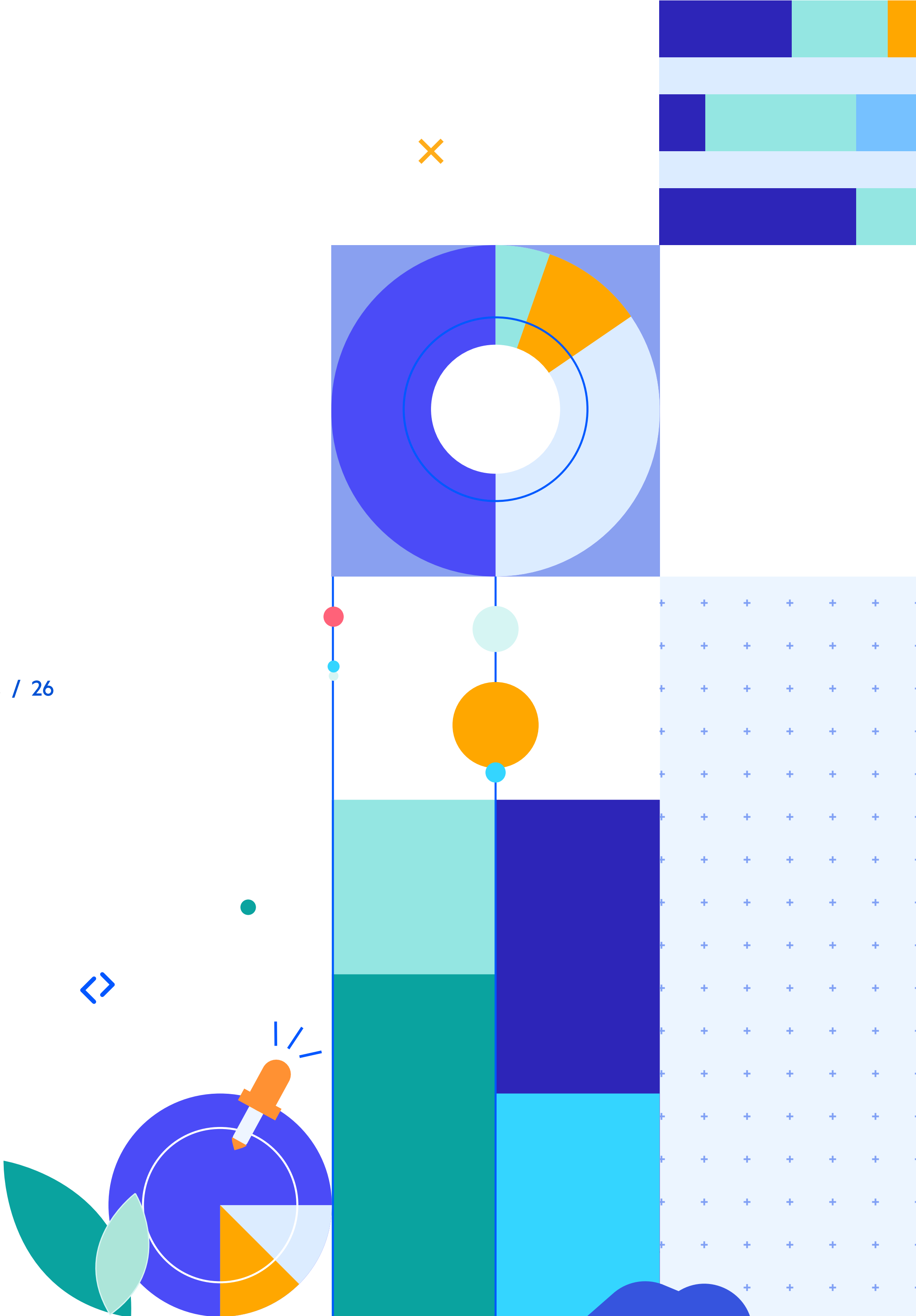
Progress® Telerik®

State of Designer-Developer Collaboration 2024

SURVEY REPORT

Table of Contents



Introduction

Dear designers and developers, business owners and executives, and folks wearing many hats,

Whether you're on the development or design end of creating web applications, it's equally easy to lose track of the world beyond your daily work. Sometimes, it can be hard to break out of the silos our companies create for us, inadvertently or on purpose. However, the crossroads of design and development is a bustling, beautiful, and, yes, sometimes a bit of a bumpy place, and it is one worth visiting often for anyone looking to be a better and happier professional.

In this survey report, we'll go deep into the findings of the State of Designer-Developer Collaboration 2024 survey and surface everything there is to learn about the relationship between designers and developers, including ideas for making it more efficient and satisfying. To quote Figma's Front End Developer Research Report (Oct 2023), "The designer/developer relationship just might be the most important partnership in business today."

Your authors,

Kathryn Grayson Nanz

Nora Petrova

Manuela Gogova

Designed by **Deepika Prabhakaran**

Highlights

While 36% of engineers say designer-developer collaboration goes smoothly and without challenges,

only 10% of designers would agree.

The top three interventions your teams yearn for are:

- For designers to be trained on the technical constraints developers work with
- For developers to be brought into the design process earlier
- For developers to learn more about the principles of design

On design systems:

For the folks who don't yet have a finished design system – better user experience, improved consistency and faster design to dev time are the top three reasons why they want one. And great news: those are also the top three benefits listed by folks who do have them!

Methodology

The State of Designer-Developer Collaboration 2024 survey was conducted through an online self-completion questionnaire. The questionnaire consisted of pre-defined questions exploring the designer-developer collaboration process and design system creation and adoption with their related challenges and benefits.

The target audience for the survey was everyone involved in the design and development of web applications.

Respondents were invited to take the survey through various channels—being prompted on the telerik.com website, through social media and blog post promotion, and through paid promotion in newsletters and on social media.

The survey was open for submissions for the period July 18 – September 08, 2024.

We received a total of 467 responses from 51 countries. Before proceeding with the analysis, we cleaned and validated data for accuracy and relevance using both targeted survey distribution and screening questions to filter out respondents outside our target group and bots. We also learned that bots taking surveys are getting more and more advanced, leaving AI-generated feedback on open-ended questions. After the clean-up, we ended up with 222 respondents.

Note on the Data

All results published here have been tested for statistical significance —and passed.

If the sum of all responses is more than 100%, the question was multiple-choice.

Note on the Wording

Whenever we look at the differences between designer and development roles, we'll refer to them as "design/designers" and "development/developers/engineers/dev" for short.

The job roles are grouped in these two categories in the following way:

Designer Roles	Developer Roles
Design (UI/UX) manager	CEO/CTO/VP of engineering
Design system lead	Engineering manager
Graphic designer	Front-end developer
Interaction designer	Full-stack developer
Product designer	Product manager
UI/UX designer	Web developer

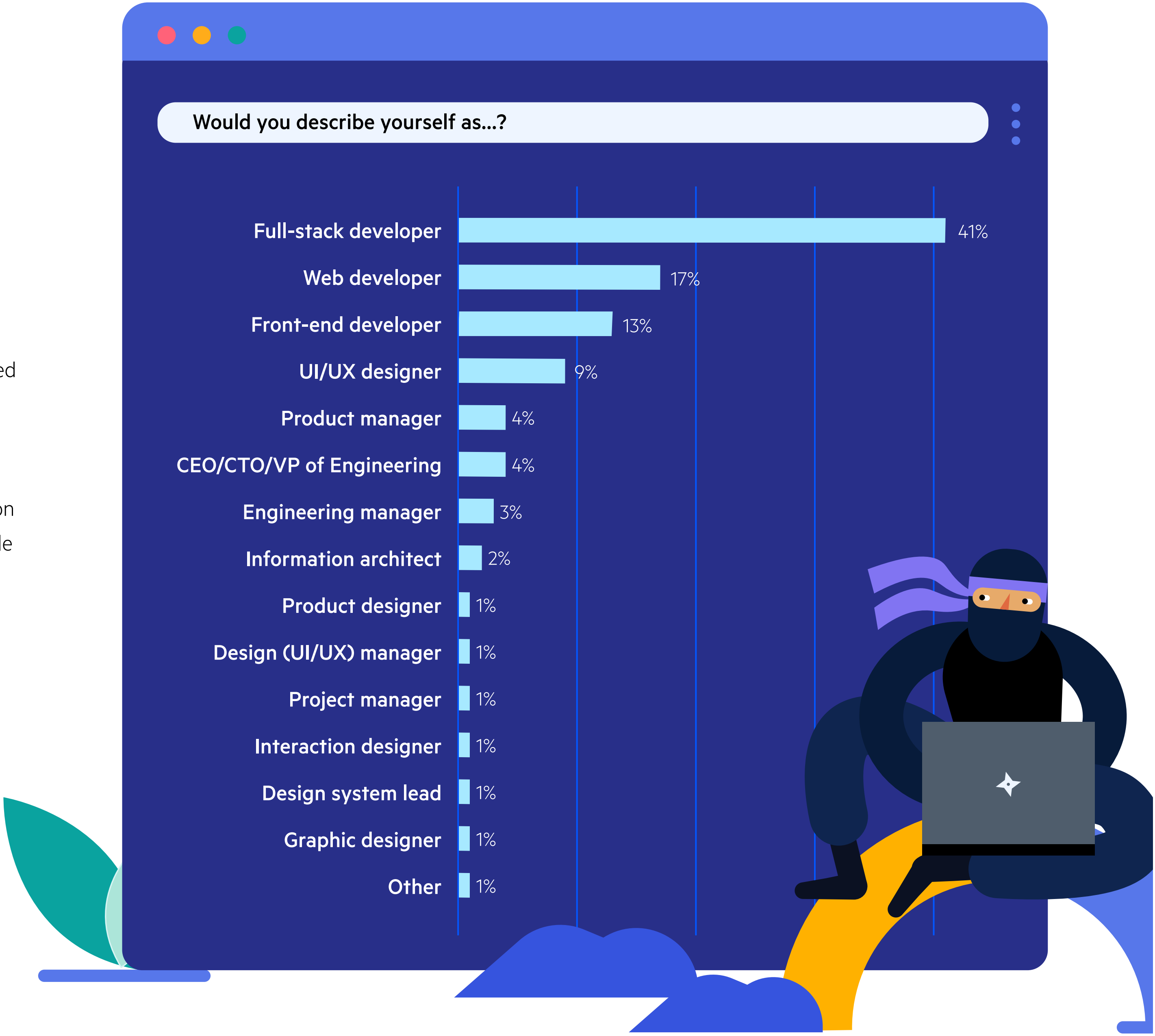
Participants

The survey invitation specifically recruited people who were:

- Web designers or developers
- Stakeholders in the designer-developer handoff (e.g. managers, product owners, senior leadership)

Anyone who selected a role outside this scope exited the survey.

