The state of developer happiness

June 2023

Created by

/DATA SENTRY

/DATA

About SlashData

SlashData is the leading analyst company in the developer economy, tracking global software developer trends based on more than 30,000 software developers annually in over 160 countries. Our surveys track the changing landscape of Web, Desktop, Cloud, Mobile, Industrial IoT, AR/VR, Machine Learning and Data Science, Games, Consumer Electronics and Apps/Extensions for 3rd party ecosystems.

Understand developers. Inspire the future of technology.

SlashData Ltd.
Swan Buildings (1st floor)
20 Swan Street
Manchester, M4 5JW
+44 161 240 0603
hello@slashdata.co

/DATA

Understand developers. Inspire the future of technology.

We survey 30,000+ developers annually – across Web, Desktop, Cloud, Mobile, Industrial IoT, AR/VR, Machine Learning and Data Science, Games, Consumer Electronics and Apps/Extensions for 3rd party ecosystems - to help companies understand who developers are, what they buy and where they are going next.

WHO DEVELOPERS ARE

Developer population sizing
Developer segmentation

WHAT THEY BUY

Why developers are adopting competitor products – and how you can fix that

WHERE THEY ARE GOING

Emerging platforms – augmented & virtual reality, machine learning

Can I share data from this report?

1. License Grant
This report is licensed under the Creative Commons Attribution-NoDerivatives Licence 4.0 (International) . Put simply, subject to the terms and conditions of this license, you are free to:

Share — You can reproduce the report or incorporate parts of the report into one or more documents or publications, for commercial and non-commercial purposes.

Under the following conditions:

Attribution — You must give appropriate credit to SlashData™ , and Sentry, as sponsors of this report, and indicate if changes were made. In that case, you may do so in any reasonable manner, but not in any way that suggests that SlashData™ endorses you or your use.

NoDerivatives — you cannot remix or transform the content of the report. You may not distribute modified content.

2. Limitation of Liability
SlashData™, believes the statements contained in this publication to be based upon information that we consider reliable, but we do not represent that it is accurate or complete and it should not be relied upon as such. Opinions expressed are current opinions as of the date appearing in this publication only and the information, including the opinions contained herein, are subject to change without notice. Use of this publication by any third party for whatever purpose should not and does not absolve such third party from using due diligence in verifying the publication's contents. SlashData™ disclaims all implied warranties, including, without limitation, warranties of merchantability or fitness for a particular purpose.

SlashData™, its affiliates, and representatives shall have no liability for any direct, incidental, special, or consequential damages or lost profits, if any, suffered by any third party as a result of decisions made, or not made, or actions taken, or not taken, based on this publication.

The analyst of the developer economy | formerly known as VisionMobile

ABOUT THE AUTHORS

Brayton Noll

Senior Market Research Analyst

Brayton Noll is a behavioral scientist with a background in climate change and environmental research. He holds a PhD from TU Delft in computational social-science with his thesis focusing on human behavioral dynamics and climate adaptation. He has five years of experience working with data analytics.

brayton.noll@slashdata.co

Konstantinos Korakitis

Director of Research

Konstantinos heads the Research Product team at SlashData and is responsible for all syndicated research products and custom research projects. With more than 10 years of experience as an engineer, consultant and manager, he oversees research planning, survey design, data analysis, insights generation and research operations.

konstantinos@slashdata.co

TABLE OF CONTENTS

INTRODUCTION

In Q1 2023, SlashData designed and hosted an online survey on behalf of Sentry, with the principle aim of better understanding developers and their workflows. The survey outreach targeted professional developers who spend at least 20% of their working time writing software and work for a company that has at least one live application. The survey was fielded by SlashData in conjunction with Sentry, and reached more than 1,100 developers across the globe, working for a variety of companies and holding a number of different roles.

The majority of respondents analysed in this report are based in North America and Western Europe, with 30% of the respondents coming from each of the respective regions. The remaining 40% of the developers surveyed come from various countries around the world and collectively provide rich and diverse insight into the minds, motivations, and workflow of the modern developer.

In this report, we offer insights into what makes developers happy, what frustrations and challenges they face, and how these factors contribute to productivity. In Chapter 1, we present the background of the sampled developers to better contextualise the remaining analysis. In Chapter 2, we focus on developer productivity and happiness. We use questions derived from the Oxford Happiness Questionnaire to measure developer happiness in their current role and measure developer productivity using three of the DORA metrics. We examine the relationship between happiness and productivity and conclude that happiness in a developer role is associated with greater productivity.

Chapter 3 delves into developers' workflows and examines which aspects they enjoy and dislike. As expected, developers enjoy the development process and we break down the highlights and pain points of this activity. Finally, in Chapter 4, we dive into some of the challenges that developers face and offer insights into the root causes of common frustrations.

KEY INSIGHTS

- A developer that is 10% happier than another will need 10% less time to accomplish common programming tasks. →
- For each additional 500 employees a company has, developers working there are 1% less productive, on average. →
- Not being able to identify the root cause of an issue is most frequently identified as having the worst impact on developers' workflow; followed by not having a clear list of issues to prioritise and being unable to identify the impact of issues on end users. →
- When debugging, identifying the owner of a section of code is a hindrance to workflow, particularly in large enterprises. →
- 10% of developers identify communication as the primary obstacle in shipping code to production. →
- 69% of developers overall take pride in writing good code; followed by 65% of developers who feel pride in debugging code and/or improving software performance. →
- Two-thirds of all developers do at least some testing in the building phase, with more than one-third, 35%, primarily testing in this phase. →
- Surprisingly, developers are equally likely to report feeling pride in fixing bugs as they are when they improve an app's performance. →
- Developers who report that meetings have a detrimental effect on their productivity are more likely to work in teams that are 18% larger than those who do not identify meetings as obstacles. →
- When asked about their ideal schedule, developers wish that they could spend 19% less of their time being held up by internal messaging and processes. →
- 26% of developers identified internal processes or bureaucracy as an obstacle to their workflow. These developers work for companies that were 24% larger compared to developers who didn't identify internal processes as a major hindrance. →

PROFILE OF SURVEY RESPONDENTS

01

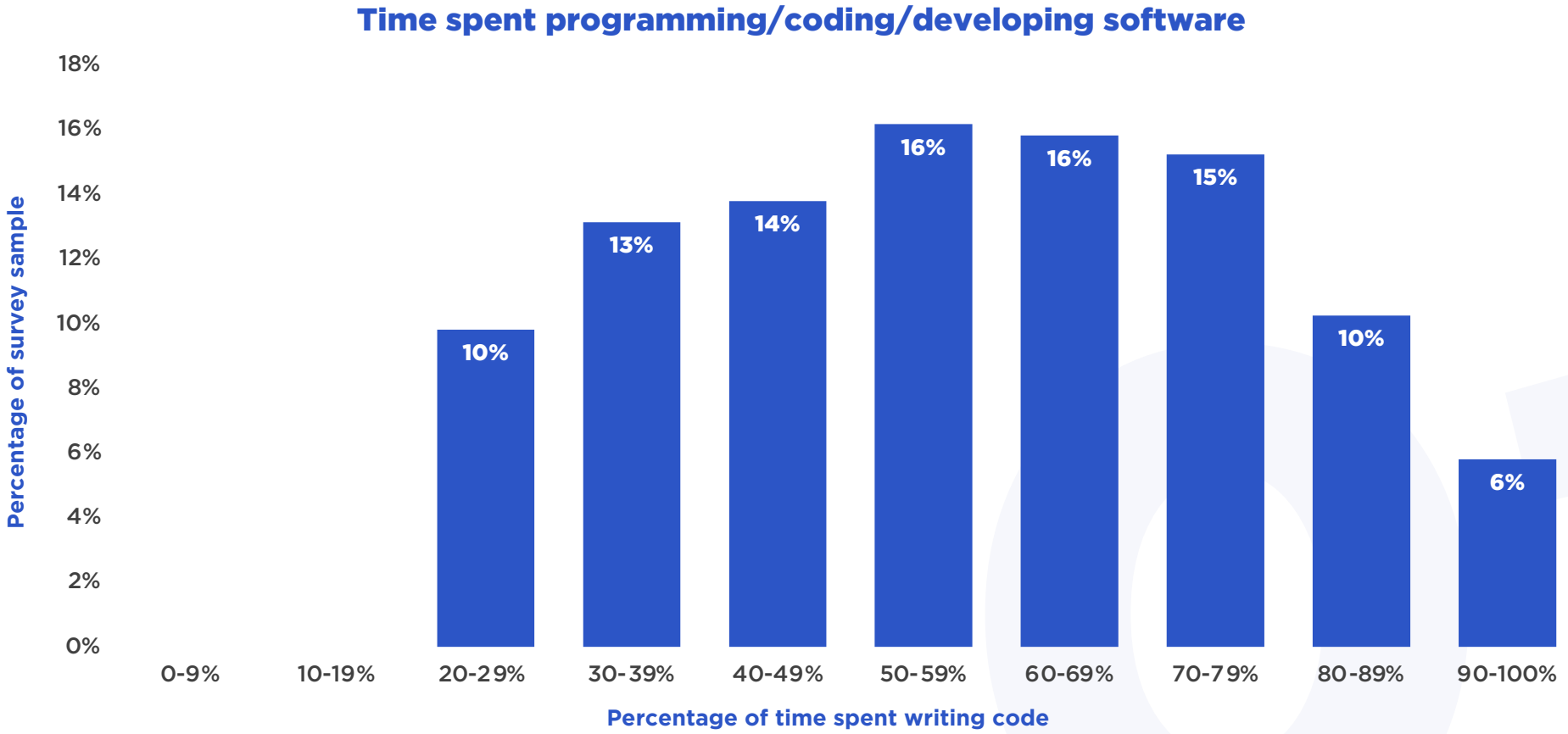
1. Profile of survey respondents

The survey respondents analysed in this report are professional developers who write software on a regular basis. All these developers have a live application or work for a company that has at least one. These stringent criteria were selected to ensure that those responding to the survey questions were properly equipped to offer meaningful responses to the posed questions.

