Survey report: Key open source usage trends in professional development teams

WHAT DO TEAMS VALUE MOST ABOUT OPEN SOURCE, WHAT ARE ITS BIGGEST DRAWBACKS, AND HOW MUCH TIME IS SPENT ON CODE MAINTENANCE?

April 2019

TIDELIFT

INTRODUCTION

The professional open source survey results we published last year highlighted the impressive reach of open source usage among professional developers. We discovered several interesting data points, including that over 90% of professional developers use open source in building their applications. We also discovered that open source maintainers, when paid, will work on the very same things professional developers want more of—including predictable new features and releases, responsive security fixes, and more.

In our latest survey, which we ran in November and December of last year, we set out to answer some of the follow-up questions that arose after we analyzed the earlier results. Over 300 developers responded to our survey, which dives deeper into how professional developers use open source today.

CONTENTS

EXECUTIVE SUMMARY

Professional development teams benefit immensely from the availability of open source packages. Our survey listed ten reasons to use open source—from *makes me more productive* to *increases application performance*—and asked developers to rate them from *not at all important* (0), to *extremely important* (10). None of the ten reasons we listed scored below 5—that is, none is seen as unimportant. Topping the list of open source benefits are *makes me more productive, speeds up development and deployment time,* and *reduces development costs*.

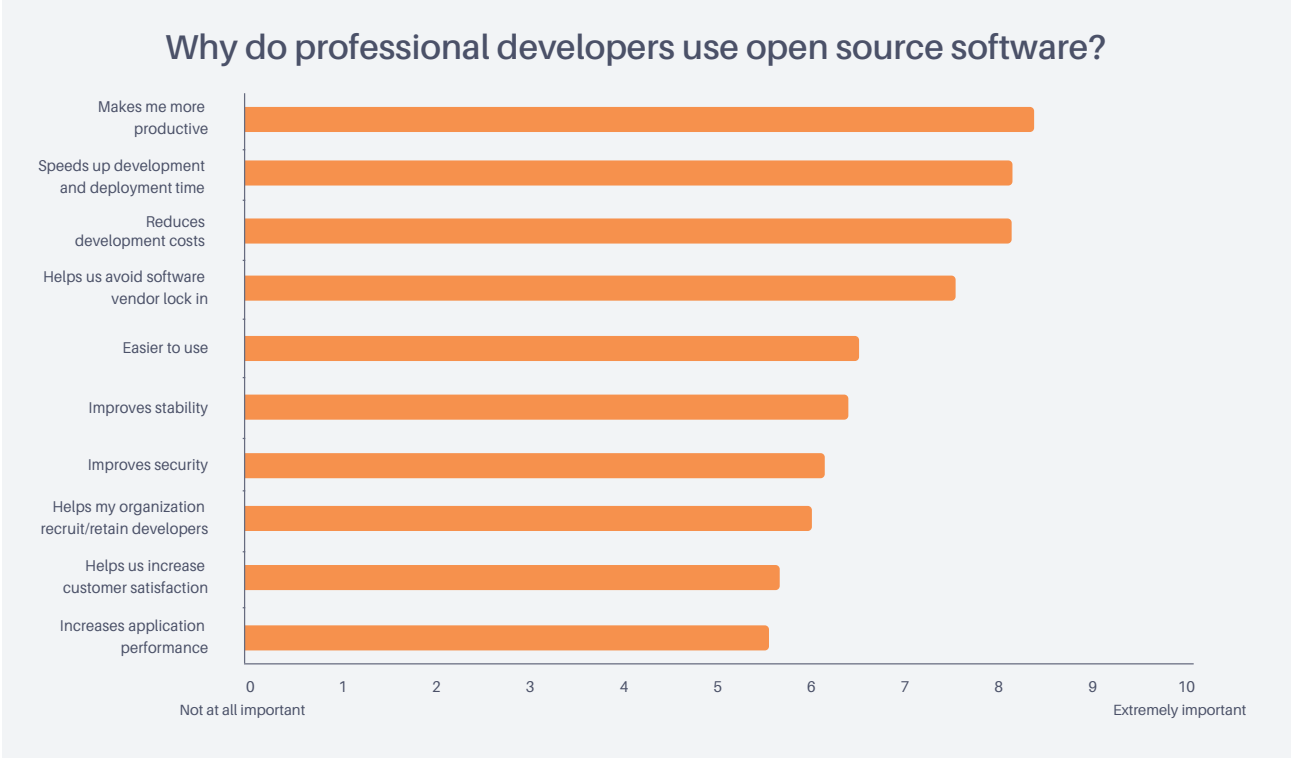
But concerns about using open source remain. *Avoiding security vulnerabilities, making safe bets on packages being maintained into the future,* and *staying current on packages they're using* are top of mind among survey respondents. Respondents in organizations with development teams of 1,000 or more people are significantly more concerned with *keeping track of the open source code in use* than are respondents from smaller dev shops.

Developers spend on average 30% of their time on code maintenance, with open source contributing about 25% of this workload. Developers on larger teams spend more time on code maintenance than do developers on smaller teams. And while open source represents a significant minority of code maintenance work for most developers, nearly 10% of respondents say most to all of their code maintenance relates to open source.

Finding #1: open source helps developers get more work done, more quickly and cost-effectively

When asked to rank the reasons why they use open source, professional developers name things that help them get more work done more quickly and cost-effectively. The top three benefits to using open source are:

1. Makes me more productive
2. Speeds up development and deployment time
3. Reduces development costs



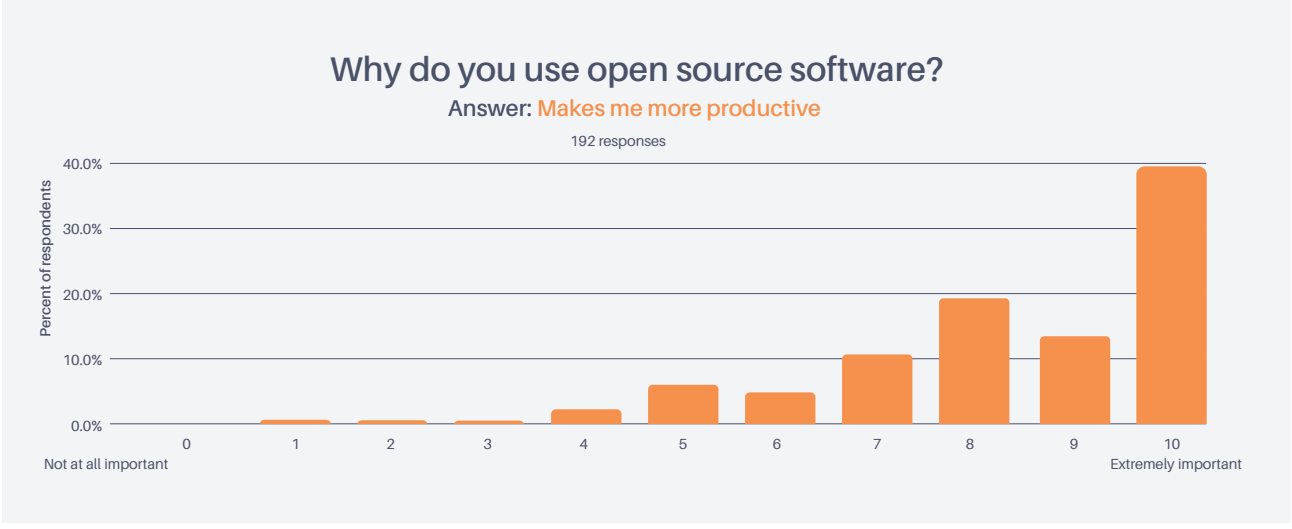
We examined all responses across geography, development team size, and role/title, and the results are almost completely consistent across these parameters. The one unsurprising exception is respondents with engineering manager and technology executive titles gave more weight to "helps my organization recruit/retain developers" than did software developers. One respondent added an exclamation point to this, saying: "open source is a talent on-boarding engine like none other."

We'll take a closer look at the top three benefits of using open source below.

BENEFIT 1: OPEN SOURCE MAKES ME MORE PRODUCTIVE

Survey respondent:
"If it isn't open source, I don't even bother unless I am forced to."

Developers rank the productivity boost open source gives them as its biggest benefit. For professional developers, whatever reduces the amount of time it takes to complete a task, or that results in more output—without hurting quality—increases productivity. Almost 40% of developers rated this benefit a "10" on a scale of 0-10.



Survey respondents' comments reveal how open source makes them more productive. Certainly the great functionality open source packages provide is important, as expected. Equally important is the efficiency gained by collaborating upstream and the ability to go beyond documentation to see the code and exactly how a package operates.

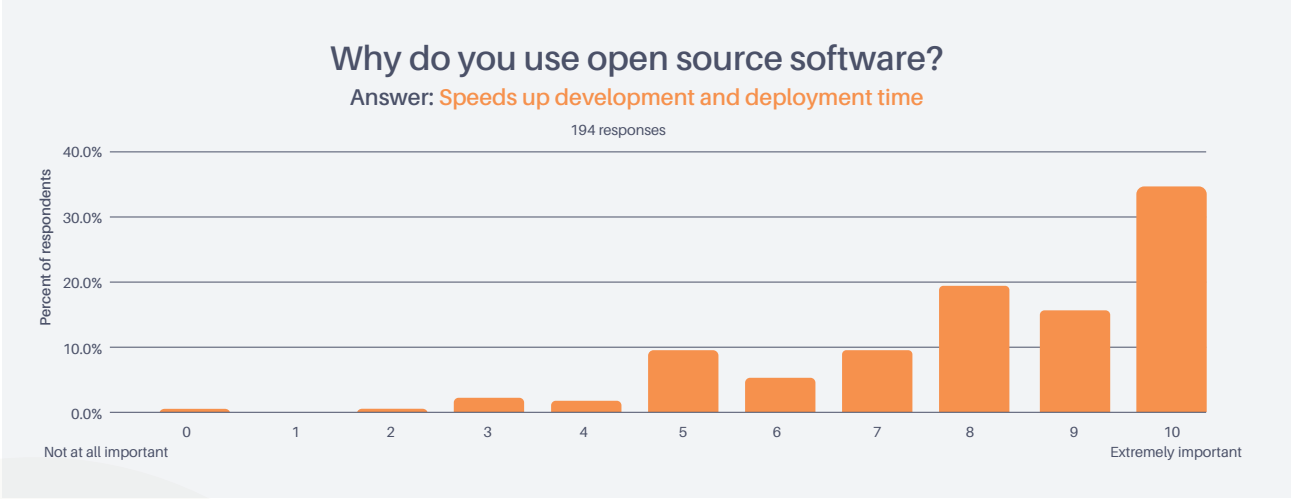
Here are a few representative comments to illustrate:

- "Open source provides an existing solution to avoid creat[ing] something from scratch"
- "Ability to issue fixes to bugs found, and fix them at source rather than downstream work arounds"
- "Ease of troubleshooting and fixing issues vs. proprietary frameworks"
- "Can read and understand the source code so that if the documentation is bad for an API call, I can trace what's happening myself"

BENEFIT 2: OPEN SOURCE SPEEDS UP DEVELOPMENT AND DEPLOYMENT TIME

Survey respondent:
"Open source helps/allows me to solve my clients problems quickly and with quality."