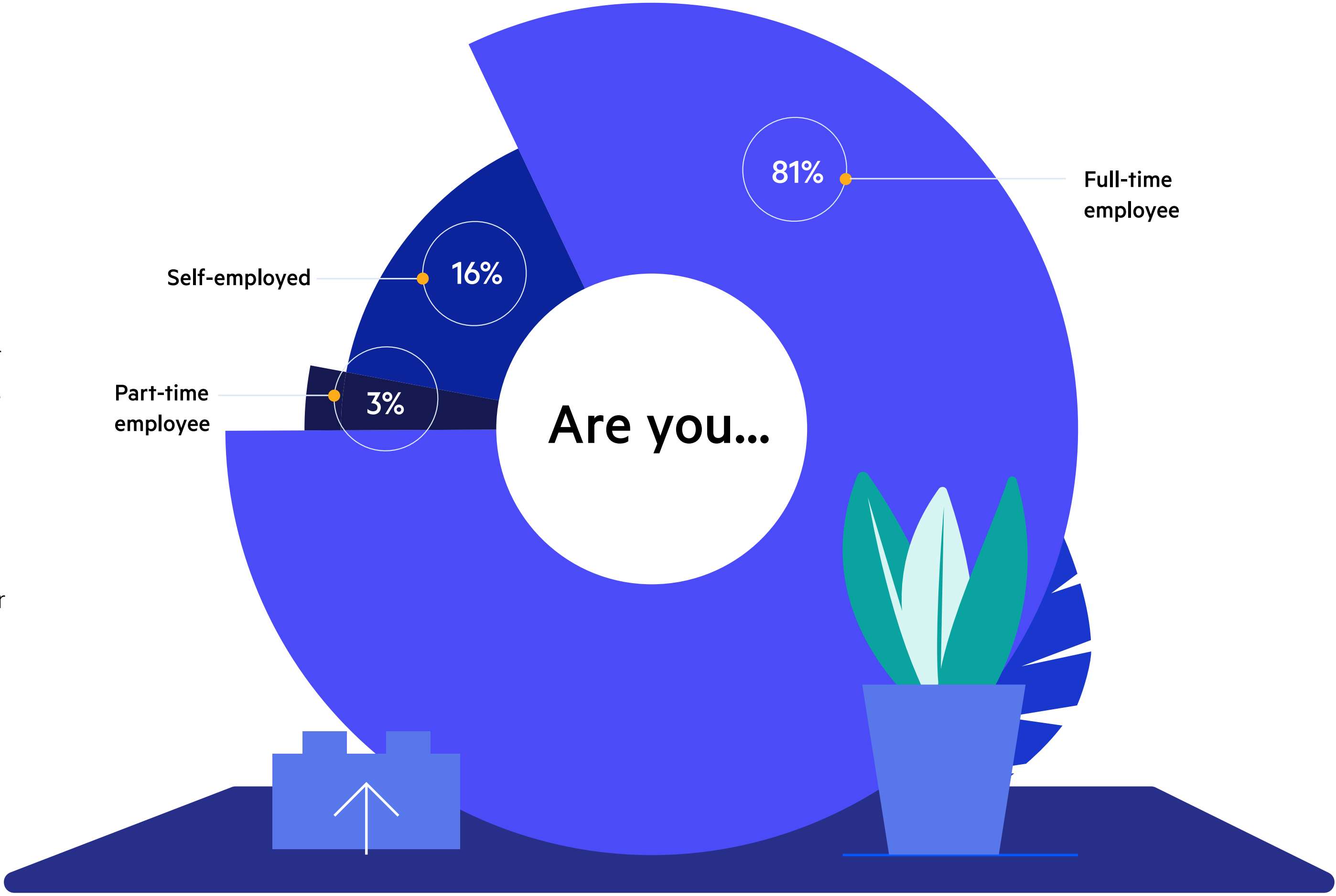
While the responses we got are skewed (82% of respondents were on the engineering side vs. 14% on the design side and 4% in adjacent roles), the sample is large enough to give us valuable insights.



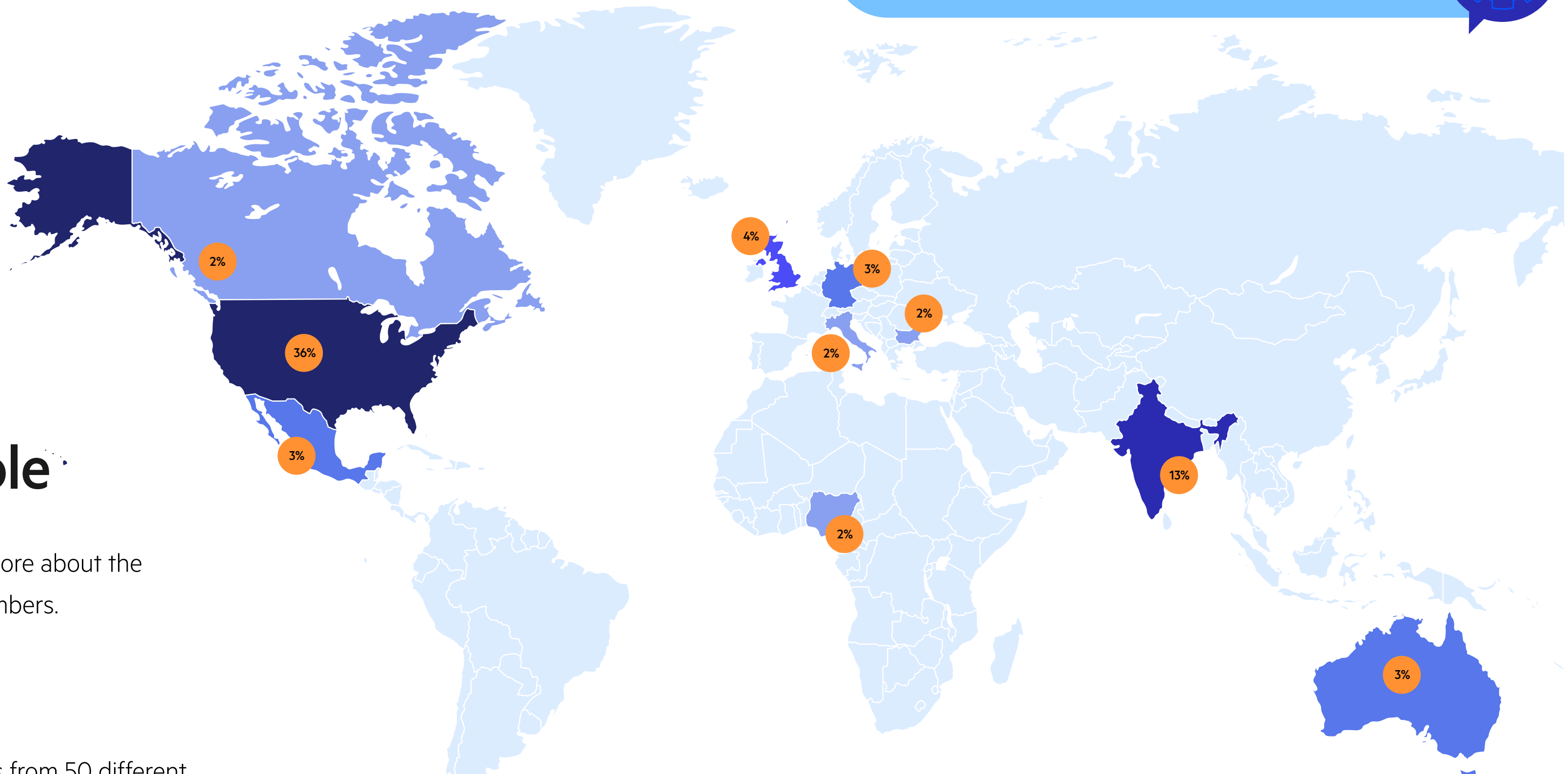
The majority of respondents were full-time employees.

One interesting difference between developer and designer roles is that while 18% of all engineering roles reported being self-employed, only 7% of designers did. What's more, while 20% of developers chose the "solo entrepreneur" label, none of the designers did.

Without having asked follow-up questions about this section, we can't say for sure where this discrepancy comes from. However, we don't think it's a reach to say that – on the balance – the need for a developer is more straightforward than that for a designer, and designers have historically been undervalued in the IT world. This makes it harder for designers to be self-employed. Often, designers are faced with the challenge of selling themselves and making a case for why their skills are necessary, whereas developers are not. This may be especially true in our current tech landscape; AI and prefabricated design templates can get a design halfway there. For many small businesses and startups, that's good enough (for now). Perhaps we'll see this percentage shift next year as AI-powered coding becomes just as accessible as AI-powered design.



In which country do you currently reside?