As previously noted in the introduction, only developers that reported they spent at least 20% of their professional time writing software were allowed to participate in this survey. Of these developers, we see that almost half (47%) spend between 50-79% of their professional time developing software.

Almost half of the survey respondents spend between 50-79% of their time writing software

% of survey respondents (n=1,129)



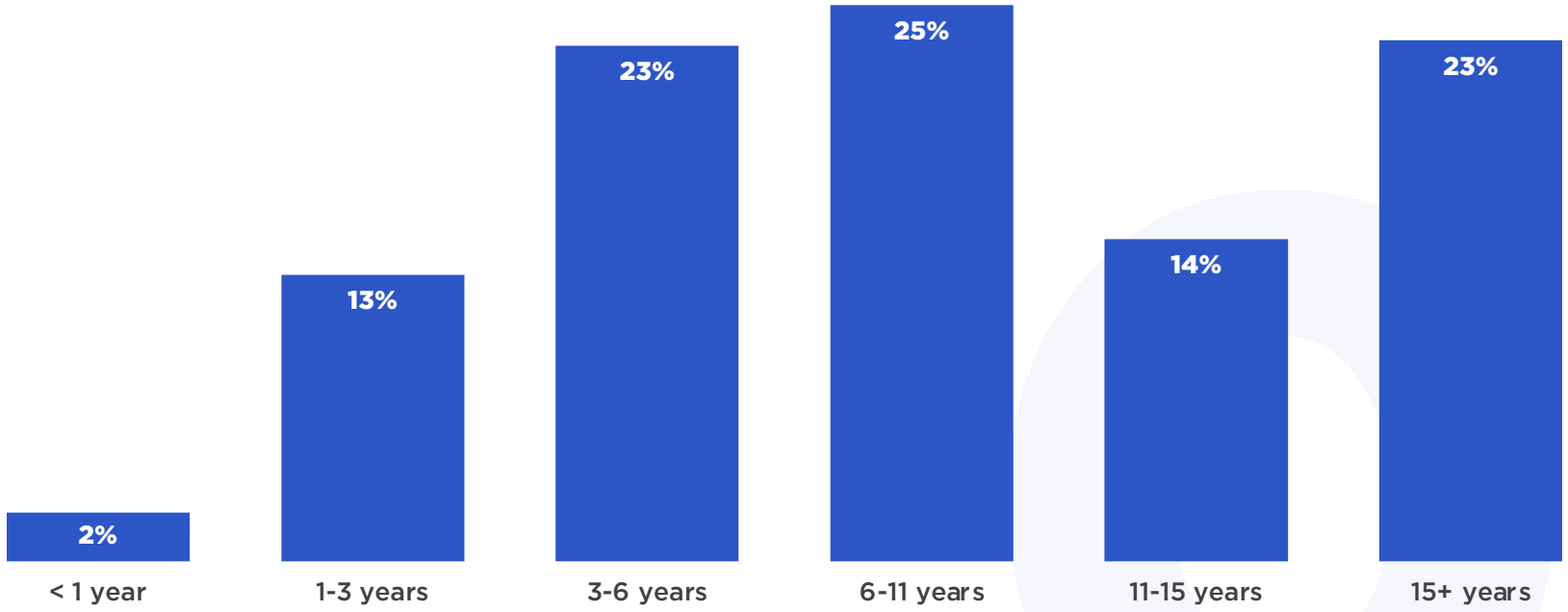
1. Profile of survey respondents

Another key metric that contextualises the survey respondents is their experience in software development. Likely due in part to the screening criteria of requiring them to have a live application, very few novice developers (2%) – those with less than one year of experience – participated in this survey. This intentional filtering for developers with more limited experience engenders that the average developer has developed software for 10 years and is well-equipped to offer robust insight into the software development process.

The majority of survey respondents (62%) have six or more years of experience developing software

% of developers (n=1,129)

Experience developing software



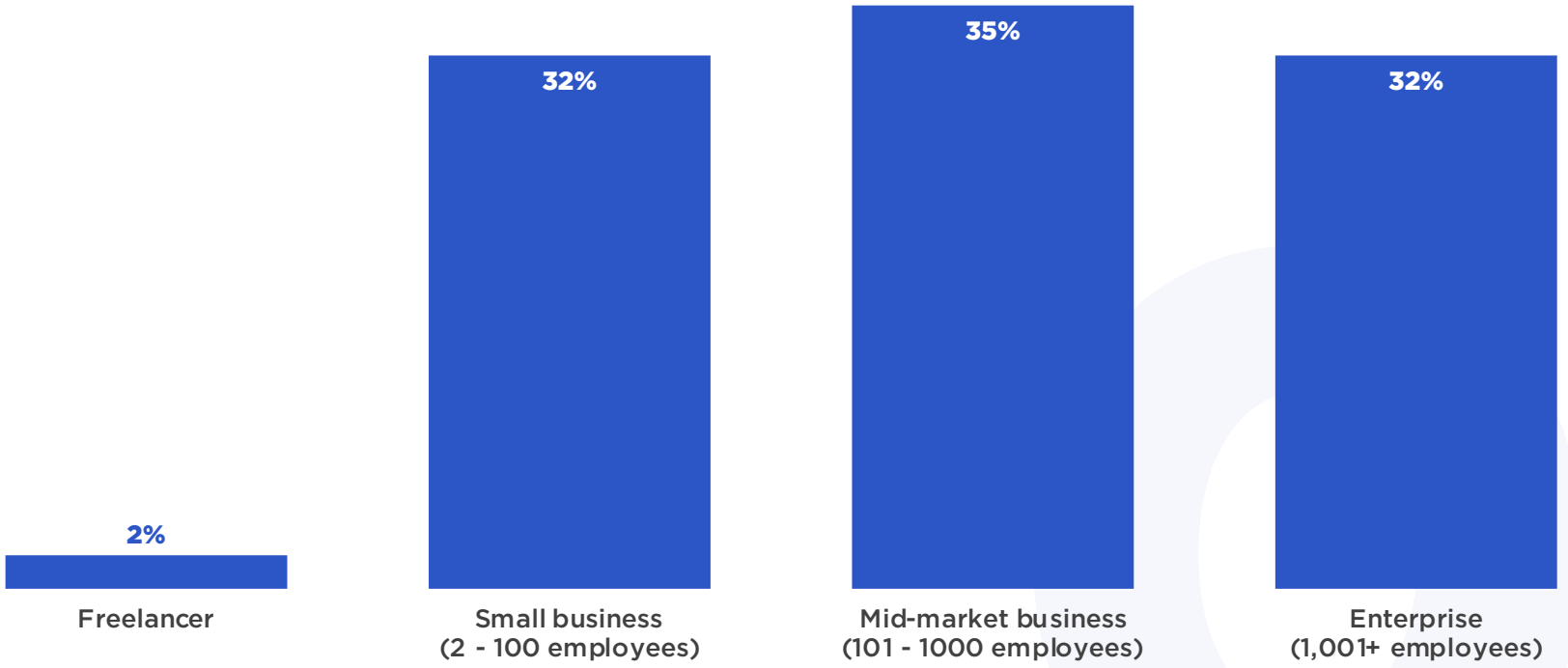
1. Profile of survey respondents

A final key metric that we will examine throughout this report is the size of the company that developers work at. Very few (2%) of developers who work as freelancers participated in this survey. An approximately even mix of developers working in small (32%), medium (35%), and large (32%) size businesses are represented in the analysis here. Throughout this report, when we segment by company size, freelancers will be excluded since there is an insufficient sample to reliably report results here.