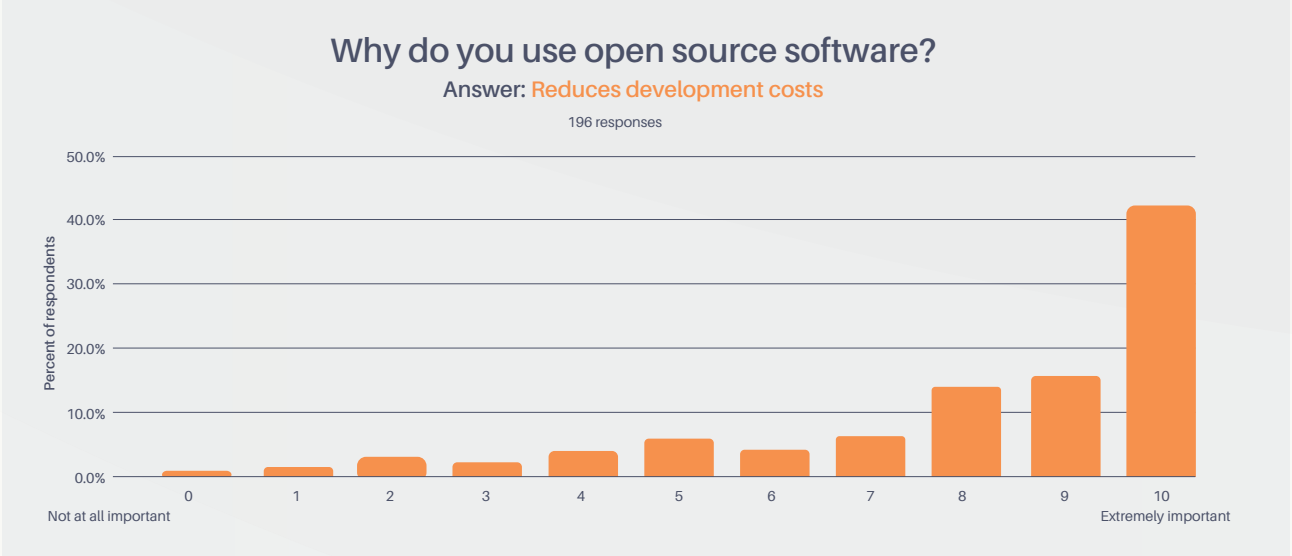
Accelerating development and deployment ranks as the second most important reason developers use open source. Productivity and velocity are related concepts, though not identical. Where developer productivity addresses the output per developer, velocity more typically applies to the speed of a team working together.



Interesting, less developers rated this a "10" on the 0-10 scale than either "open source makes me more productive" or the #3 response, "open source reduces development cost," but overall responses were still heavily weighted toward 7-10 ratings.

BENEFIT 3: OPEN SOURCE REDUCES DEVELOPMENT COST

On its face, the third most popular benefit of using open source—reducing development costs—requires little explanation. But when we dug into respondents' comments, we saw multiple interpretations of what this means.



For example, one respondent said, "As a non-profit tech organization that ships our own open source-powered products, we would be remiss to use expensive, vendor-lock in proprietary software. Hence, we love the open source software we use! We are better able to utilize our limited resources to achieve our important mission by building upon many open source libraries, operating systems and programs."

Yet another respondent saw a reduction in development costs come from sharing development responsibilities with a community: "[Open source] enables us to improve the components to better suit our needs. Providing these improvements back to the community lowers cost of maintaining them as custom additions to proprietary software."

Interestingly, almost 43% of respondents rated this benefit a "10" on the scale of 0-10, which was the highest "10" rating of any benefit we tested.

OTHER IMPORTANT REASONS WHY DEVELOPERS USE OPEN SOURCE

None of the options we provided scored less than 5 on a scale of 0-10—that is to say, none of the options we provided are unimportant to developers. Among the other benefits that our survey shows really matter to developers are helping avoid vendor lock-in, which was rated the 4th highest benefit. Ease of use, improved stability, and improved security all grouped fairly tightly together in the 5th through 7th spots.

We also gave respondents an opportunity to write in benefits that we might have missed, and many of them noted traits like collaboration, sharing, freedom, transparency, and community as important reasons why they use open source.

Here are some representative thoughts in respondents' own words, which we've categorized into three additional themes (some comments cross categories):

Community

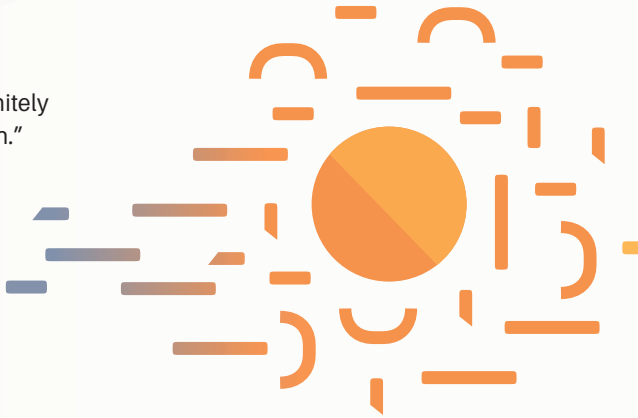
- "We like the idea of a community of coders that build software for everyone's benefit."
- "Trust, networking (community)."
- "Ethics. Collaboration and cooperation, rather than roadblocks, helps the world progress."

Principle

- "We value open source on principle. Software freedom. (That's arguably related to avoiding vendor lock-in, but it's important to me in and of itself.)"
- "I believe in open source/open content from an IP philosophical perspective."
- "I believe in the moral mission of open source software."

Culture and professional growth

- "Creates internal incentives for transparent and ethical behavior, and framework for accountability (i.e., helps build internal culture). Helps leverage and build ties to community beyond the organization. Promotes commons culture broadly. Attracts problem solvers to consider questions that are important beyond our organization."
- "Showing good code that works, for others to say 'wow' —will definitely make you happy! So open source will always be the tool for growth."
- "Helps everyone to make better code."

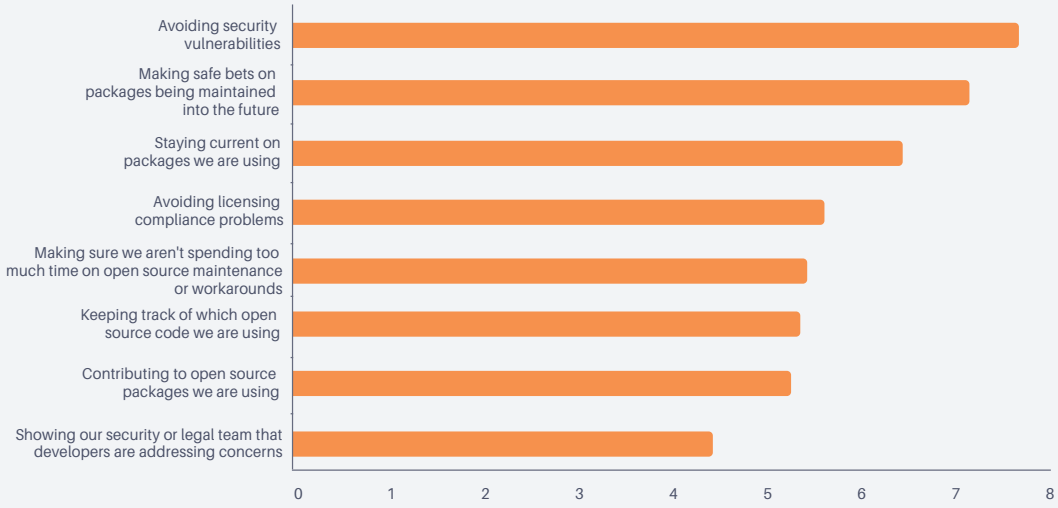


Finding #2: concerns with using open source remain

Notwithstanding developers' love of open source, there remain a number of areas where it can be improved. These three areas of concern stood out from the others among developers we surveyed:

1. Avoiding security vulnerabilities
2. Making safe bets on packages being maintained into the future
3. Staying current on packages they're using

Which of the following concern you most about using open source today?



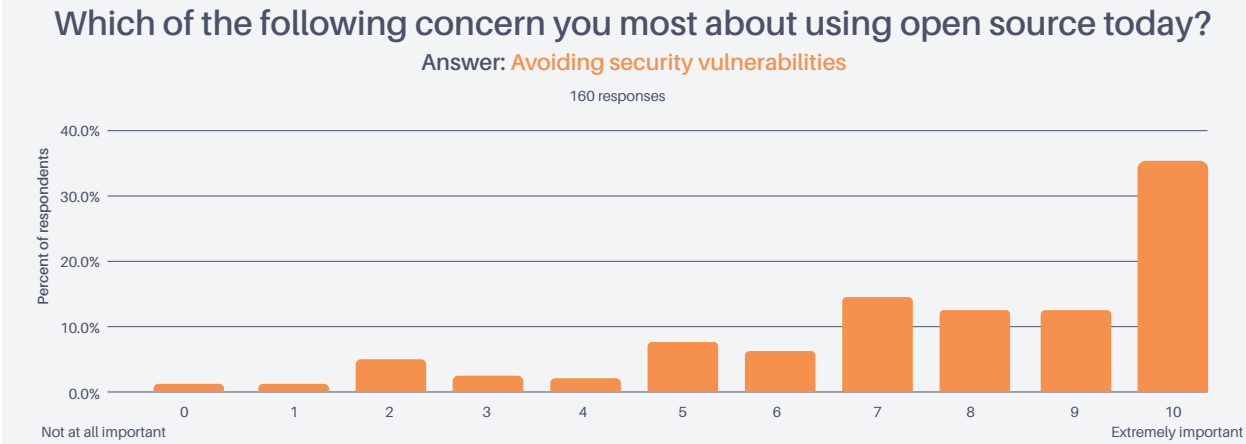
As with the reasons why developers use open source, we examined concerns by respondent geography, development team size, and role/title. The results are mostly consistent across these parameters, with a few interesting exceptions.

- Software architects, engineering managers, and product managers tend to be more worried about making safe bets on packages being maintained in the future than are respondents with software developer titles.
- Similarly, software architects worry more about avoiding license compliance problems than do others we surveyed.
- Development team size only impacted one concern—keeping track of the open source code in use. Respondents in organizations with development teams of 1,000 or more people are significantly more concerned with this than are respondents from smaller dev shops.

We'll take a closer look at the top three open source concerns below.

CONCERN 1: AVOIDING SECURITY VULNERABILITIES

As the figure below shows, three quarters of professional developers rate avoiding security vulnerabilities a "7" or higher on a scale of 0-10, and 35% rated their degree of concern at a "10." It's no wonder considering the series of newsmaking exploits like EventStream, Leftpad, and floatdrop in JavaScript and colorrama in Python.



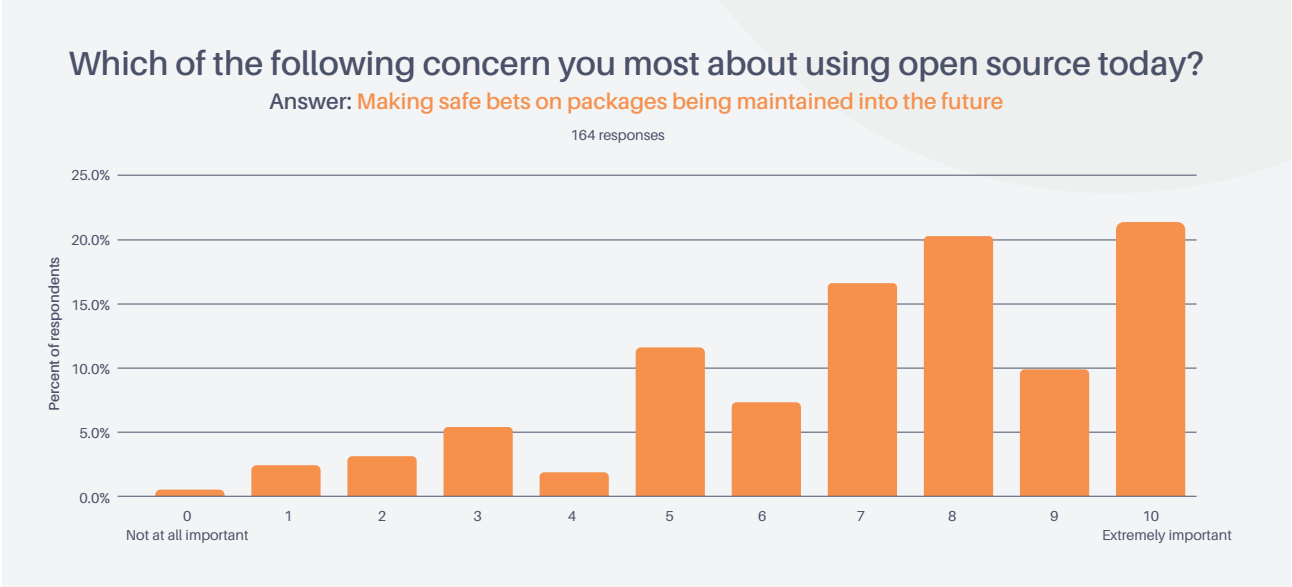
This concern is consistent across geography, title, and company size.