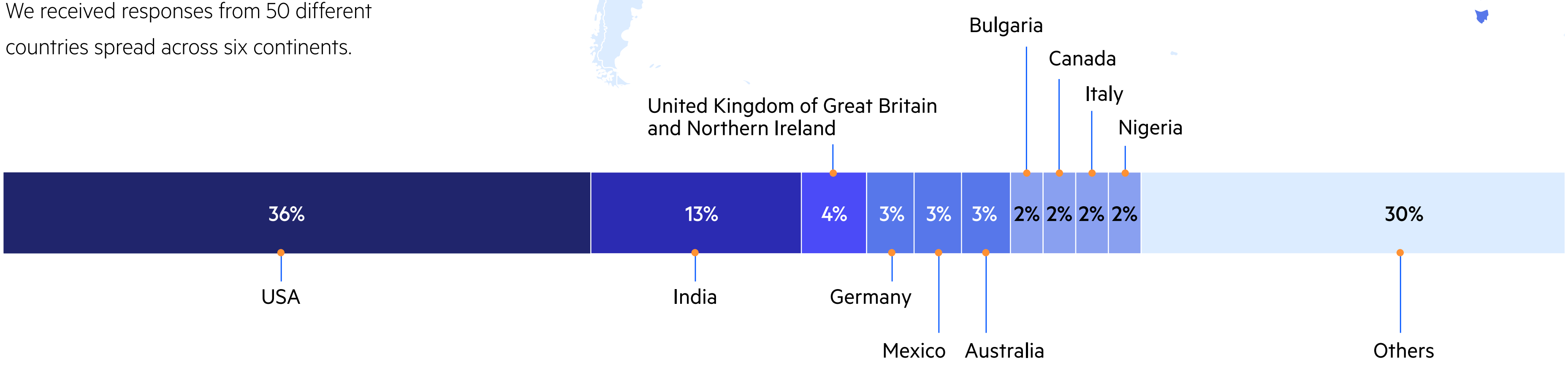


The People

Let's learn a little bit more about the people behind the numbers.

Geography

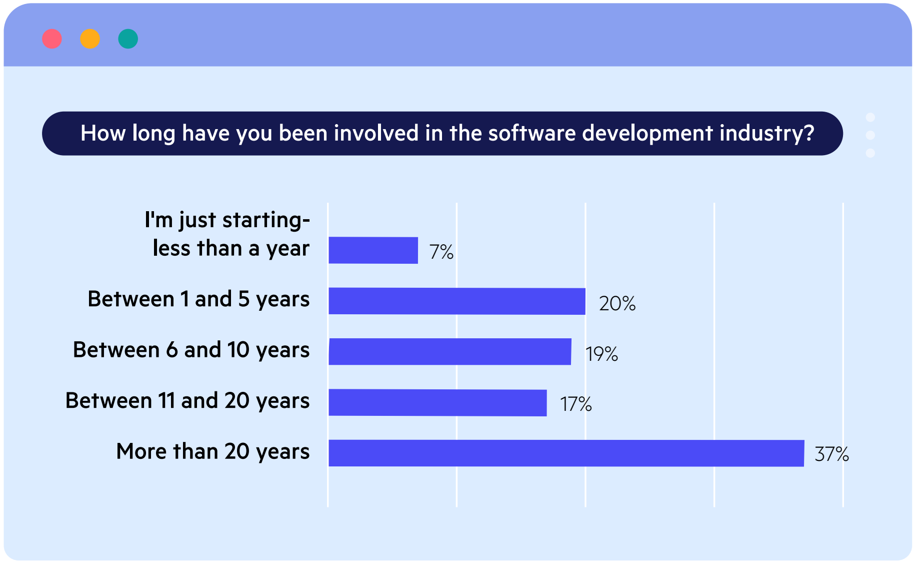
We received responses from 50 different countries spread across six continents.



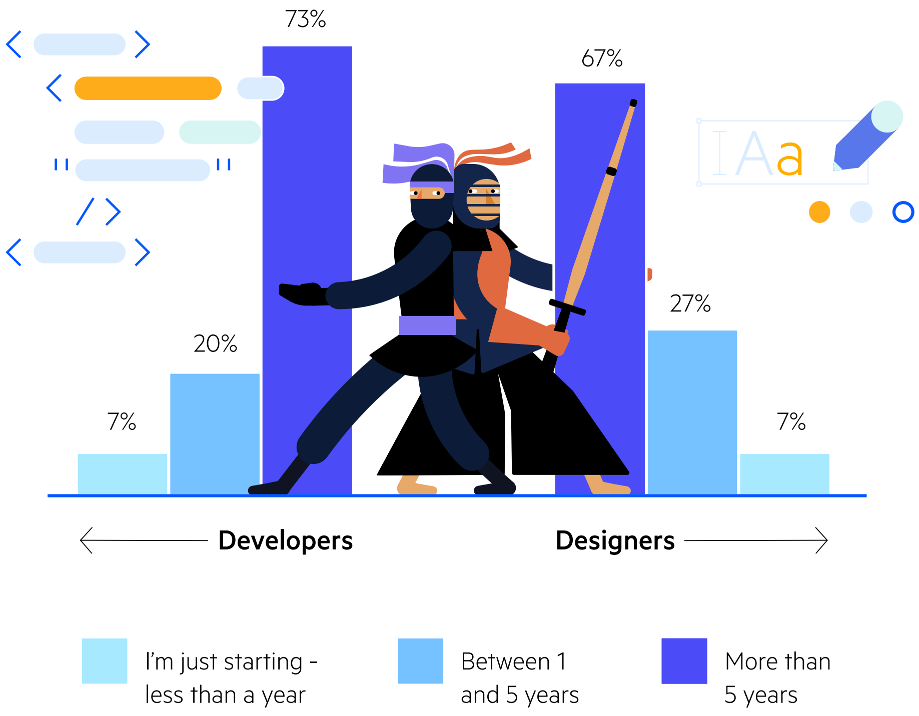
Experience

Age is but a number; you could be 60 and just starting on a new career path. That's why, instead of age, we asked for the respondents' years of experience. A majority (54%) of respondents have 11+ years in the software development industry, and 27% have been at it for 5 years or less. When you split the data by design and development roles, the distribution is similar in both camps. The only notable difference is that 30% of the designers have between 6-10 years of experience versus 18% of developers. On the other hand, 38% of developers have more than 20 years, as compared to 20% of designers.

Similarly, 58% of Stack Overflow Developer Survey 2024 respondents have coded for 10+ years. Perhaps more experienced professionals see more value in the information they get from these kinds of surveys, so they're more willing to take the time to respond.