There is an even mix of developers working in small, medium-sized, and large companies

% of survey respondents (n=1,129)

Size of the firm where developers are working



DEVELOPER HAPPINESS AND PRODUCTIVITY

02

Two crucial factors for any software company to consider when making decisions that will affect their employees are developer happiness and productivity. A nuanced understanding of what drives happiness and productivity is key for companies when selecting tools, designing working environments, and implementing policies and procedures. In this survey, we measured both happiness and experience and explored the factors associated with each construct in this section.

Happiness in developers' current role

We asked four questions that capture various aspects of developers' satisfaction in their current roles. To measure happiness, we asked respondents; on a scale of 1-5, to what extent they agreed or disagreed with the following statements. Their responses were then combined to get a measure of a developer's overall occupational happiness.

- I am fulfilled in my current occupational role.
- I am happy with my current role.
- I am satisfied with the impact I have on my company.
- I look forward to going to work each morning.

What makes developers happy?

Neither company size nor the number of software development colleagues a developer works with has a significant impact on a developer's happiness. Furthermore, developers across all experience levels are equally likely to be happy. Interestingly, developers who spend more time dealing with infrastructure issues report being happier. For every 10 additional hours a developer spends on resolving infrastructure issues a week, they report being, on average, 3% happier.

The increased happiness due to working on infrastructure issues is likely related to the developer's role in the company. If a developer reports being a manager or having 'chief' in their title, they are more likely to work on infrastructure issues. Managers and chiefs are also, on average, 6% happier than non-managers. The inflated happiness that managers report is likely associated with a number of different factors; including both workflow (responsibilities, tasks, etc.) and non-workflow-related (pay, benefits, etc.) issues. In Chapters 3 and 4, we examine developer perceptions relating to their workflow; both in terms of their expectations and how they perceive their current working environment.

Productivity

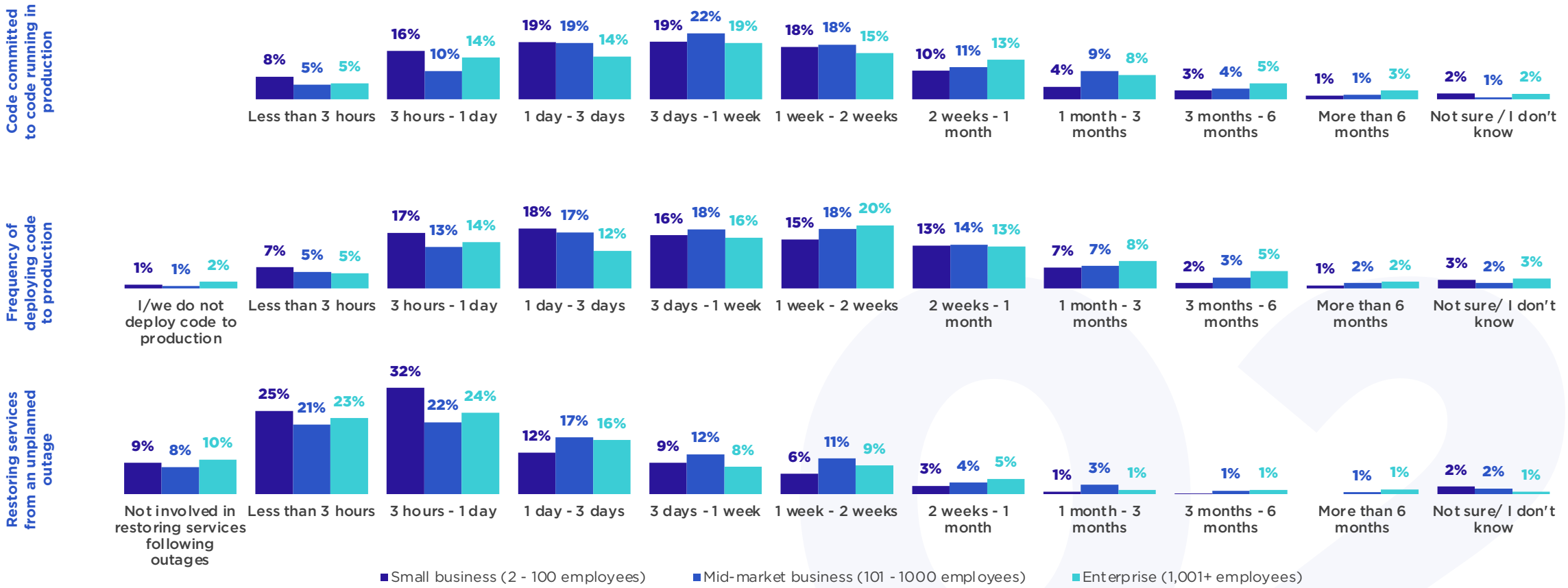
In this report, we combine three metrics that measure developer efficiency in terms of time needed to accomplish programming tasks and the frequency of shipping code to production to create an overall metric: productivity. The productivity metric is a combination of the three measurements listed below:

- The time it takes developers to go from the code being committed to in-production.
- The time it takes to restore service from an unplanned outage.
- The frequency at which developers deployed code to production.

In the graphs below, we report, segmented by company size, how long it takes developers to accomplish the three productivity metrics. We note that developers working at larger organisations report needing slightly more time to accomplish tasks compared to their colleagues working at smaller organisations. This is further supported by our regression model findings, reported in the subsequent section. Later in this report, we delve into possible explanations for why developers working at larger firms are less likely to be as productive.

Restoring outages is a clear priority, with 49% of developers reporting that they and their team do so in a day or less

% of developers (n=1,107)



The drivers and barriers of developer productivity

Above, we presented three productivity metrics separately. To gain insight into overall productivity, we combined the frequency and time metrics into a single variable that measures productivity in terms of hours. We then ran a regression model using the joint developer productivity metric described above, and examined the relationship between developer happiness and productivity as well as other drivers and barriers. The findings in this section take into account the differences across geographical regions.

Happy developers are productive developers. A developer that is 10% happier in their role will require 10% fewer hours to accomplish the three productivity tasks mentioned above. We also find that experience plays a crucial role. Each year of experience a developer gains in software development results in a 6% increase in productivity. These two factors both play significant roles in driving developer productivity.

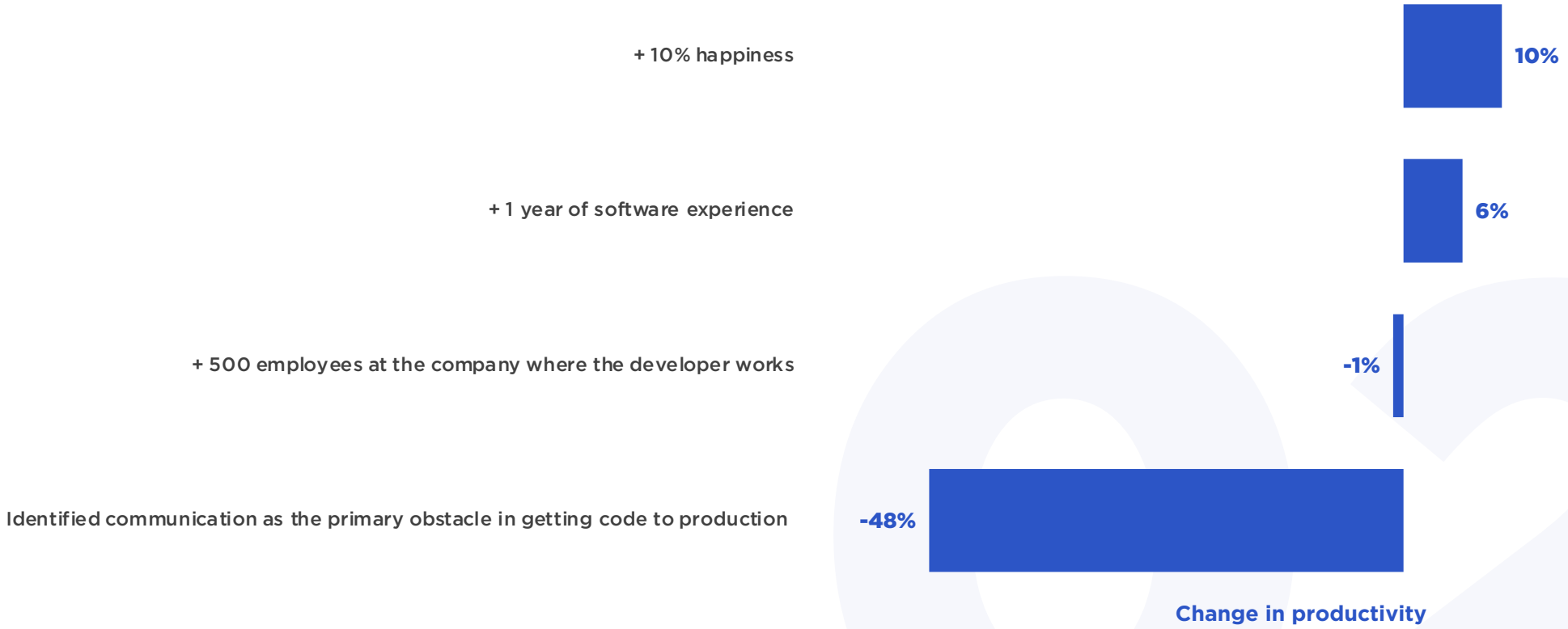
When considering barriers to productivity, we note two key factors that stunt a developer's efficiency. The larger the company developers work for, the more likely they are to report lower productivity. Specifically, for every 500 additional employees a company has, developers working there are likely to be 1% less productive. As we will see in Chapter 4, two of the largest obstacles that prevent developers from effectively doing their jobs are internal processes and bureaucracy, features typically associated with larger organisations. Hence, it is especially critical for larger organisations to optimise workflow wherever possible.