CONCERN 2: MAKING SAFE BETS ON PACKAGES BEING MAINTAINED INTO THE FUTURE

Survey respondent:
"It's sometimes hard to understand if a package is being maintained or not. Lots of issues and no updates for two years = not maintained, but it's not always that clear cut."

Over two thirds of professional developers rate their level of concern with making safe bets on packages being maintained in the future a "7" or greater. This is no surprise, as we hear this topic come up time and again when speaking with professional development teams.

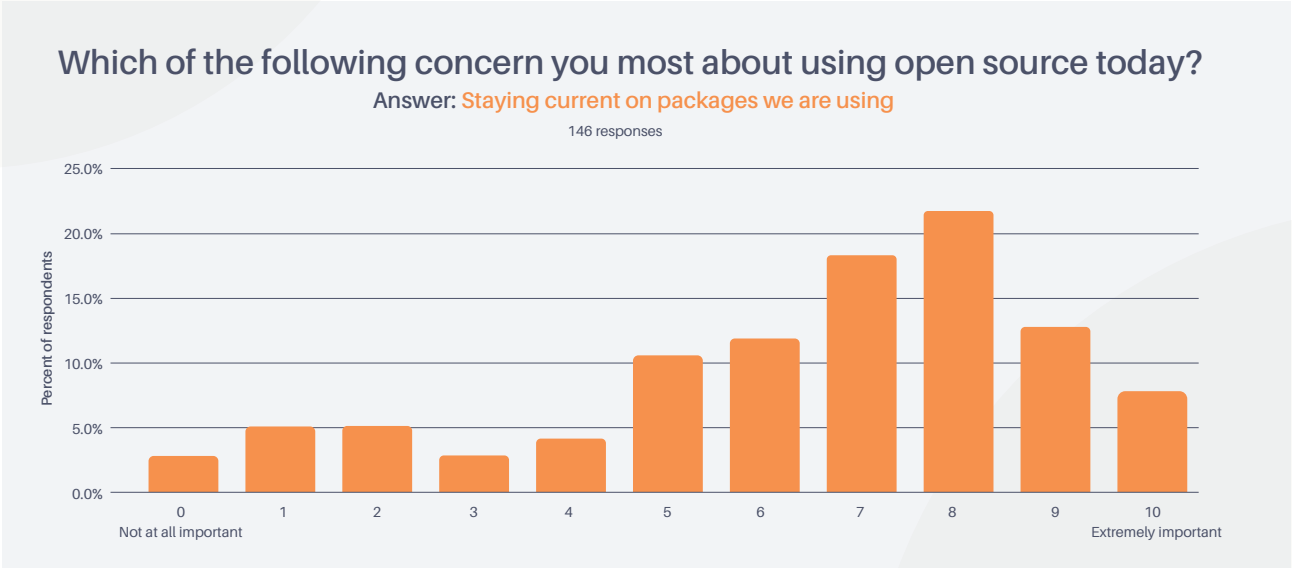
Survey respondents with titles software architect, engineering manager, and product manager worry more about this area than respondents with software developer titles. This may be a reflection of more senior developers having experienced more of the pain associated with having to replace a deprecated package. It also may reflect these respondents' broader perspective on the architecture of a codebase and the long view that comes with this.



CONCERN 3: STAYING CURRENT ON PACKAGES THEY'RE USING

Nearly 60% of respondents rate this concern with using open source a "7" or higher. Uncoordinated, and in many cases unplanned, release schedules complicate staying current for professional developers.

Survey respondent:
"We use React Native and there's a lot of effort to update to a new version after a long time without [an] update."



OTHER CONCERNS DEVELOPERS HAVE WITH OPEN SOURCE

Only one of the options we provided scored less than "5"—showing the security or legal team that developers are addressing concerns.

And while responses were largely consistent, development team size did impact responses to 'keeping track of which open source code we are using.' Respondents in organizations with development teams of 1,000 or more people are significantly more concerned with this than are respondents from smaller dev shops.

Survey respondent:
"There are times where an open source platform or dependency's developers no longer have the time to contribute to the projects and the teams can dwindle down until there's no longer support from the leaders in that community."

Respondents also express concern over the future maintenance of open source. In the words of one developer "most useful open source libraries and tools are dramatically under-funded. Only a small minority of these projects have enough active maintenance for it to be worthwhile for me to depend on them."

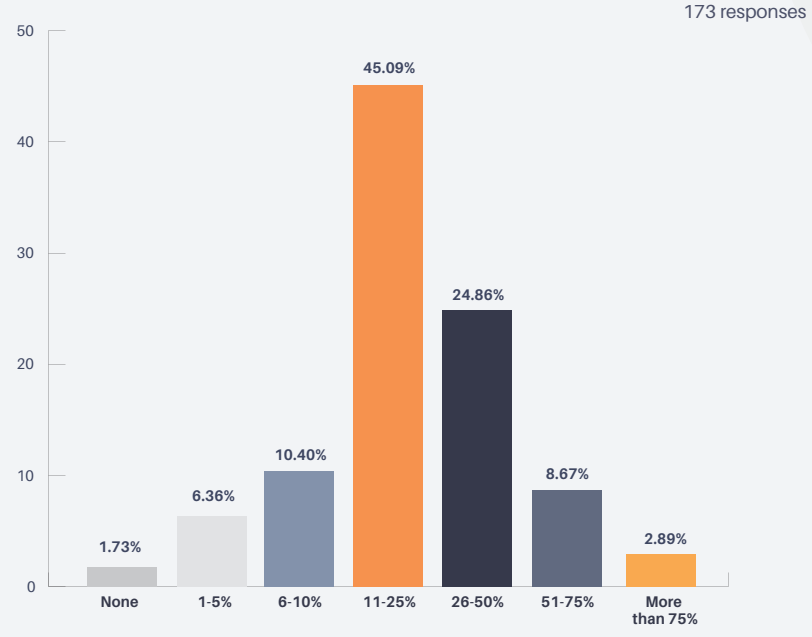
Finding #3: code maintenance takes 30% of the average developer's week, and a quarter of that time is related to the open source packages they use

Our survey found that most respondents (70%) spend between 11 percent and 50 percent of their time on code maintenance. For a 40-hour work week, this equates to between 4.4 and 20 hours per week.