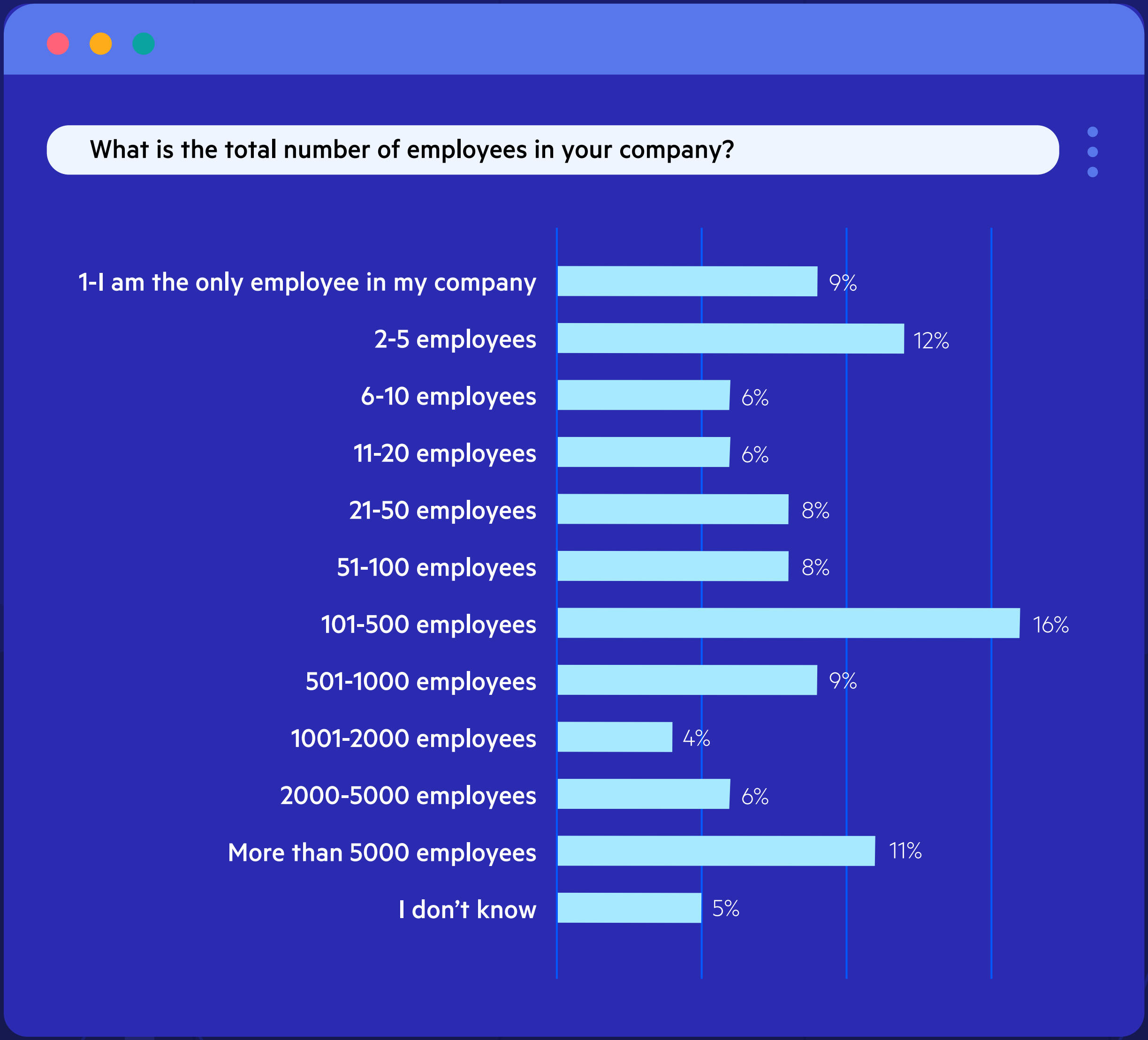
Experience by role



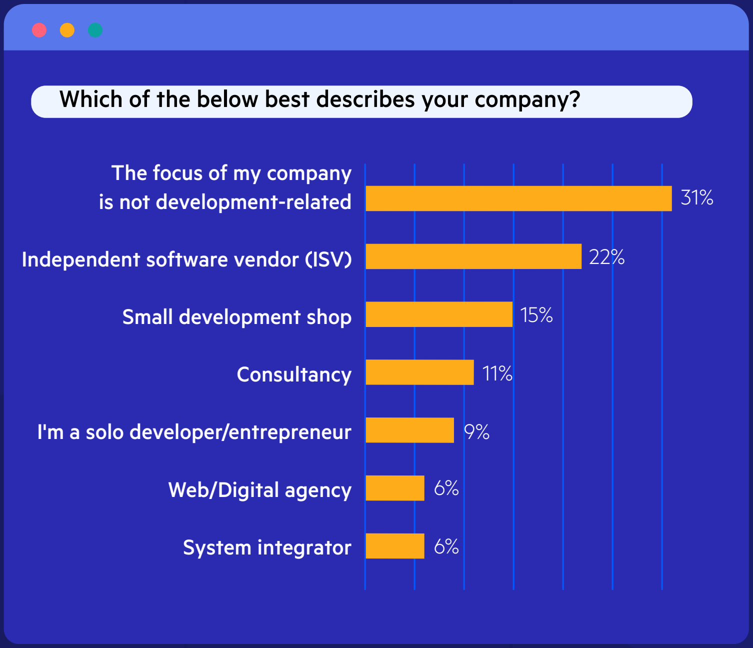
The Businesses

The survey represents the full spectrum, from one-person businesses to small and medium-sized companies and enterprises with 5,000+ employees.

30% of all respondents work in large enterprises of 500+ employees, while 41% work in small businesses of 2-100 employees. The distribution is very similar when we look at the design/dev split with one curious difference – 17% of designers have answered with, "I don't know," versus only 3% of developers. Could this mean that designers are more siloed and have less of a chance to know the business outside their immediate scope, as opposed to developers?



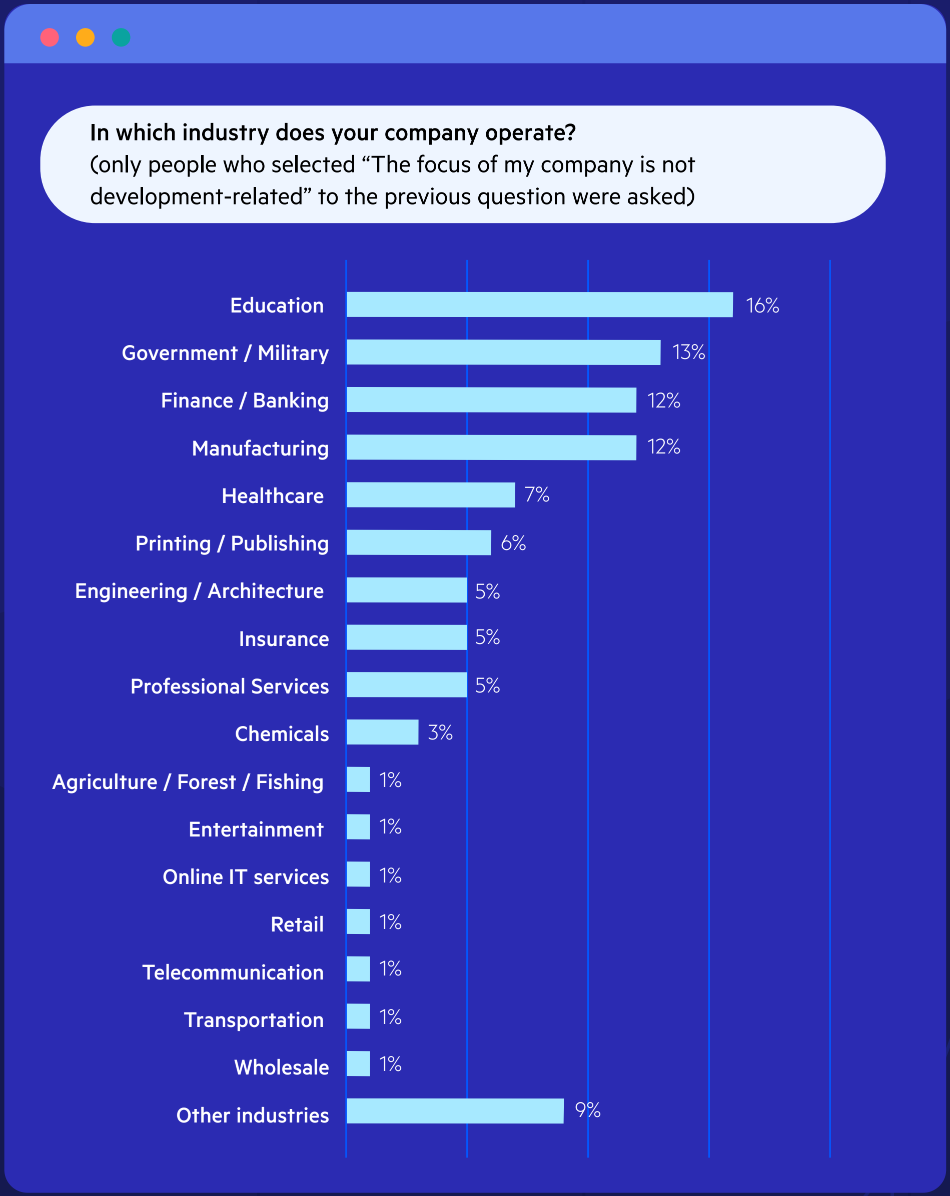
What is the total number of employees in your company?



Which of the below best describes your company?

The largest share of respondents (31%) work at companies whose focus is not on software development (e.g. a manufacturer or an educational institution), and 22% work for independent software vendors. These are two very different situations – won't it be interesting to see how design & development work together in each type of company? Keep reading and you'll get to this cross-section!

People working in over 20 different industries took the survey. Education, government, finance, manufacturing and healthcare were the most represented ones.



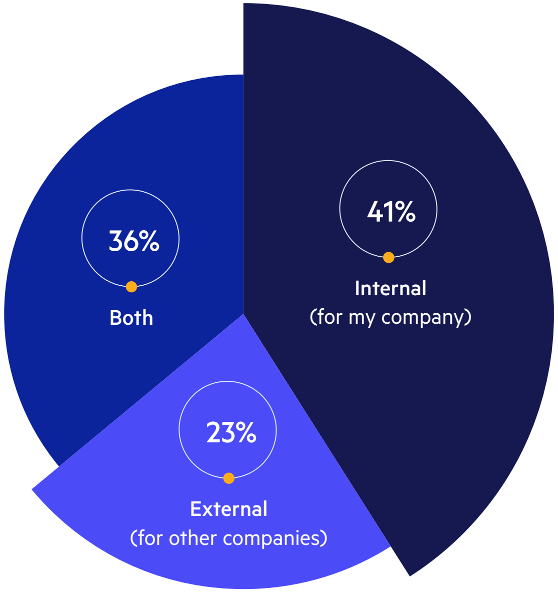
In which industry does your company operate?
(only people who selected "The focus of my company is not development-related" to the previous question were asked)

The Context

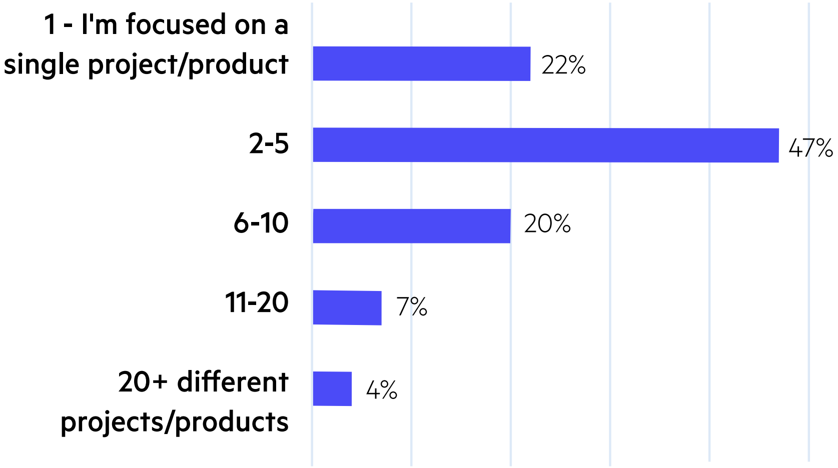
The work environment can significantly affect the relationship between partnering teams. While working for internal vs external stakeholders did not correlate with the relationship quality between design and development (a data point you'll learn more about soon), the number of projects per year did. You'll find out how in the fragment dedicated to cross-splits.

Work Type

What kind of web apps are you currently working on?



How many different projects/products do you usually work on in one full year?



Aren't we all feeling sympathetic to the 4% of folks working on 20+ different projects a year—this sounds challenging, if only because of all the context switching!

It's easy to imagine how teams delivering 10+ web apps a year could have more friction points than teams having the luxury of focusing on a single project (unless it's that legacy app with no comments in the code and barebones documentation that you need to bring into the 20th century). However, the data shows something harder to imagine: teams working on 10+ projects are mostly fine, teams working on a single project are also fine, and those working on 2-10 projects are least likely to have a positive relationship.