A final key factor that significantly affects developers and their teams' productivity is communication. If developers identify team communication as the primary obstacle in getting from the committed code stage to the code in the production stage, they are likely to be 48% less productive than developers who do not identify this as the primary obstacle. Only 10% of developers in this survey identify communication as the primary obstacle; those who did, however, need significantly more time to accomplish tasks in their workflow compared to developers who did not.

Happier developers are generally more productive

% of change in developer productivity (n=949)

Drivers and barriers of developer productivity



INSIGHTS INTO DEVELOPERS' WORKFLOW

03

Developers' workload – preferred vs reality

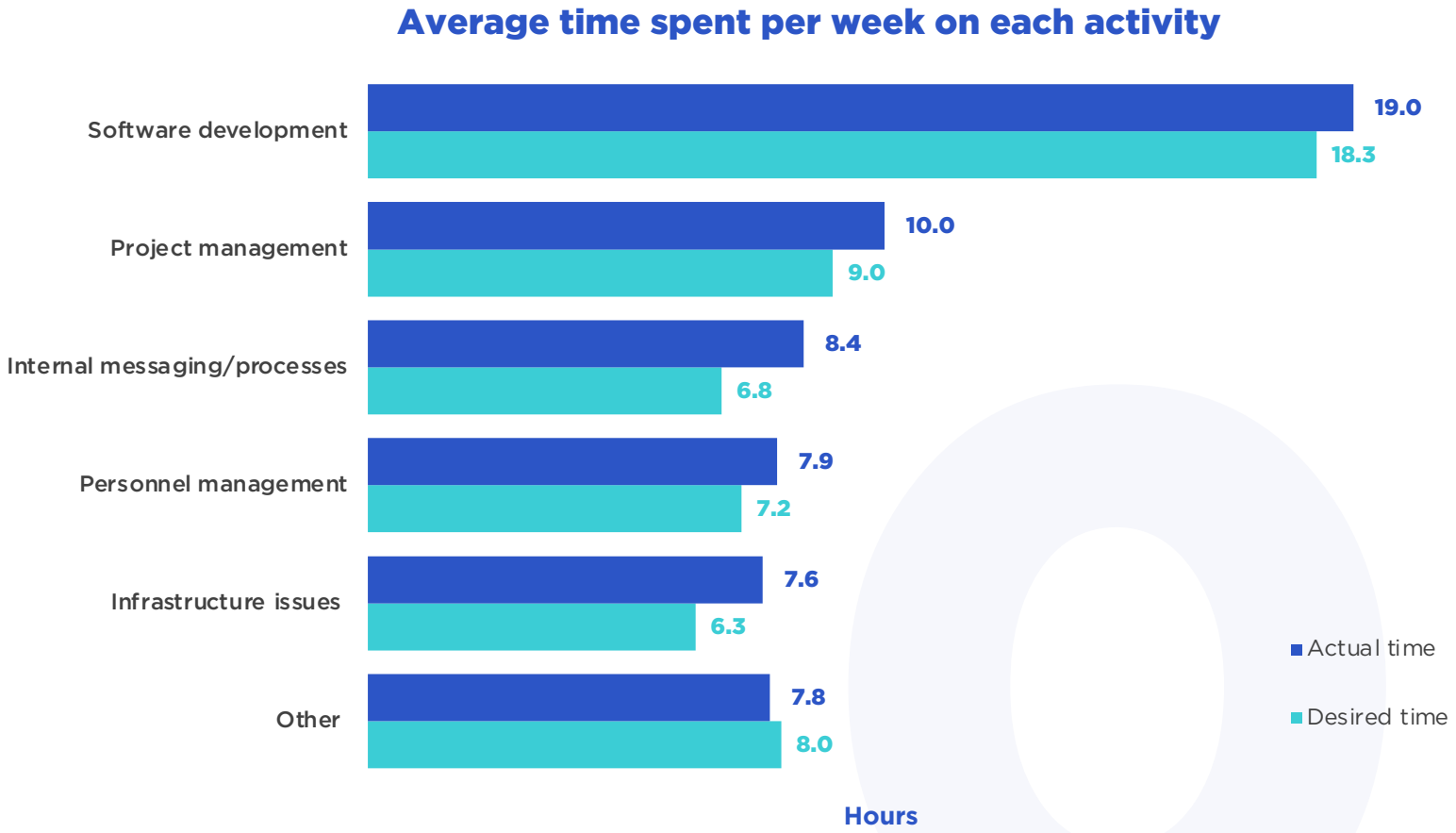
Following the analysis of productivity, in this chapter, we examine how developers spend their time to better understand what their working life consists of and how they wish it looked. We asked developers to first report how their week looked and then how they wished it looked.

We find that developers, on average, spend the largest proportion of their time on software development, followed by project management. They report spending on average 19 and 10 hours, respectively, on these tasks per week (31% and 16% of their total reported weekly time, respectively). We see that, in general, developers want to keep doing these two tasks as they are the two leading components of what developers wished their week consisted of as well. Additionally, the more time developers spend developing software, the happier they are; emphasising a crucial, but occasionally-overlooked principle – developers want to develop software.

The largest difference we see between what developers wished their week looked like vs their actual week is in dealing with internal messaging and processes and infrastructure issues. Developers wish that they could spend 19% and 17%, respectively, less of their time being held up by these time-sucks; highlighting the need for efficient and effective tools for communications and workflow across all aspects of business.

Developers want to spend the greatest proportion of their time developing software and managing projects

Number of hours per week (n=1,123)



Software development

Next, we break down the activity where developers spend most of their time: software development. We asked developers about the software development stages they spend the majority of their time on, as well as about the stages they enjoy the most.

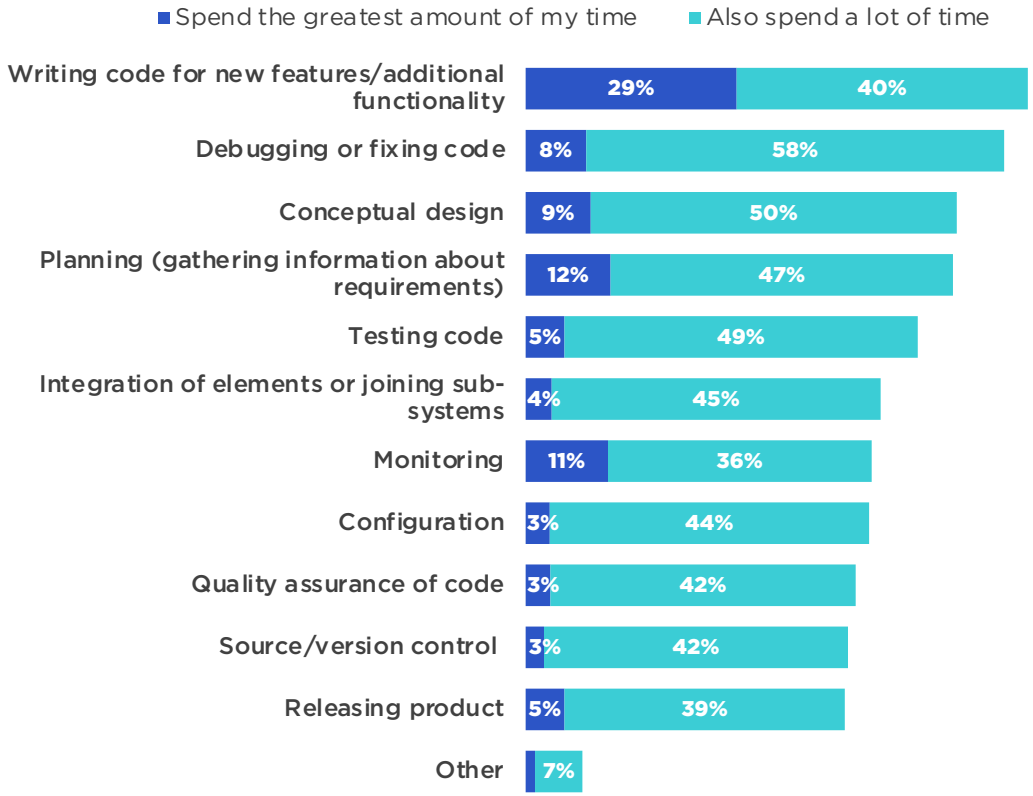
Writing code naturally takes up the greatest percentage of developers' software development time. 29% of developers report spending the greatest amount of time on this activity and 69% report that they spend a lot of their overall time doing so. The conceptual design phase is also a time-consuming activity that developers typically enjoy. 60% of developers report spending the majority of their time or a lot of their time on this phase, with an equal percentage reporting that they enjoy it as well.

