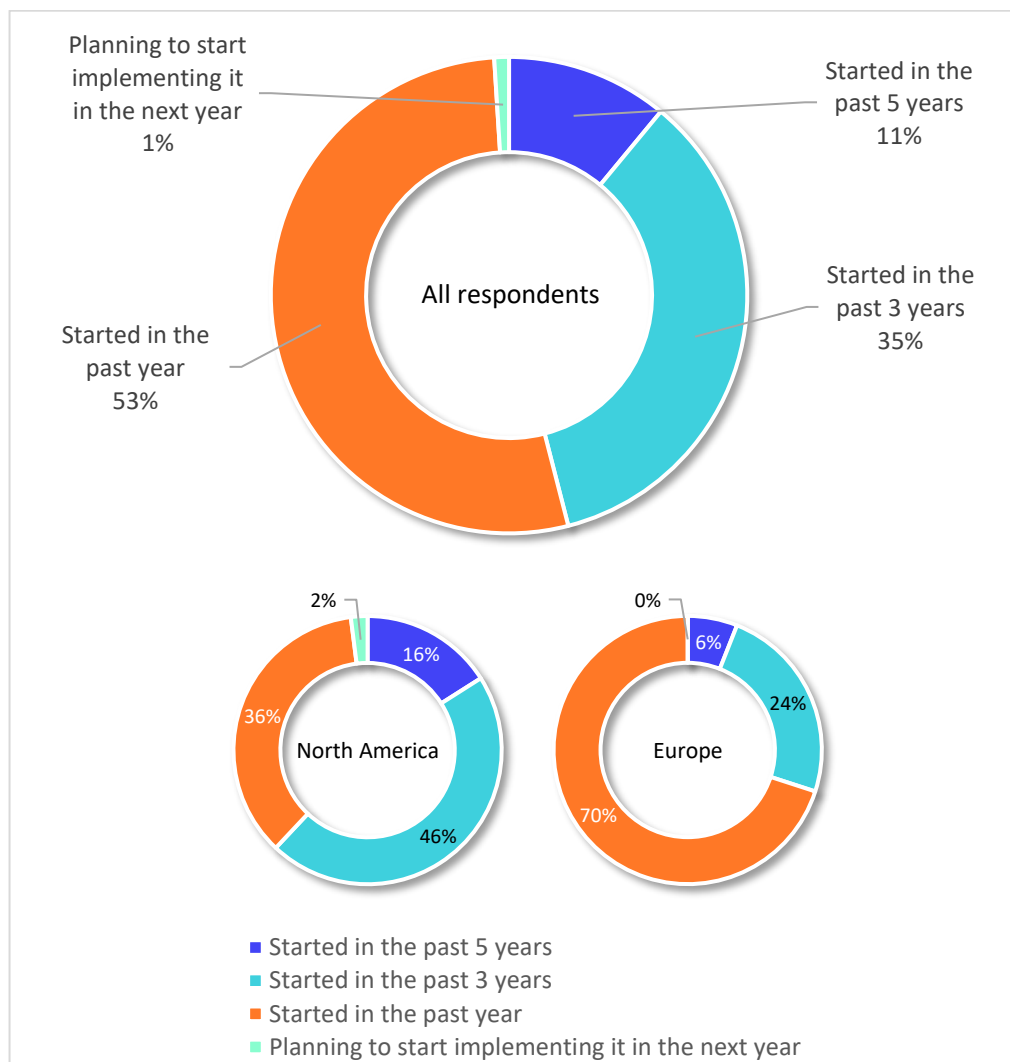


Figure 10: Platform Engineering Implementation

83% report implementing GitOps into their current DevOps practices

A noteworthy 83% of survey respondents report a partial implementation of GitOps into their current DevOps practices, underscoring its status as a foundational best practice for both platform engineering and internal developer portals. Additionally, 17% of respondents express an intention to implement GitOps within the next 12 months.