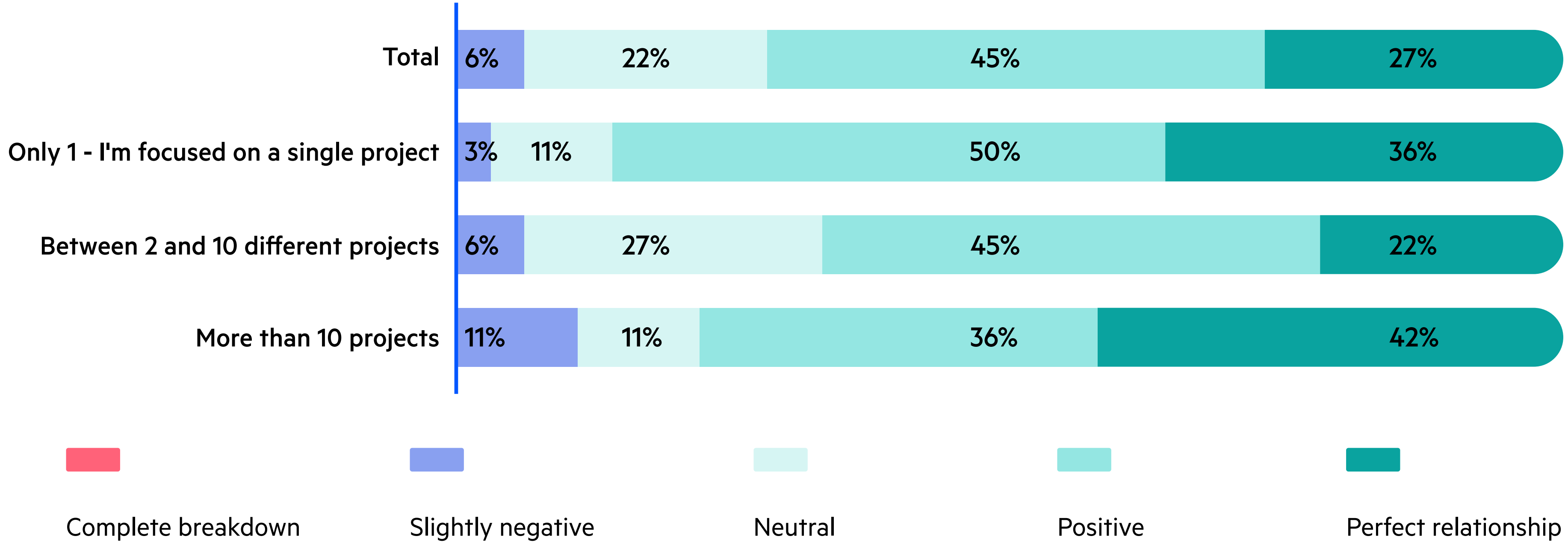
Perhaps people working on 10+ projects are forced to establish a structure and processes that enable them to be at their most efficient so as not to drown in deadlines. That's another question we'll leave for 2025.

Cross-section of:

"If 10 is a perfectly functioning relationship with nothing to add and 1 is a complete breakdown where no one talks to the other, how would you rate the relationship between designers and developers working on the same project?"

X

"How many different projects/products do you usually work on in one full year?"



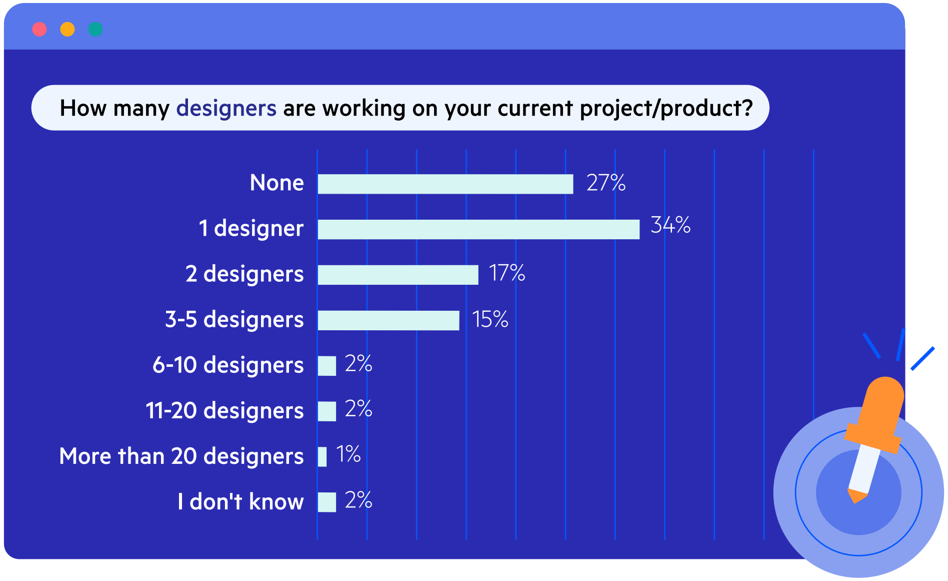
Team Composition

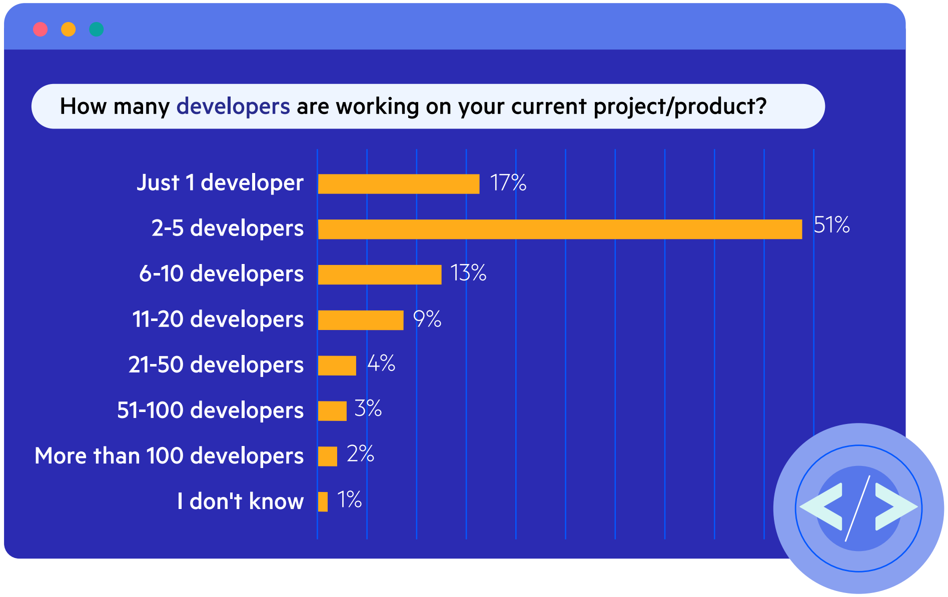
The more, the merrier, the saying goes—but is that so when it comes to team composition? **The most prevalent team model involves 2-5 developers and 1 designer.** This aligns with the ratio that zeroheight saw in their Design Systems Report 2023, where they saw ranges between 1:6 and 1:15 depending on company size.

One in four developers we surveyed doesn't have a designer assigned to their current project. Often, this means that developers are leveraging resources with design "built-in" and making minimal UI changes. This could include UI component libraries or pre-existing design systems such as Material or Fluent. For specific design needs, these groups may hire a freelancer or contract worker to help with a one-off design project (read on for more insight on this). Generally, we see this kind of setup in startups, small businesses or other similarly under-resourced ventures. As businesses (and userbases) grow, so does the need for a dedicated designer.

On the design side, 17% have just one developer they partner with on their current project and 51% work with 2-5 developers. The 1:1 setup has its advantages, as it's simpler to navigate than a larger team; however, it has its downsides as well—if these two people don't get along, they will have a hard time (and so will their project).