The second most frequently reported activity to be consuming a lot of time is debugging or fixing code. Two-thirds (67%) of developers report spending the majority or a lot of time on this task. Developers are less likely to report enjoying this phase, however (51%). Debugging in particular can create major challenges in a developer's workflow and has a detrimental impact on productivity. As we will see later in Chapter 4, identifying the root causes of the issue while debugging code is identified as a major hindrance in this process.

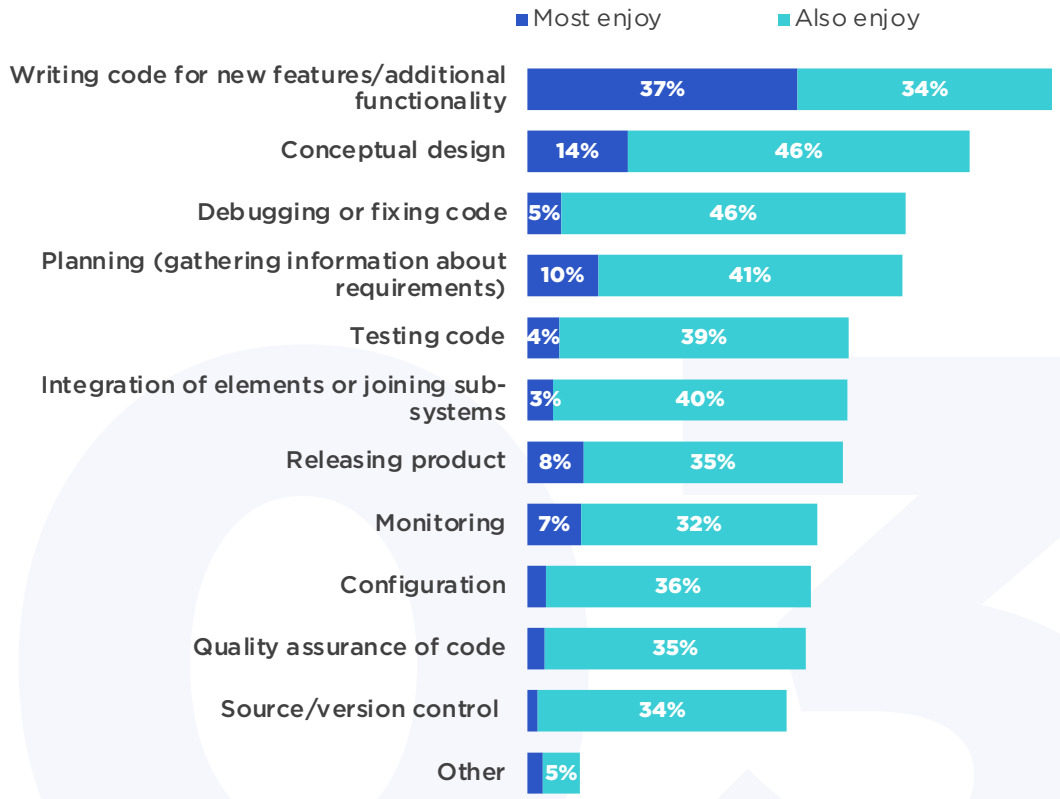
Developers enjoy writing code, but dislike quality assurance and source control

% of developers (n=1,124)

Time spent at software development stages



Software development stages developers enjoy



How developers think about their own tasks

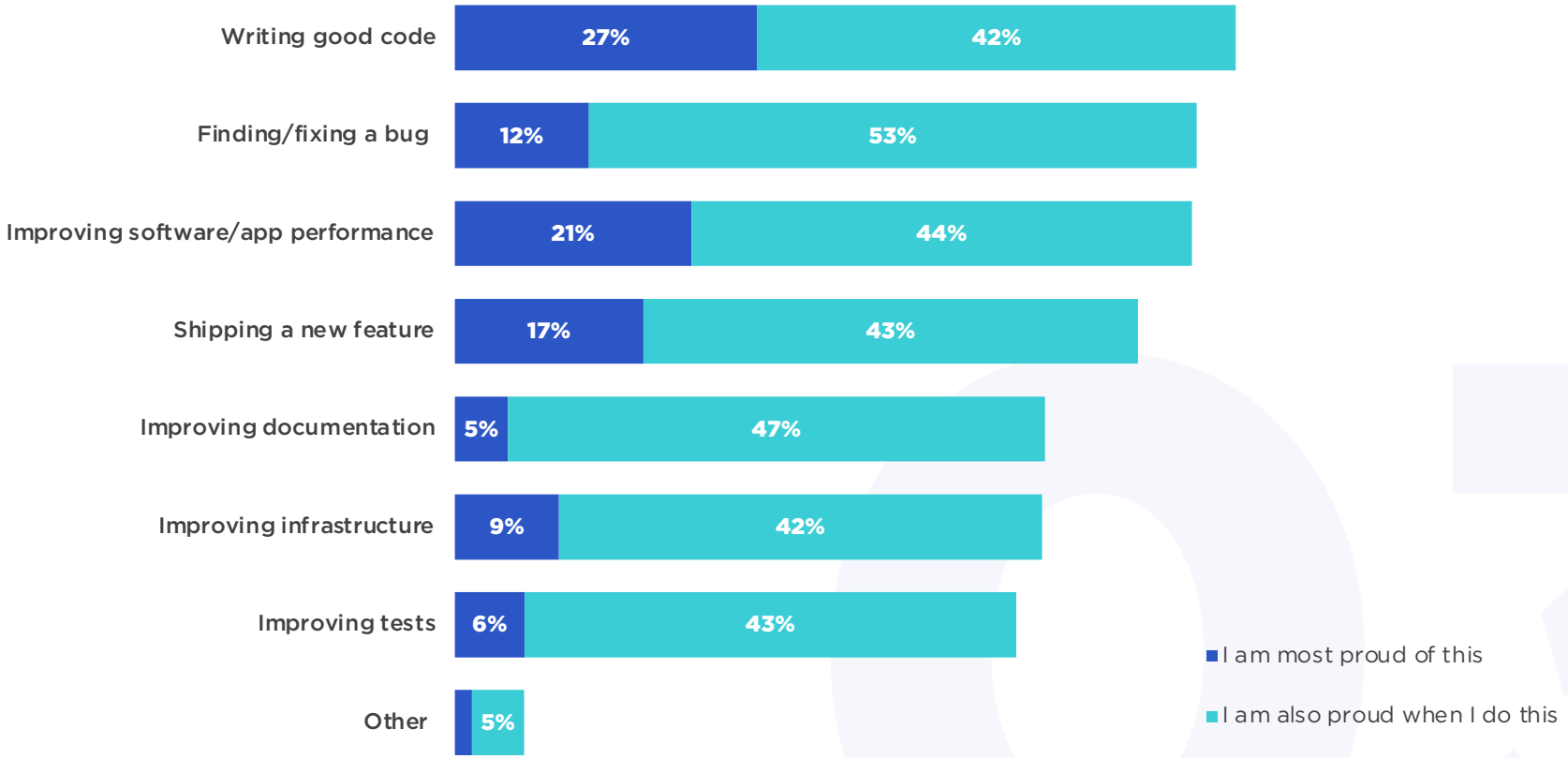
To understand how developers think of the work they do, we asked them what software development activities they are most proud of accomplishing. Writing good code is the leading source of pride for developers with 27% of developers reporting that they are most proud of this and 69% of developers overall taking pride in this task.

Finding and fixing bugs and improving software/app performance are tasks that additionally make developers feel good; with 65% of developers reporting that they take pride in these two tasks. 21% of developers report that they are most proud of improving software/app performance compared with 12% who report the same for debugging code. As a crucial part of the software development process, in the next chapter, we examine the challenges associated with the debugging process.

69% of developers feel proud of writing good code

% of developers (n=1,100)

Pride in accomplished tasks



DEVELOPER CHALLENGES

04

4. Developer challenges

In this final chapter, we start small and zoom out while examining the main challenges that software developers face. We initially look at the challenges developers encounter debugging code and then pan out to understand at what stages they are testing for these errors. Next, we expand our focus to assess the root causes of the issues developers report facing and examine factors that are associated with common issues. Finally, we conclude broadly by analysing general roadblocks developers face in their workflow and we evaluate the root causes of these issues. All issues – from small to large – can have compounding effects on developer productivity and product quality.