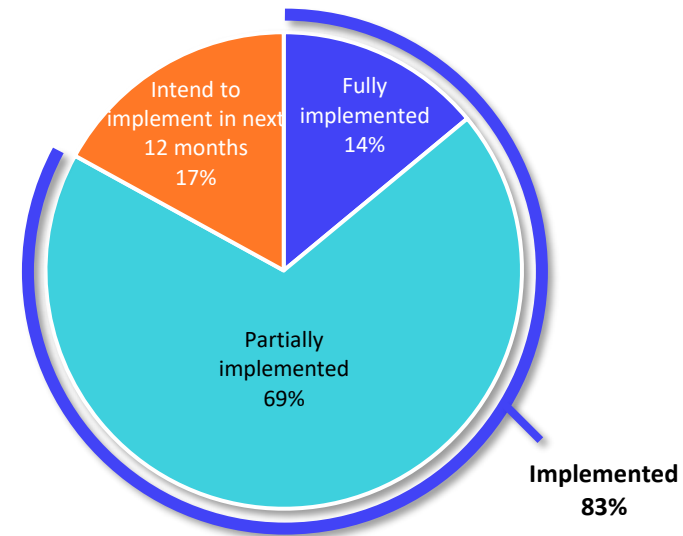
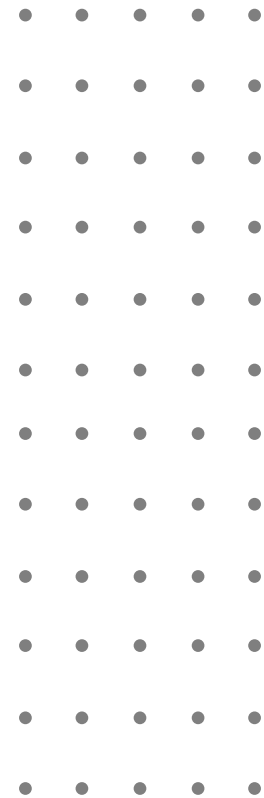