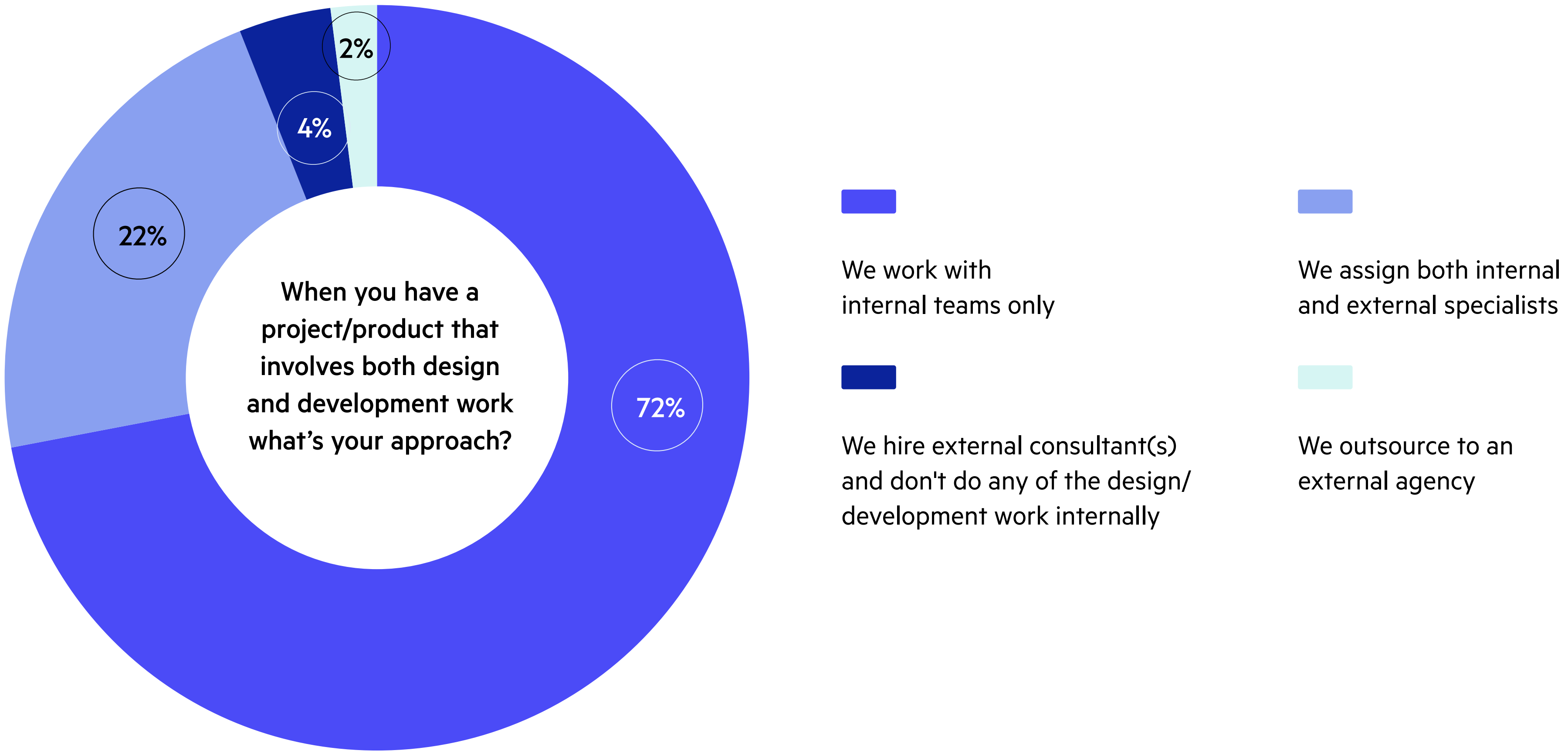
Are you as curious as we are about whether there are differences in how people get along based on the composition of their team? We'll find out when we get to the cross-sections!


How many designers are working on your current project/product?


How many developers are working on your current project/product?

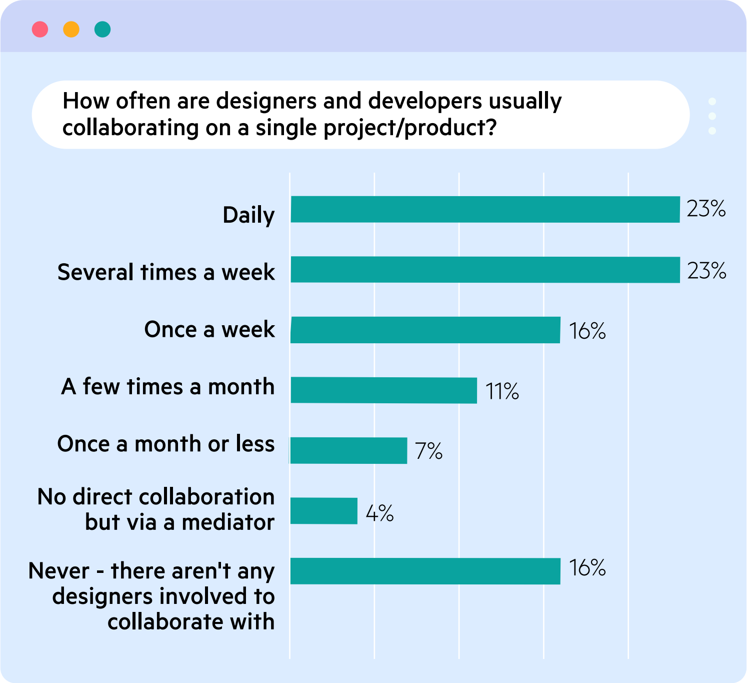
When it comes to hiring outside help, most companies seem reluctant – 72% count exclusively on their internal specialists to get the job done.

A small share of developers (5%), versus none of the designers, report that their companies don’t do any design/development work internally. Having designers on the team seems to be one of the indicators that a company is more committed to creating at least part of its web products/projects internally.



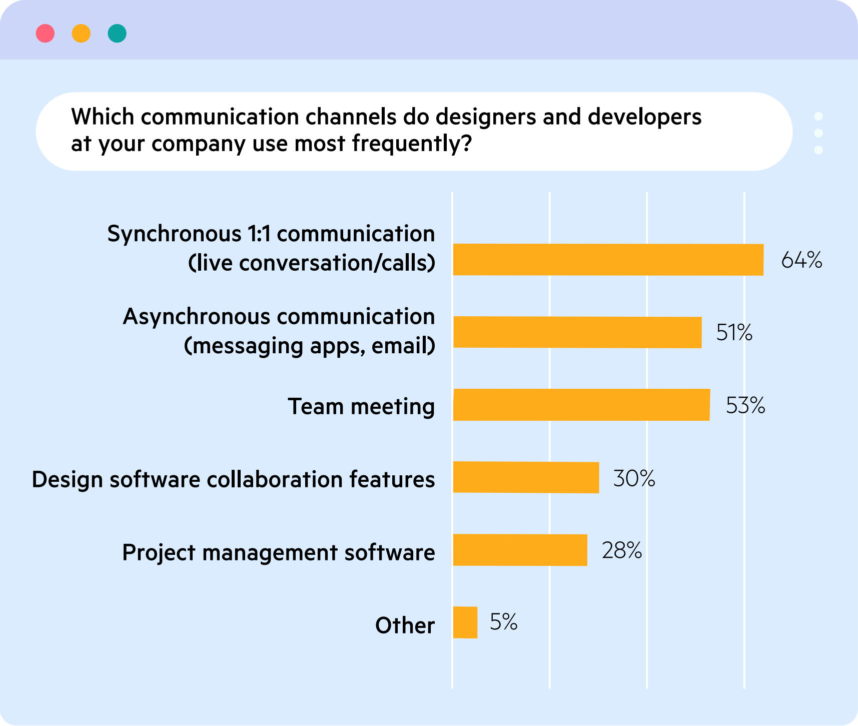
Designer-Developer Collaboration

Almost half of respondents report having a regular communication rhythm, with 46% of all teams collaborating daily – or at least a few times a week. However, 4% of collaborating designers and developers never communicate directly; rather, they use a mediator. A further 7% collaborate very rarely – just once a month or less! We all feel the pain of hours spent in meetings, but unfortunately, there's a rather high price to pay for overcorrecting and not meeting at all.



What media do designers and developers use most often to communicate? On average, each professional uses 2.3 different communication channels. Happily, 64% of them opt for synchronous one-on-one conversations—either in person or by getting on a call. We say happily because anyone who's ever read any book on relationships knows that talking to each other is step one in building a solid bond. Work relationships are no different.

Team meetings (53%) come in a close second place, followed by messaging and email (51%). Two free-form responses saying they use Azure Dev Ops to communicate piqued our curiosity. That's unique, isn't it? But our favorite response to this question is definitely: "The developers are the designers at our company thru [sic] direction of the IT Manager." We feel you, folks. We feel you.



The Good and the Bad

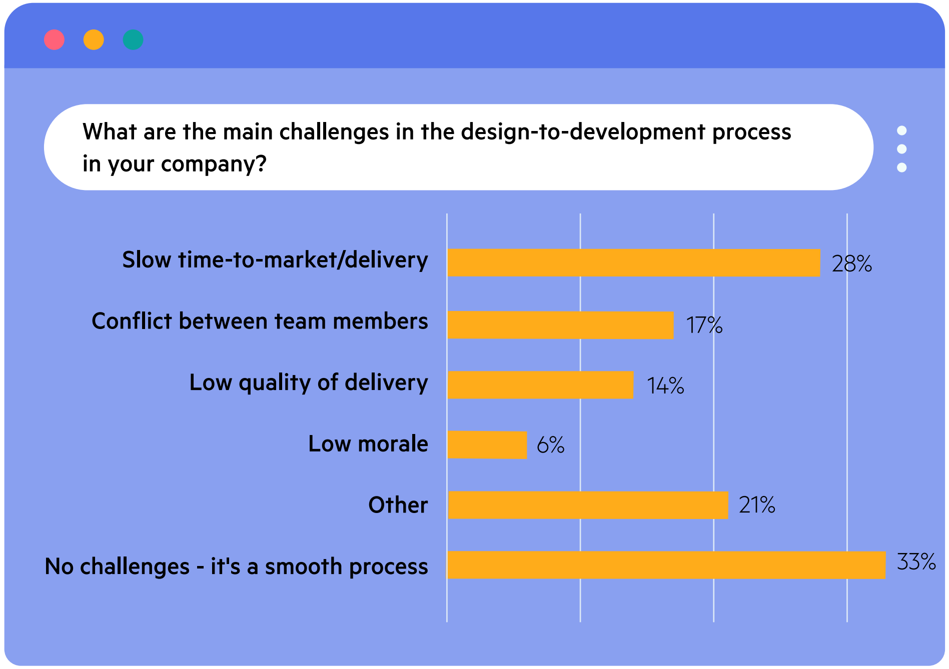
The good news is that **33% of respondents feel satisfied with their design-to-development process**—that's every third person! The bad news is that **at least 65% experience challenges** (we say at least because many of the free-form responses in the "Other" category also spoke about challenges).

And now we will find out why making sure that designers and developers collaborate well is business-critical:

> **The price many businesses pay for not establishing a smooth design-to-development process is slow time to market.**

Slow time to market is the price 28% of companies pay, but that's not all. 17% of teams suffer from **internal conflict**—and we all know that conflict at work is often resolved by people leaving the team or company. That, in turn, leads to all sorts of gnarly complications.