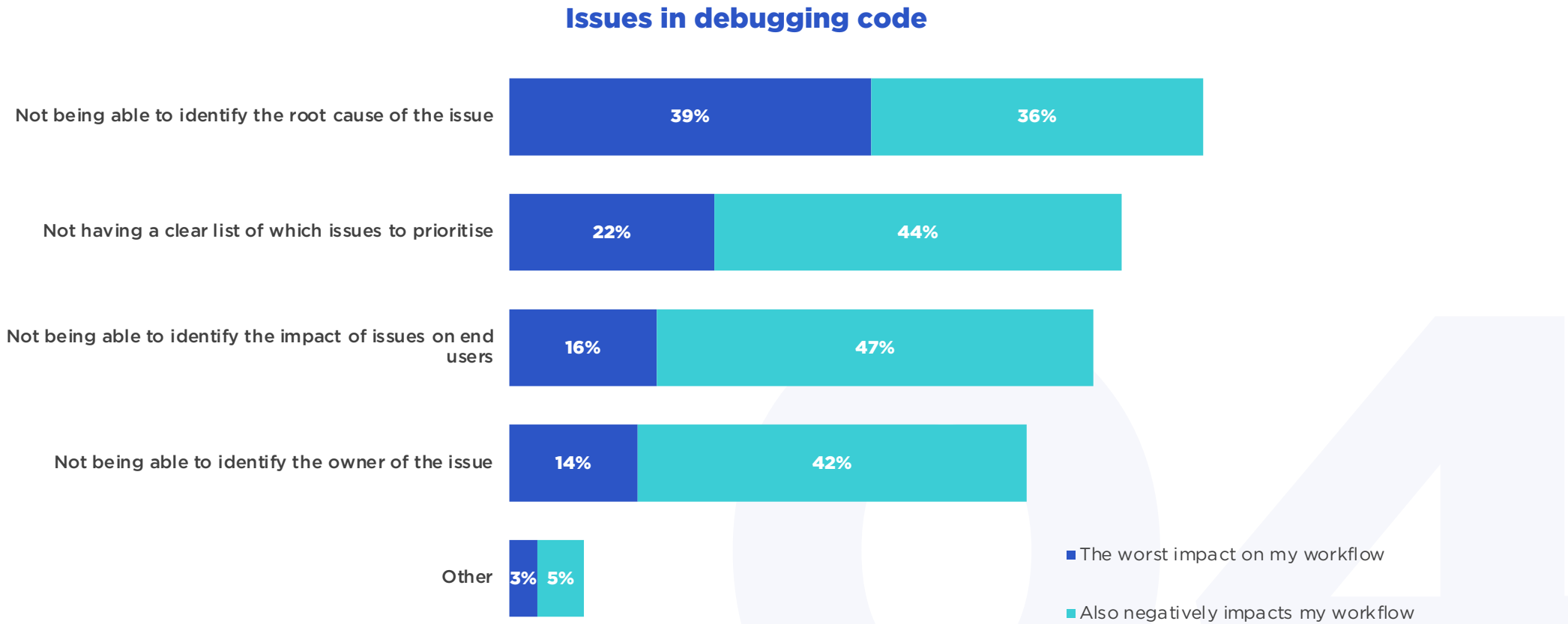
Debugging challenges

We asked developers to identify one issue that significantly impacted their workflow while they were debugging code. They could additionally select as many other issues that also had a negative impact. We break down the process of debugging code and see that not being able to identify the root cause of the issue is the cause of the largest impact on developers' workflow. Not having a clear list of issues to prioritise comes second, while being unable to identify the impact of issues on end users is the third most frequently mentioned negative impact on a developer's workflow.

Understandably, as the number of individuals working on software development in an organisation increases, so does the likelihood that developers select not being able to identify the root cause of the issue as the worst impact on their productivity. On average, developers who select this option as the item that has the worst impact on productivity work in companies with 22 more individuals working in software development. As companies' software developers grow, development processes often increase in complexity. Being able to efficiently identify the root causes of issues in software and applications is crucial for maintaining a productive work environment. This notion is supported by the graph in the subsequent section that breaks down issues by company size.

Not being able to identify the root cause of an issue is most frequently identified as having the worst impact on developers' workflow

% of developers (n=1,100)



4. Developer challenges

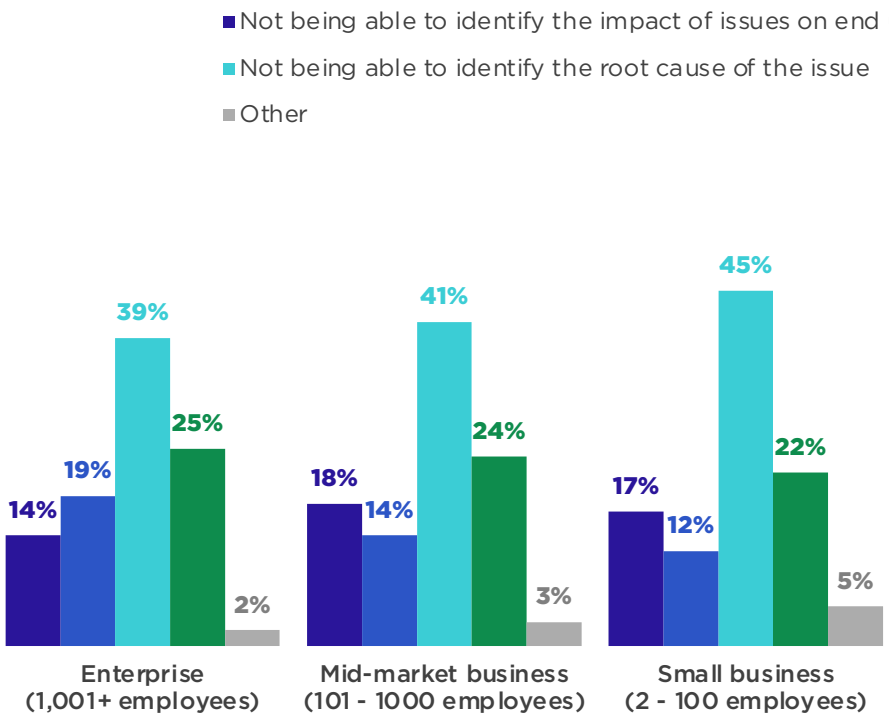
In breaking down the issues developers face while debugging code by company size, we gain additional insight into the workflow and challenges of developers. While developers working for all sizes of firms report that not being able to identify the root cause of the issue has the greatest negative impact on their workflow across all firm sizes, developers working at small businesses are 6 percentage points more likely to do so than those working at large enterprises.

Conversely, we note that developers at large enterprises are 7 percentage points more likely to report that not being able to identify the owner of an issue negatively impacts their workflow, compared to developers working at small businesses. An effective system for tracking code ownership is therefore crucial in larger firms to overcome this challenge.

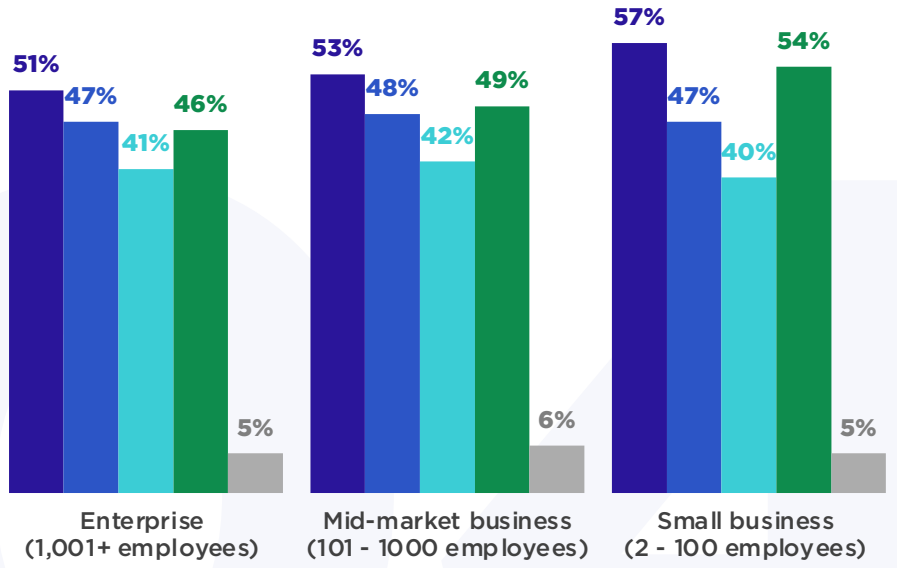
Developers at larger enterprises are more likely to report not being able to identify the issue owner as a key hindrance when debugging code

% of developers (n=1,100)