When we translate the percentages into weekly hours for all respondents, the average time developers in our survey spend on maintenance is 12 hours per week. This turns out to be slightly lower than the findings in this report from Stripe, which estimated 17.3 hours spent on code maintenance (13.5 hours addressing technical debt and 3.8 hours on bad code).

But it does beg the question, if developers could get some of this code maintenance time back, what else could they do with it?

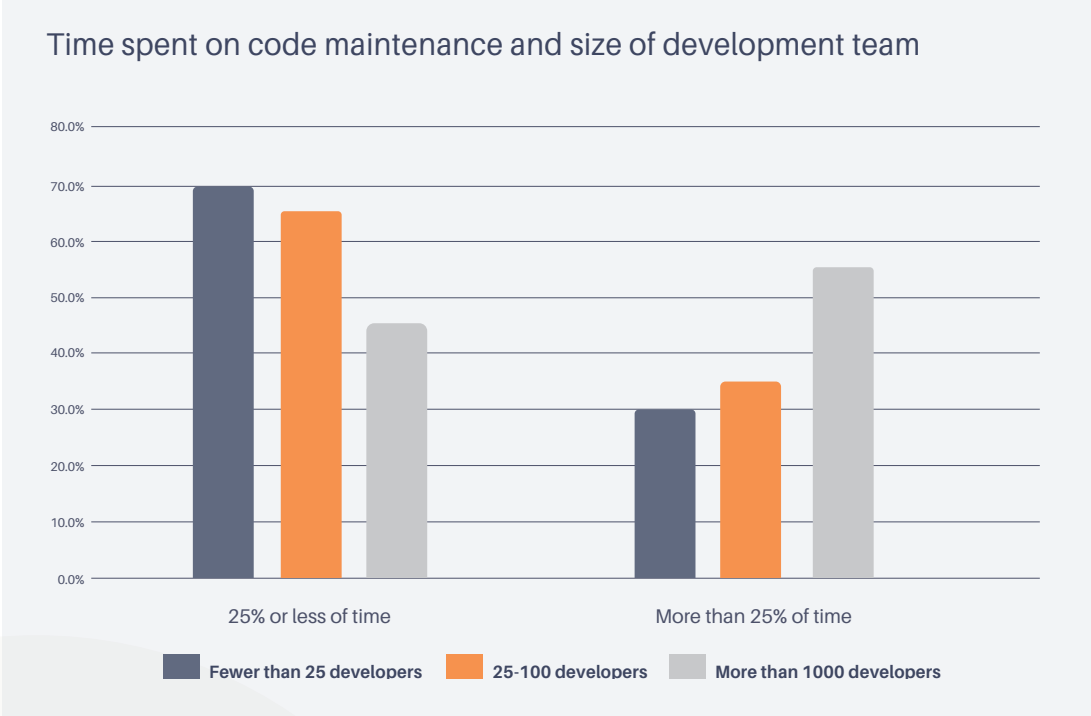
How much of your time do you spend on code maintenance (refactoring, debugging, rewriting functionality, fixing technical debt, etc.)?



BIGGER TEAMS SPEND EVEN MORE TIME ON CODE MAINTENANCE

We examined the data by respondent geography, development team size, and role/title. The results are mostly consistent across these parameters, with one important exception.

The larger the development team, the larger percentage of time respondents dedicate to code maintenance.

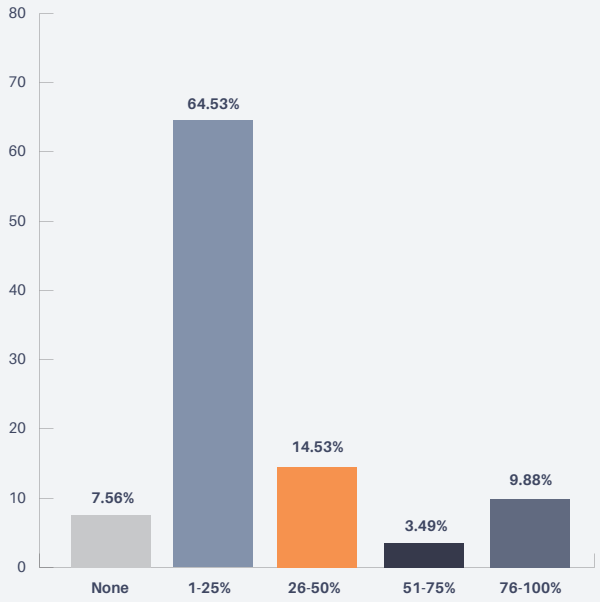


We suspect this could be a factor of the age and size of the codebase, with bigger teams typically working with larger and older codebases, which require more maintenance to keep up to date.

OPEN SOURCE CONTRIBUTION TO CODE MAINTENANCE

On average, the open source packages professional developers consume contribute 25% of the total code maintenance workload. But this average masks some interesting variance.

What percentage of the time you spend on code maintenance is related to the open source packages you use?