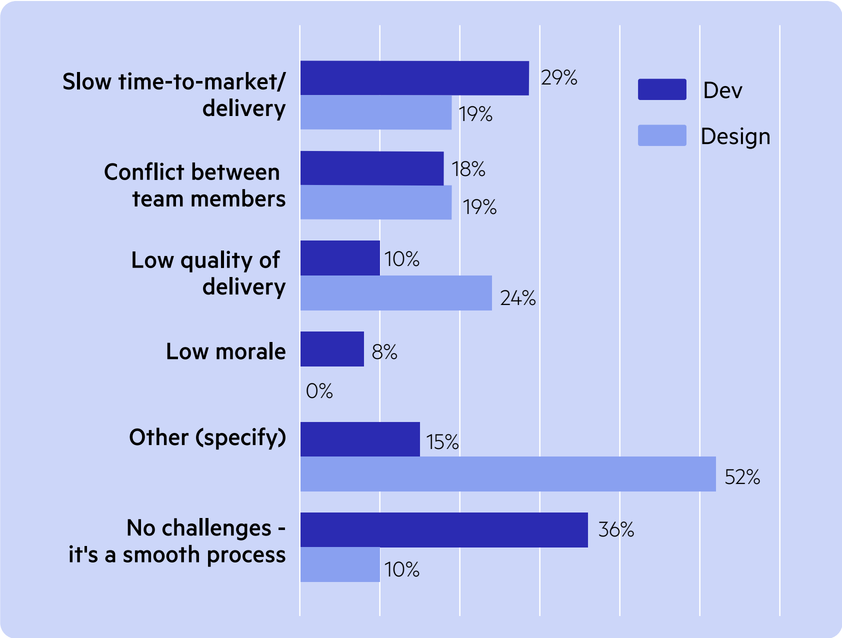
The third most prevalent consequence is **low quality of delivery** (14%). Some companies may be able to weather this (for a time), but for others, it's a death knell.


What are the main challenges in the design-to-development process in your company?

A “Developers are from Saturn, Designers from Neptune” Kind of Situation

It’s time to open Pandora’s Box now and look at the split responses of designers and developers. **While 36% of engineers say the designer-developer collaboration goes smoothly and without challenges, only 10% of designers would agree.**

Developers are 50% more likely to experience challenges around slow time-to-market, while designers are more than twice as likely to be concerned about low quality of delivery. Unfortunately, we didn’t include designers’ main concerns among the options we listed—52% of designers opted for “Other.” Among the free-form responses (modified for clarity):



“Siloed communications”

“Skills not up to date”

“Lack of communication between DEV team and UI/UX designer after initial hand-off to Dev team results in Devs making design decisions that negatively impact the UX”

“When the company hires an external agency, the internal developers have a tough time reaching out to the external designers as they rarely communicate”

“Insufficient understanding from higher management”

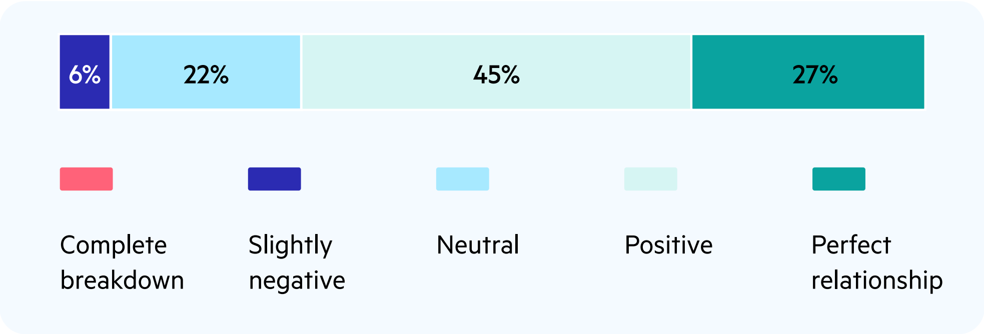
“At times the implementation does not reflect the design”

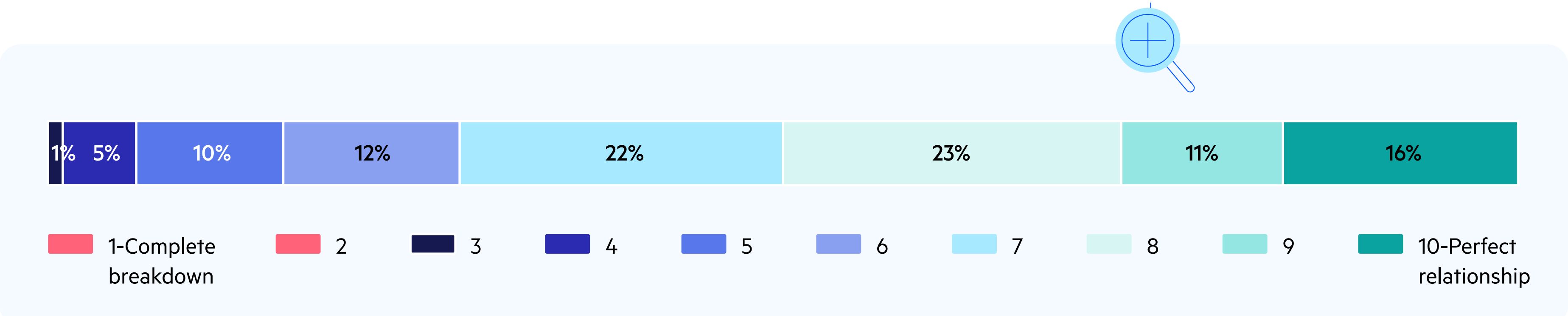
Corresponding to the 33% of people experiencing no design handoff challenges, 27% of all respondents report a perfect relationship between design and development on their project (a rating of 9 or 10 on a 10-scale). A further 45% are quite satisfied (a rating of 7 or 8).

The “meh” zone is also significant, with 22% going for a 5 or a 6. Sounds innocent enough, but if your partner gave your relationship a 5 or a 6, you’d be on high alert, wouldn’t you?

Just 6% rate their relationship as bad (3 or 4) and no one is in the “complete breakdown” zone. Phew! (And yes, we’re all thinking it: those folks do exist, but they are probably way too busy struggling to have time for a survey).

If 10 is a perfectly functioning relationship with nothing to add and 1 is a complete breakdown where no one talks to the other, how would you rate the relationship between designers and developers working on the same project?

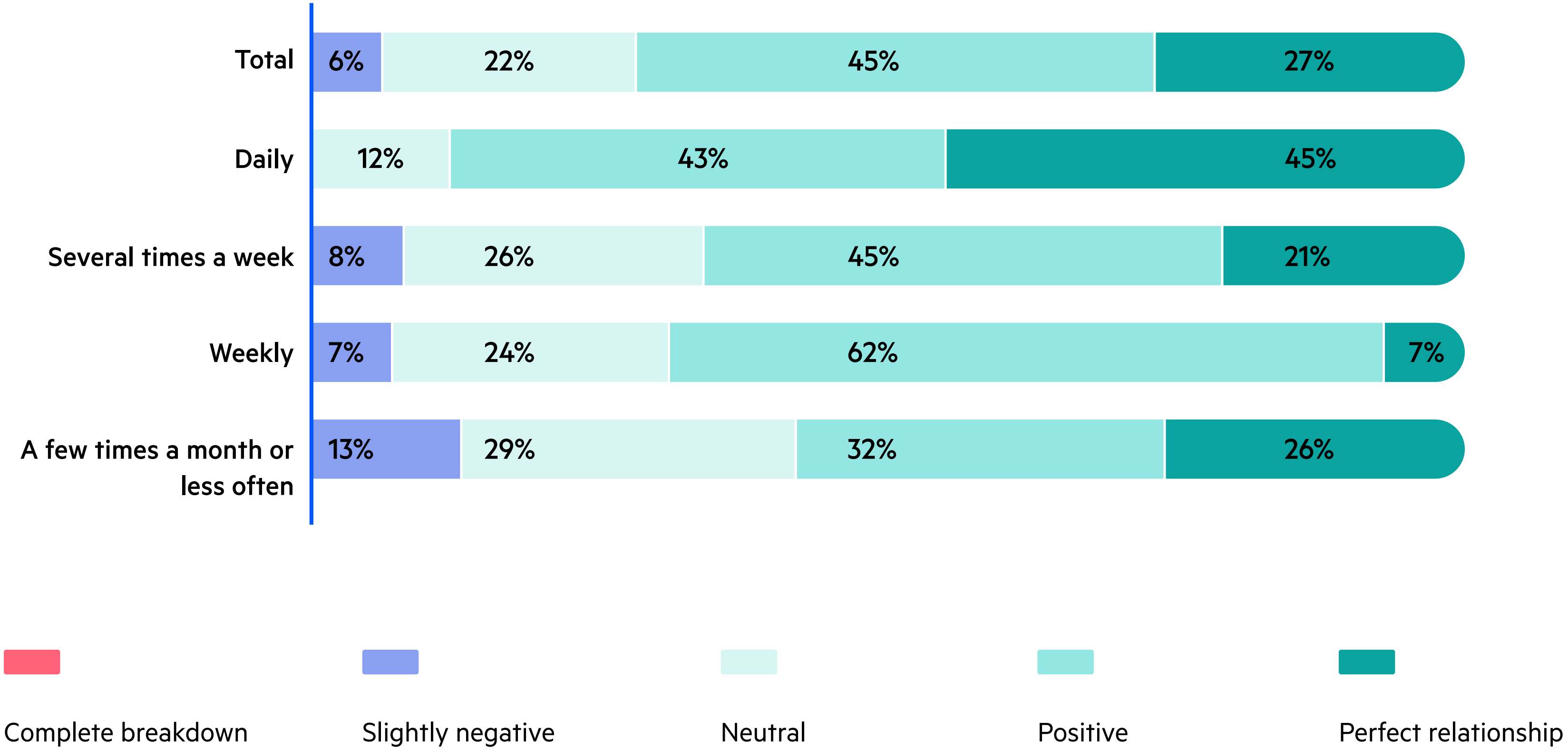




A Chat a Week Keeps the Designer-Developer Relationship at Its Peak

Now that we know how designers and developers feel about their relationship, let's go back to that interesting question we left open a few paragraphs earlier: what's the price of design and development not collaborating regularly? There is a clear correlation – work partners who communicate regularly (at least a few times a week) have stronger relationships. 45% of those who evaluate their relationship as perfect (9 or 10) report communicating daily with their partner on the other side of the fence. What's more, none of the daily collaborators report a poor relationship (4 or lower).