Impact to workflow while debugging code



Worst impact



Also negative

Testing code

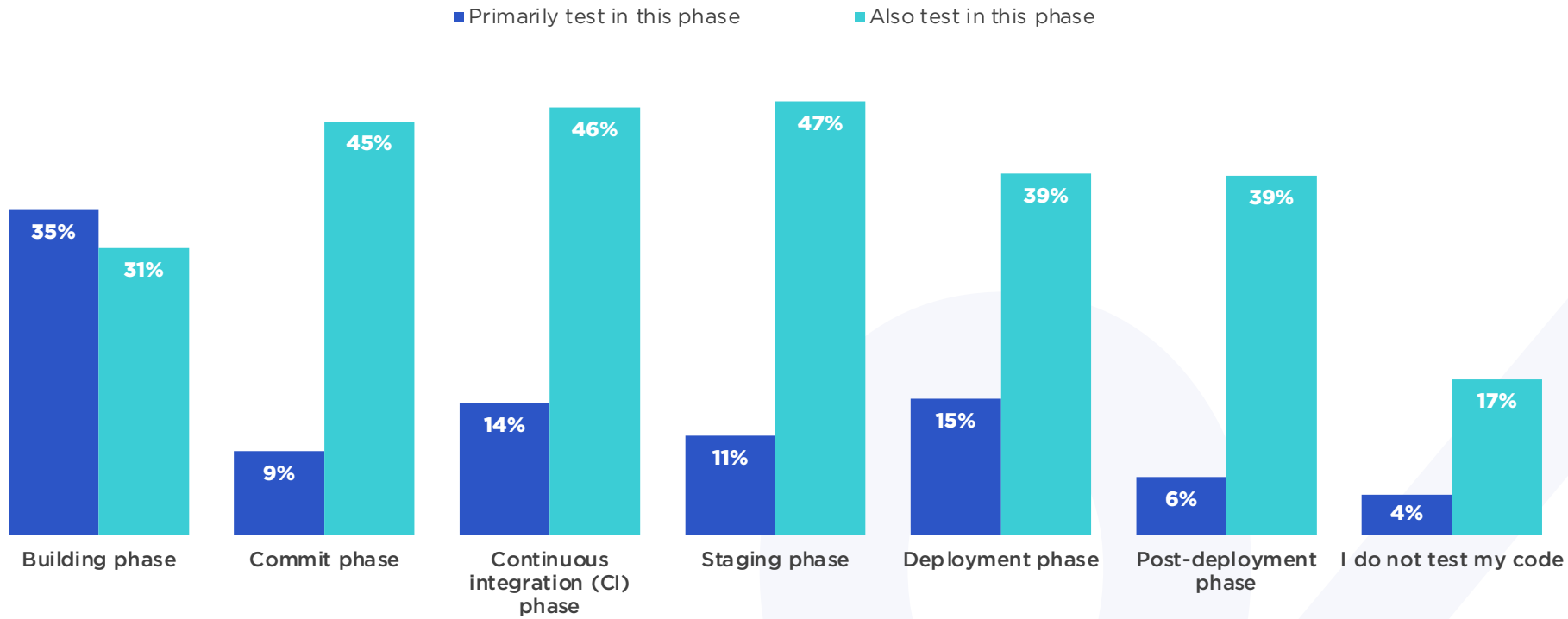
While we now have a better understanding of the challenges developers face debugging issues, we zoom out in the process to assess when developers are testing their code. We see that two-thirds of all developers do at least some testing in the building phase, with more than one-third, 35%, primarily testing in this phase. Testing in this stage is frequently undertaken by developers with significant experience. 65% of developers with 15+ years of experience test in this phase compared to ~50% of developers with 3-15 years of experience. In the lower category of developers of 1-3 years, only 40% test their code this early in the development lifecycle.

Shifting testing earlier in the development lifecycle, often referred to as “shift-left”, is recognised as good practice and experienced developers appear to appreciate this fact. This is further reinforced by the fact that developers who report not testing their code are also more likely to be less experienced, by an average of three years. Following the building phase, the second phase that developers primarily test in is the deployment phase; with 15% of developers primarily testing in this phase.

The majority of developers primarily test their code in the building phase

% of developers (n=1,100)

Which phases do developers spend time testing their code in?



Challenges with programming

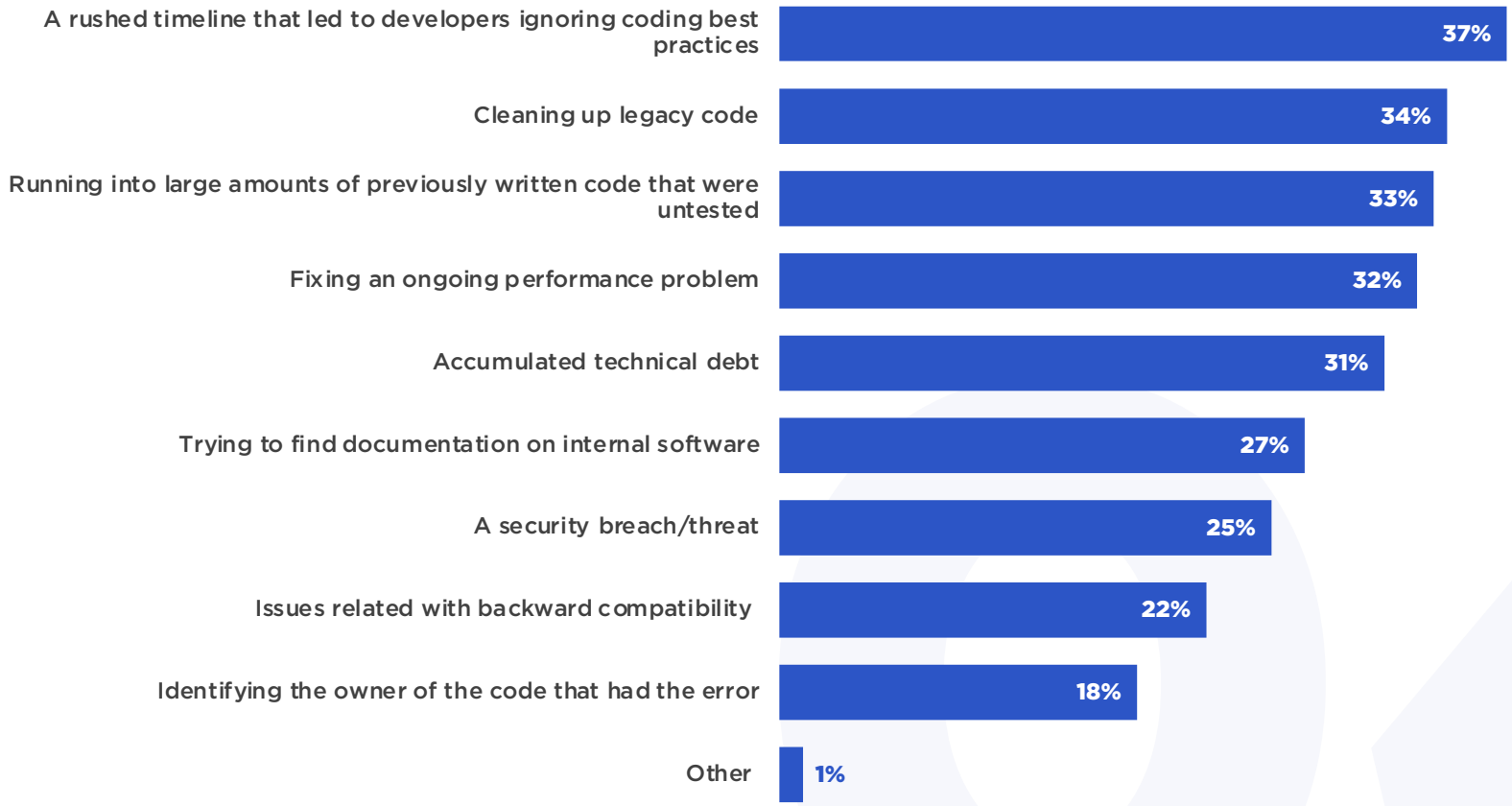
Continuing to broaden our perspective, we next asked developers to reflect on the largest programming problems/challenges they have faced and consider the source of the issue. 37% of developers report that a rushed timeline was the root cause of the largest issue they have faced. 45% of developers self-identified as programmers/software developers report a rushed timeline as a key challenge. This is 14 percentage points more than the CEOs and managers who report this as a key challenge (31%).

The second and third most commonly selected challenges were cleaning up legacy code (33% of developers selected this option) and running into untested code (32% selected this option). Reporting that cleaning up legacy code caused a large issue was significantly associated with an increased software development team size. Developers who selected cleaning up legacy code as a key issue averaged working on a 12% larger team. Running into untested code, however, was not associated with company size. This is likely due to the fact that larger teams have more personnel and resources to test code, making this less of a challenge.