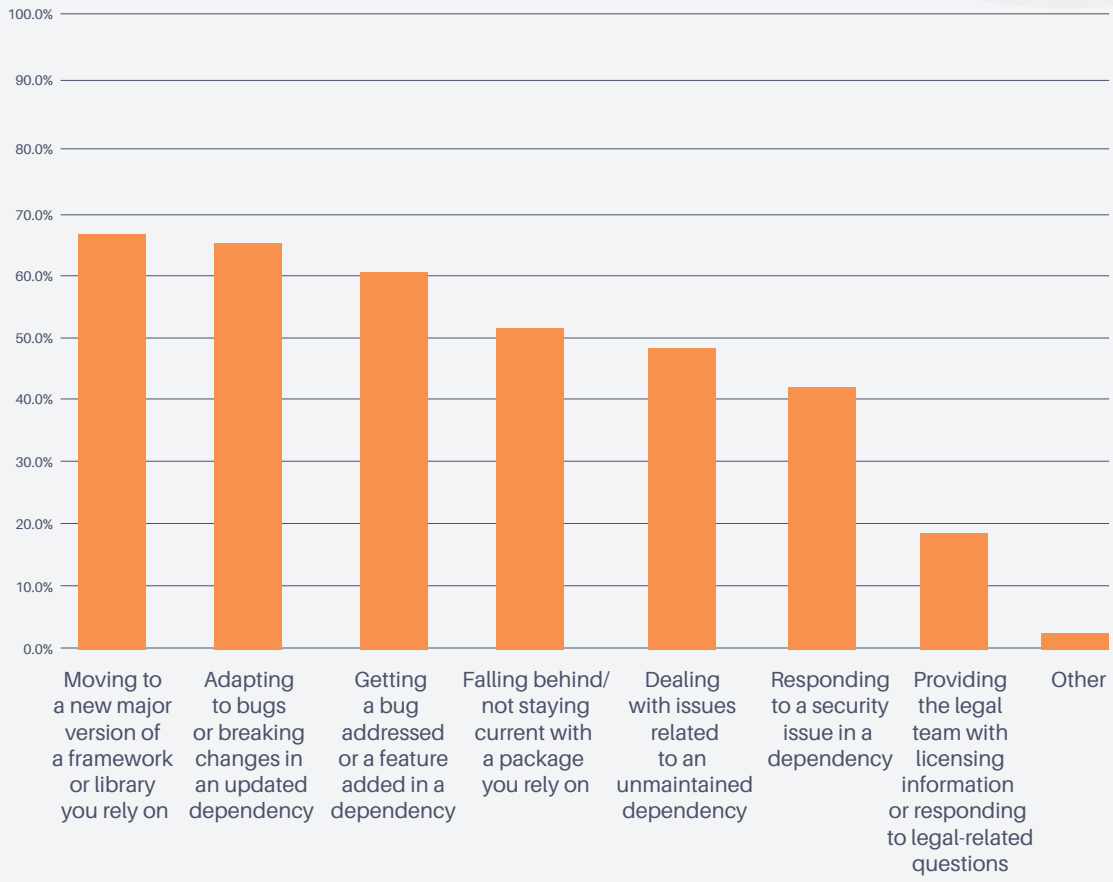


A third of developers in our survey spend more than 25% of their code maintenance time on open source components. And for nearly 10% of respondents, the vast majority of their code maintenance work (between 76 and 100%) relates to open source packages.

These percentage splits are consistent across geography, title, and size of development team, which means some other factor may be at work. One possible explanation might be that developers reporting the majority of their code maintenance work is on open source software may work on applications that only use open source components. Or it could be that the task of open source package maintenance where they work is assigned to specific developers, rather than be distributed evenly across the development team.

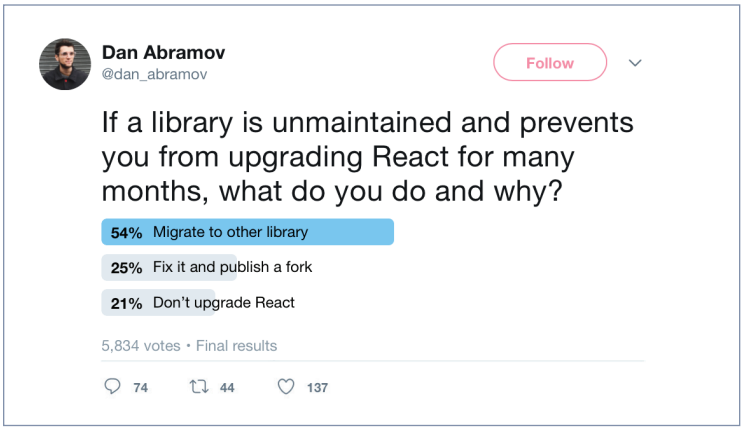
THE MOST COMMON OPEN SOURCE MAINTENANCE WORK IS IMPROVING, REPLACING, AND KEEPING UP WITH PACKAGES

Which of the following have taken up your or your team's time in the past 6 months?



The most time-consuming and common open source maintenance task is moving to a new major version of a framework or library. This is closely followed by adapting to bugs or breaking changes in an updated dependency. Other time consuming maintenance activities include getting a bug addressed or a feature added to a dependency, falling behind / not staying current with a package, and dealing with issues related to an unmaintained dependency. Somewhat surprisingly, responding to security issues in dependencies consumes less time according to our respondents (although the ramifications of security issues when they do happen can be quite severe).

Frequently, these maintenance activities occur together. For instance, a maintenance lapse in one of the hundreds of React dependencies can trigger a number of activities, as the following tweet illustrates.