Figure 11: Level of Implementation of GitOps into Current DevOps Practices



Demographics

Industry, Country, and Company Size

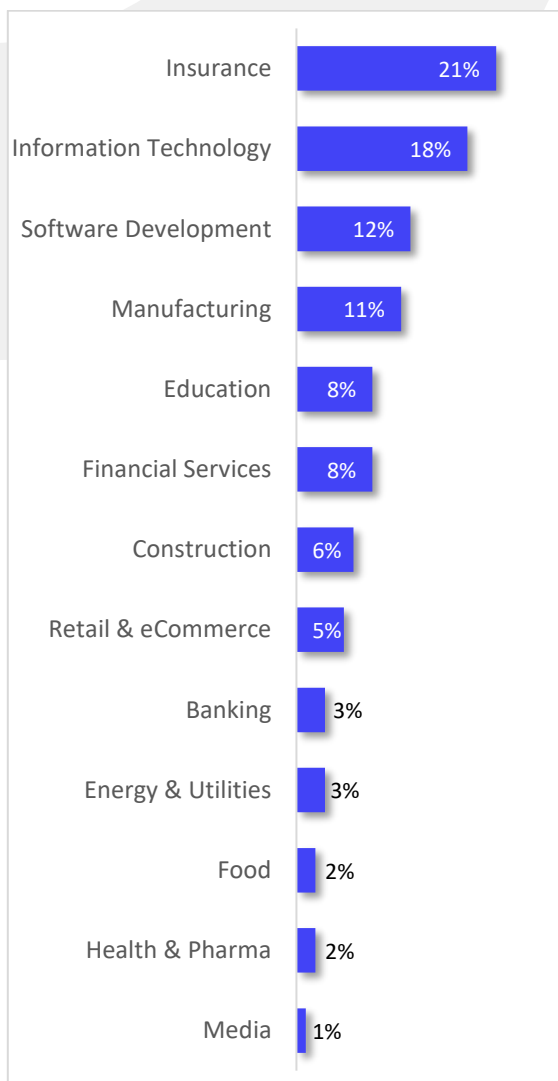


Figure 12: Industry

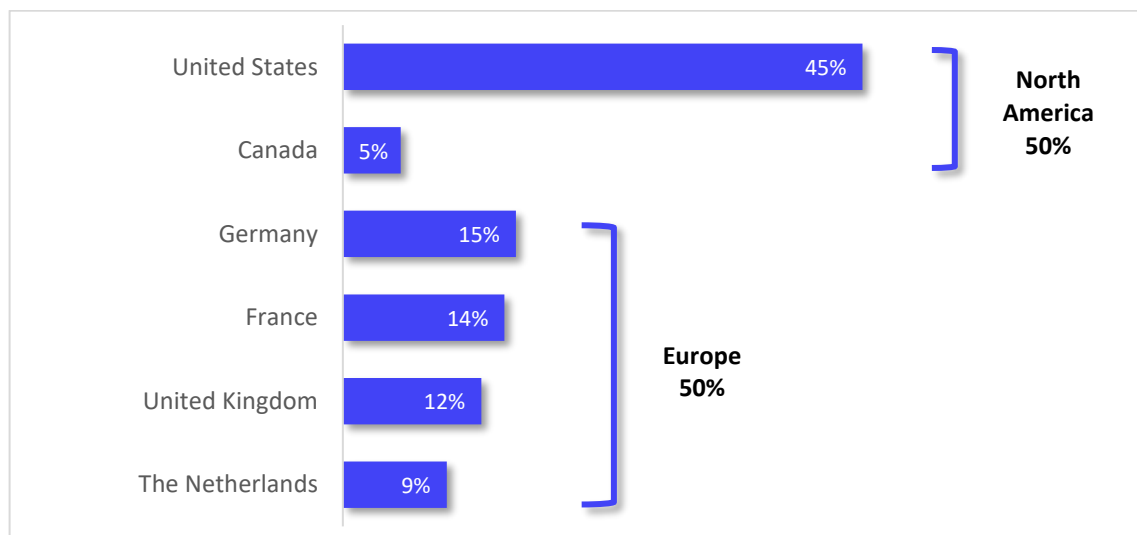


Figure 13: Country

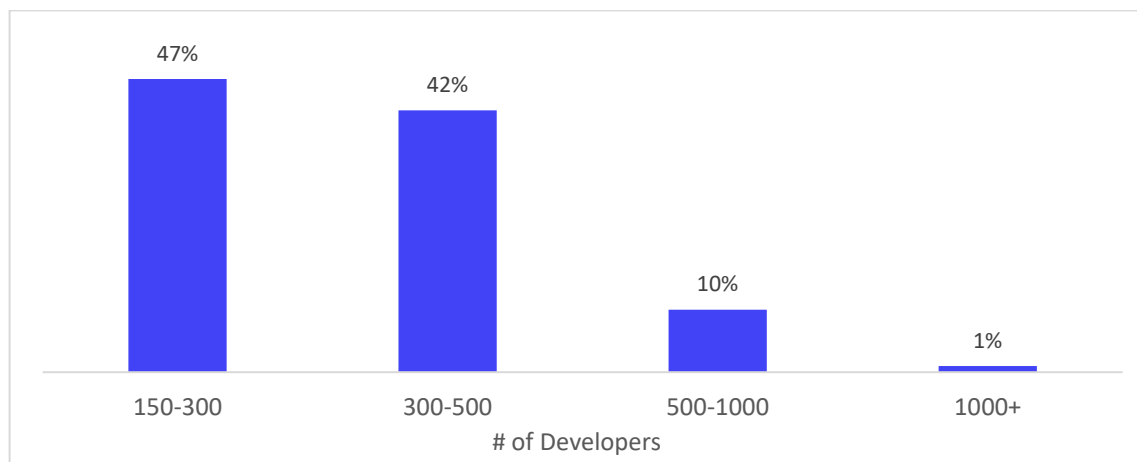


Figure 14: Company Size

Job Seniority, and Job Function

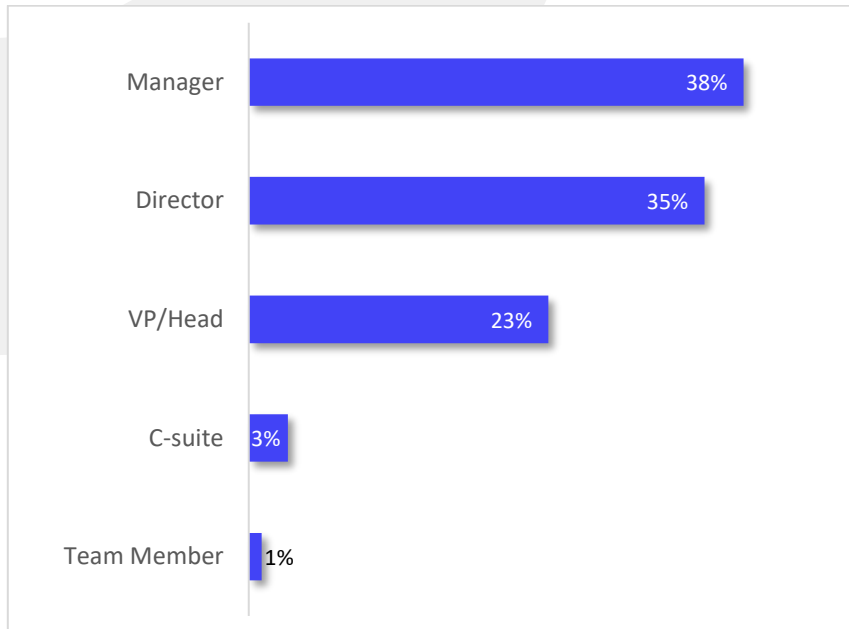


Figure 15: Job Seniority

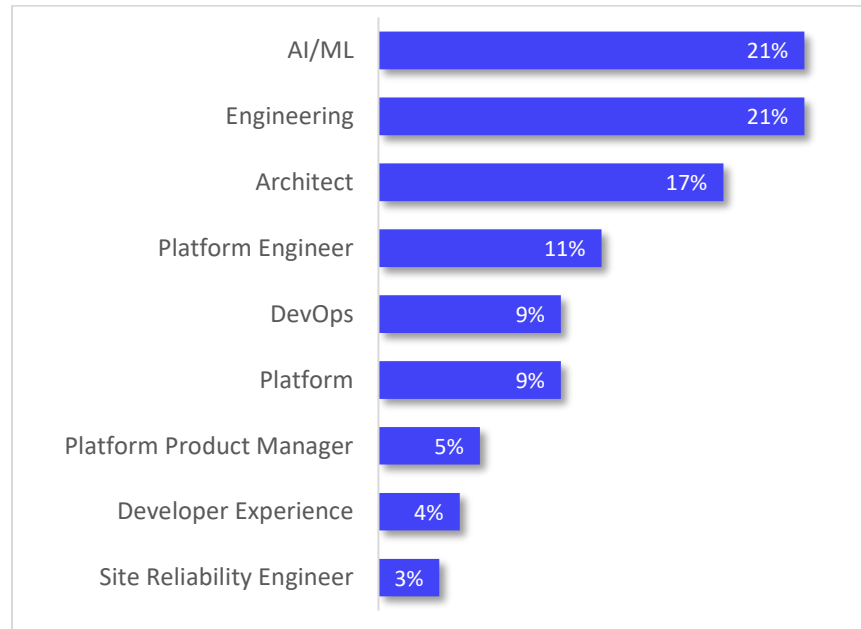


Figure 16: Job Function

About Port

Port is an open internal developer portal, owned by platform engineering teams and built for developers. Port consolidates everything developers need to know and execute to deliver software autonomously and to comply with organizational standards.

You can open a free Port account at getport.io.

Try Port now

For more information, please visit us:



Phone: +1 (646) 111-1111 | Email: info@getport.io