Rushed timelines are the top root cause of programming challenges

% of developers (n=1,100)

Largest source of software developement issues



General working challenges

All professionals face obstacles that prevent them from working effectively. Some are unique to the professionals themselves, while others are associated with the working environment. Hurdles in developer workflow are commonly thought of in terms of software development, however, as we observed in Chapter 1, developers can be responsible for a myriad of tasks. Hence, here we delve into more general challenges that developers face in their day-to-day professional lives.

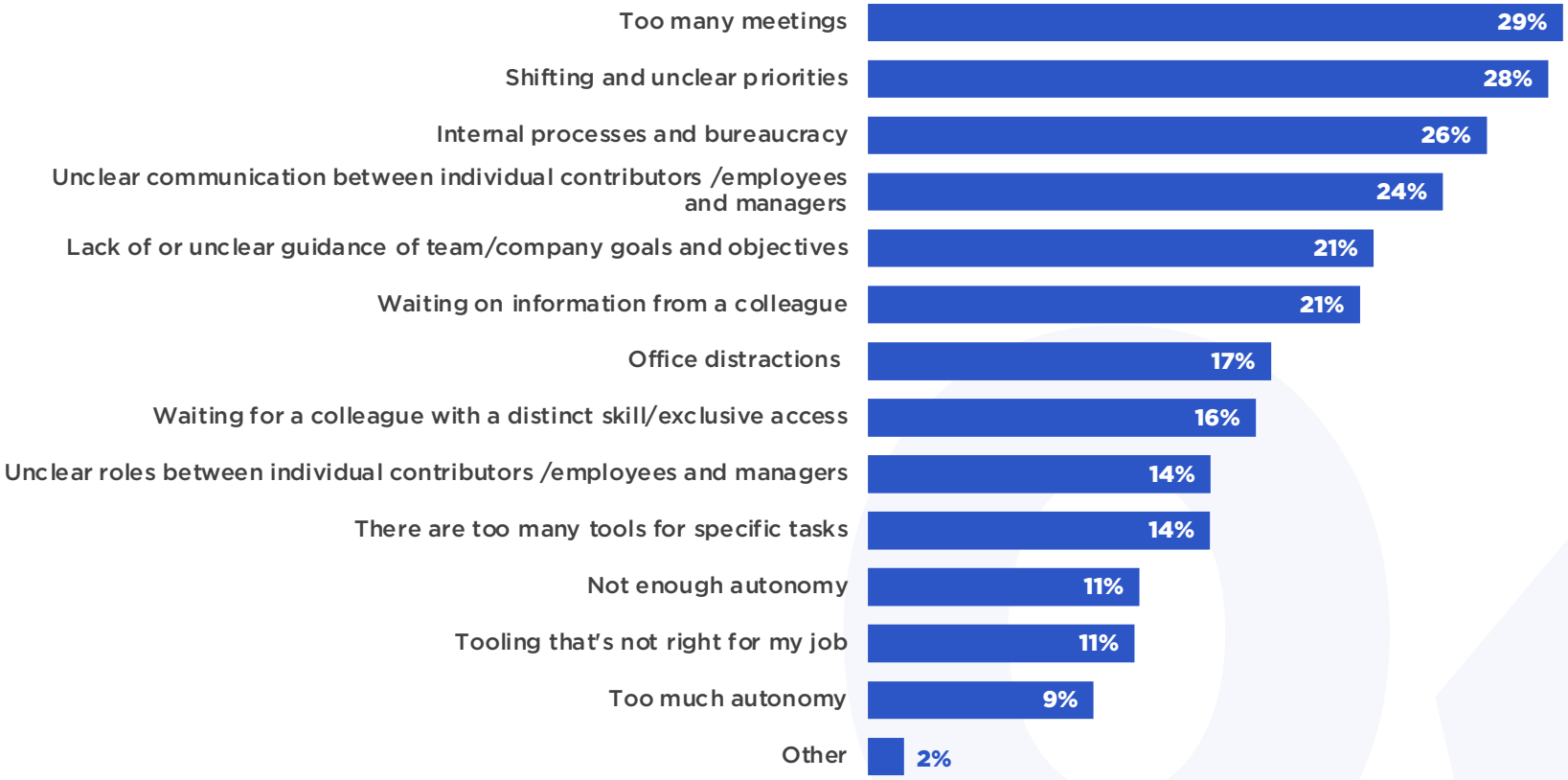
Outside of debugging and testing code, developers face various other hindrances to their productivity. The most prominent obstacle that developers deal with is too many meetings. Developers who report that meetings have a detrimental effect on their productivity are more likely to work in software teams that are 18% larger than those who do not identify meetings as obstacles. Larger teams are harder to coordinate effectively and some companies fall into the trap that more meetings are the solution.

Meanwhile, 26% of developers identified internal processes or bureaucracy as an obstacle to their workflow and, on average, the companies they work for are 24% larger (by the number of employees) than developers who did not identify this as a key challenge.

Too many meetings and shifting priorities thwart developer productivity

% of developers (n=1,100)

Largest obstacles that prevent developers from doing their jobs efficiently



4. Developer challenges

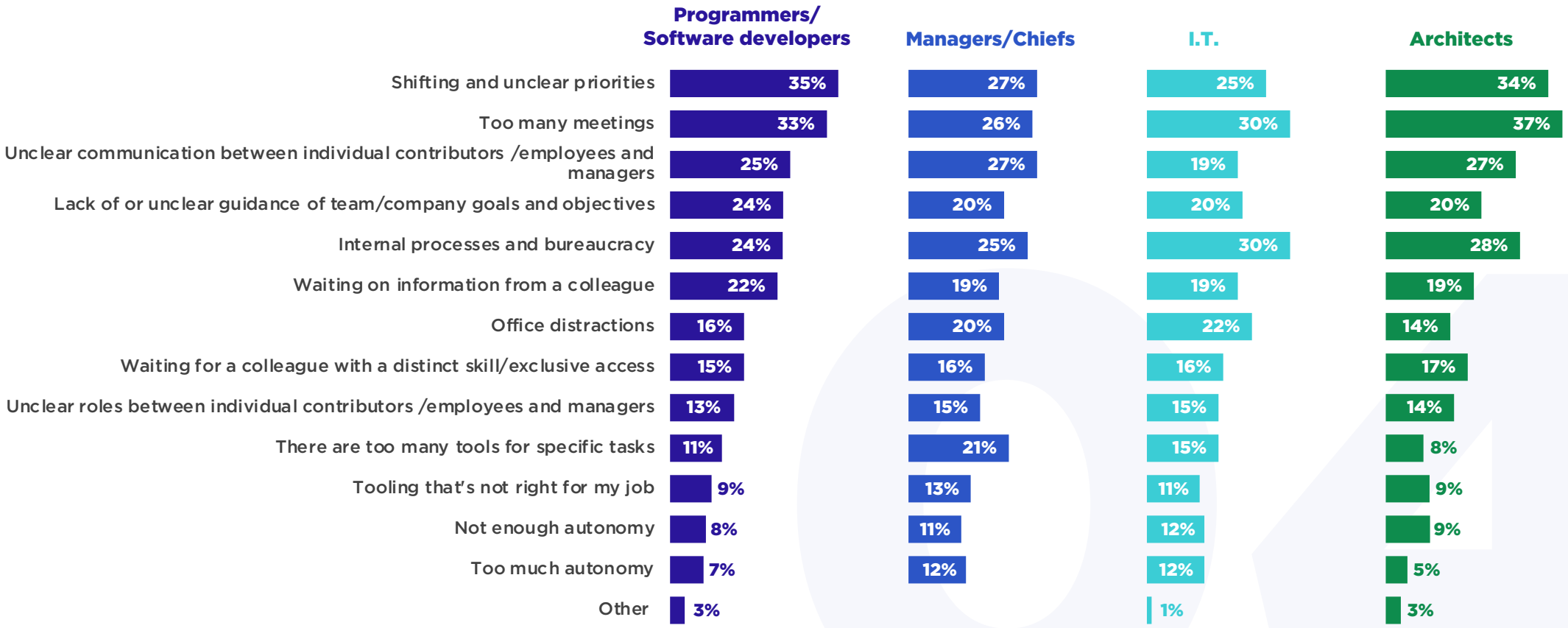
Developers work in various capacities in any organisation. To better understand the nuances of several positions, we break down the challenges by prominent roles: those working in management or as chiefs, programmers/software developers, architects, and those working in IT.

Shifting and unclear priorities rank in the top-three obstacles across all roles but are identified as the top obstacle by programmers/software developers and managers/chiefs. Too many meetings as noted above, is a prominent obstacle across all roles, but especially for architects. Many of the top challenges reported across all roles are process related, highlighting the need for companies to use good policies and procedures to optimise their workflow and developer productivity.

Shifting and unclear priorities are especially challenging for programmers/software developers and managers/chiefs

% of developers (n=1,100)

Obstacles ranked by developers working as:



TRUSTED BY

the leading tech platforms

OUR CLIENTS

Created by

/DATA SENTRY