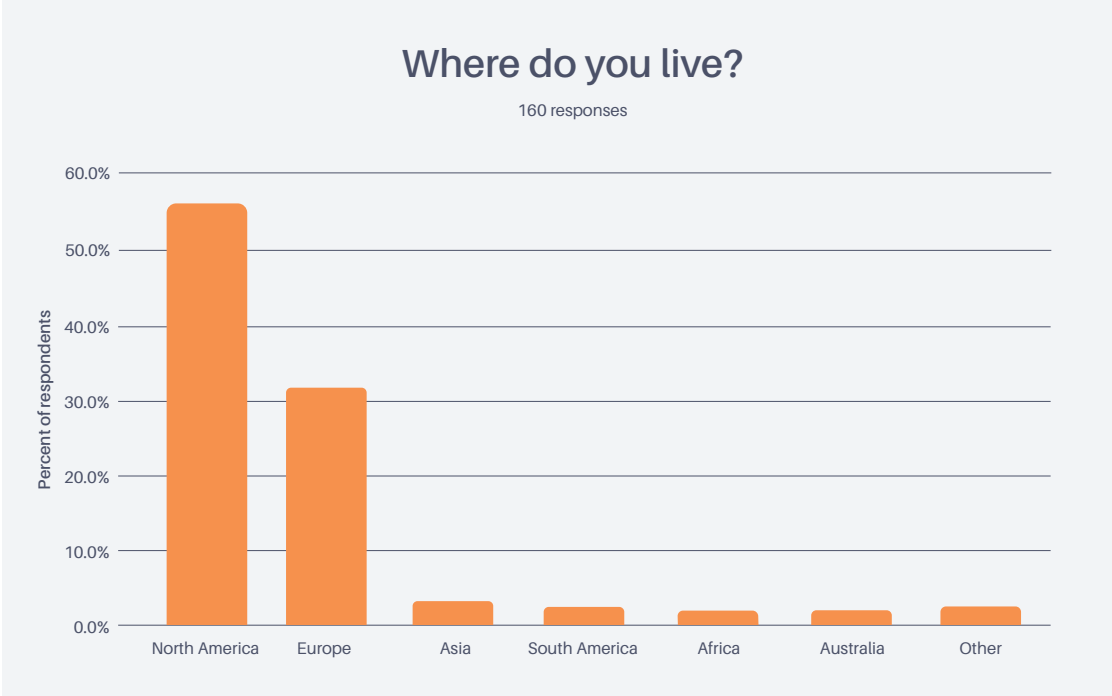


CONCLUSION AND METHODOLOGY

We are very grateful to all of the developers who took the time to complete our survey. We knew going into this project that professional developers of every stripe use open source every day, and now we know why. Our data clearly shows that open source helps you get more done, faster, and helps reduce costs in a variety of ways. We also learned more about the prevalence of challenges open source introduces.

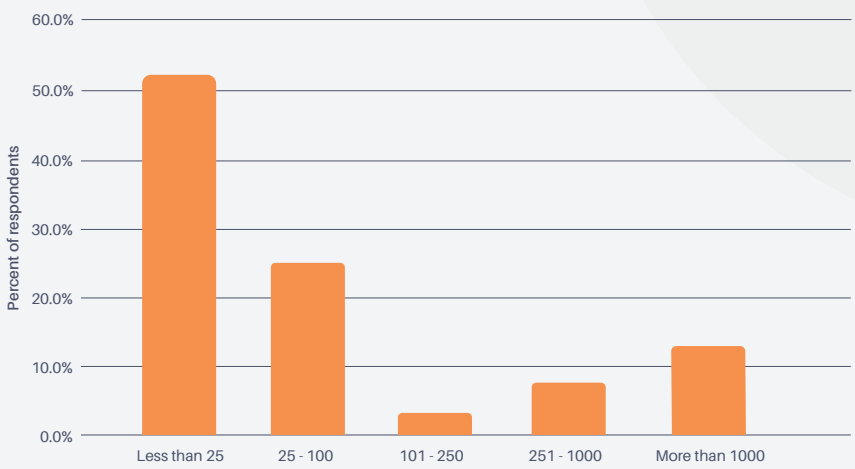
ABOUT THE SURVEY

Tidelift fielded the survey online from November 14 to December 10, 2018, sharing the survey primarily through email and Twitter outreach. These efforts yielded a total of 308 responses.



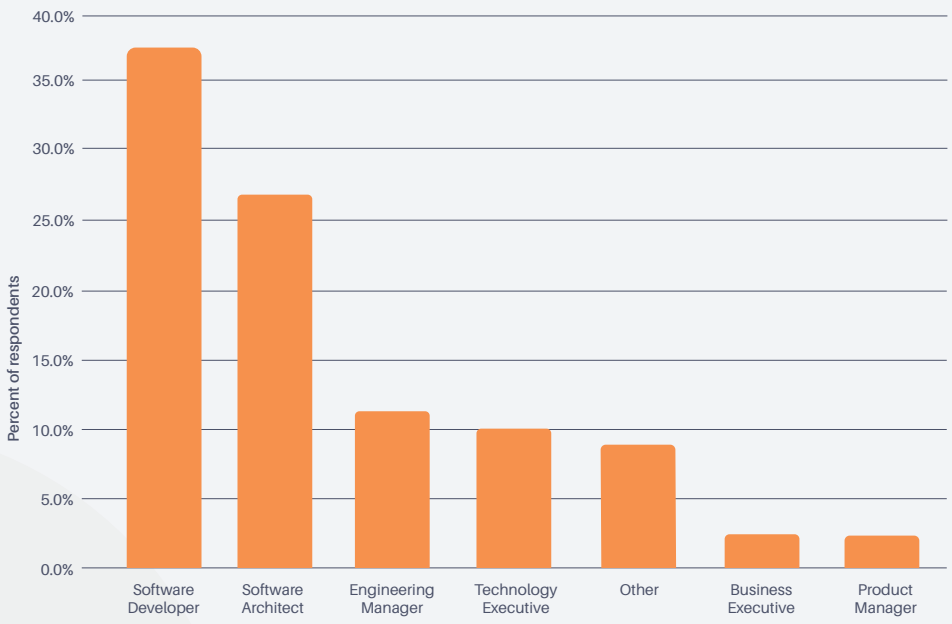
How many people work in software development at your company?

158 responses



Which of the following job categories most closely matches your role?

159 responses



ABOUT TIDELIFT

Tidelift makes open source work better—for everyone

- For software development teams, we provide a managed open source subscription backed by the creators and maintainers who built the packages they depend on.
- For open source maintainers, we provide a way to get paid for the value they create, simply by keeping their projects well maintained.
- Tidelift is making the space for open source to thrive, so we can create even more incredible software, even faster.

ABOUT THE TIDELIFT SUBSCRIPTION

The Tidelift Subscription manages your dependencies for you.

We provide the tools you need to continuously catalog and understand the open source software that your application depends on.

We partner with and pay the open source community maintainers of the exact packages you use, to ensure they meet the standards you require.

We address issues proactively, not only scanning for new security, licensing, and maintenance issues, but also working with our participating open source maintainers to resolve them on your behalf.

We help you measure and improve your open source dependencies' health—which improves your app's health—and give you a short list of high-impact steps your team can take to improve them even more.

We **add commercial assurances that don't come for free** with open source packages, like intellectual property indemnification and support under a service level agreement. You expect these guarantees from proprietary software, and you should get them when using open source as well.

The end result? All of the capabilities you expect from commercial-grade software, for the full breadth of open source you use. That means less time grappling with esoteric open source trivia, and more time building your own applications—and your business.

Ready to see how the Tidelift Subscription would look with your codebase?

Request a demo and learn more: tidelift.com/subscription