If daily catchups seem excessive, don't worry – once a week may be the sweet spot for collaboration. 69% of respondents who chat once per week report that their relationship is positive (7 or higher) and only 7% rate their relationships as poor.



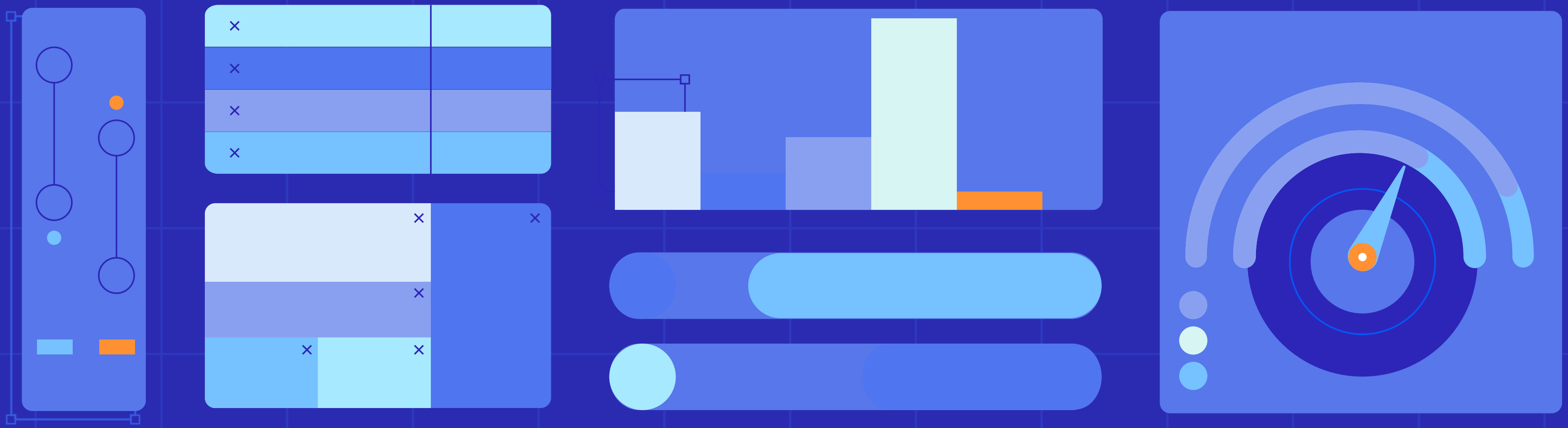
We followed up with those who rated their relationship 6 or lower and asked them what they thought would improve it.

Managers, leaders and business owners, note this:

The top three interventions your team yearns for are

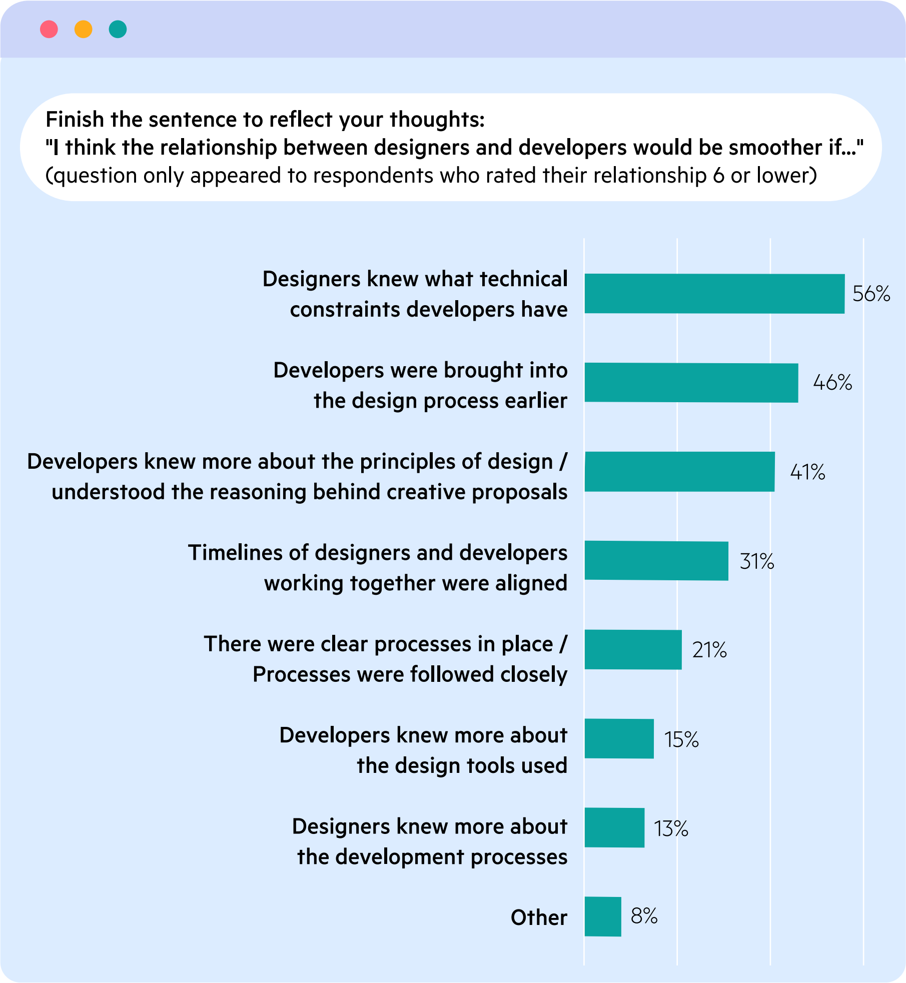
1. **For designers to be trained on the technical constraints developers work with**
2. **For developers to be brought into the design process earlier**
3. **For developers to learn more about the principles of design**

No operations consultants or island retreats are required; you can remedy the situation by helping designers and developers learn more about each other's processes and encouraging them to speak more often. How beautiful in its simplicity is that?

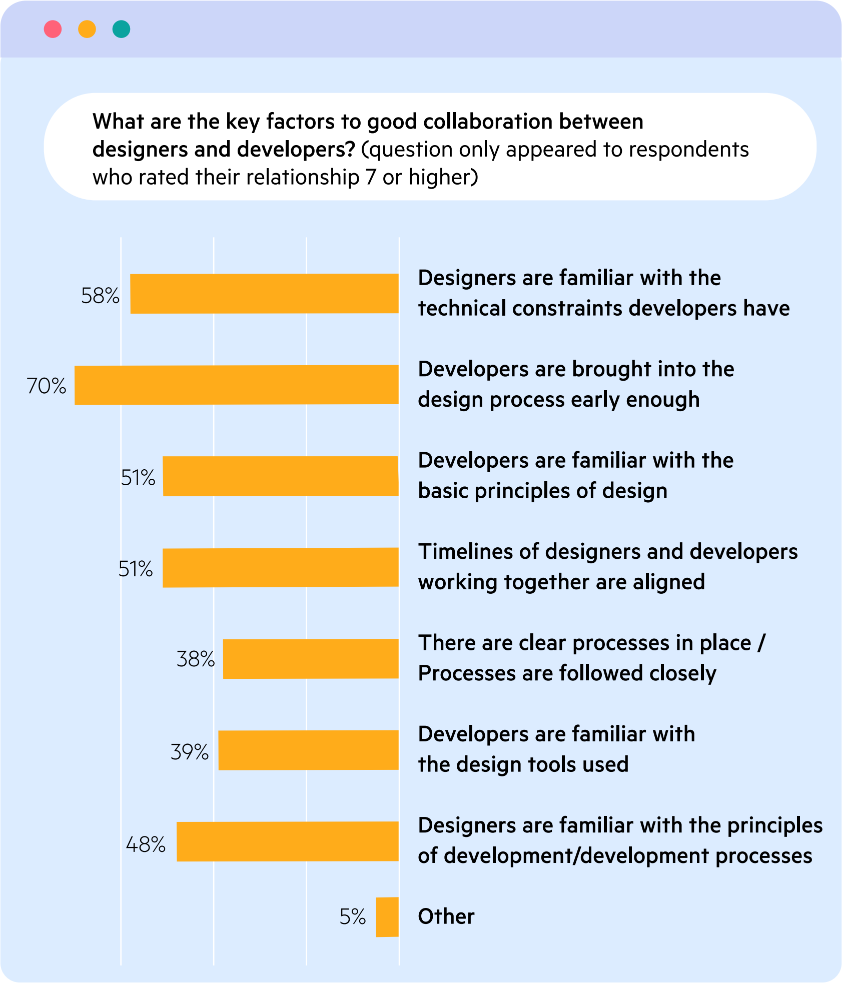


We also followed up with those who rated their relationship 7 or higher and asked them what they thought worked so well. Sneakily, we gave them the same options as the folks who rated it low. As expected, the results are mirror images. What people in poor relationships crave for is what those in good relationships give credit to for their success.

A couple of people brought up a key aspect of good collaboration we hadn't thought of: "Skills on the team should match the goals & requirements." Bullseye, isn't it? We're all now thinking of that colleague whom we constantly had to train on-the-go even though they were supposed to be the same level as us. If this isn't a cause of friction!



Finish the sentence to reflect your thoughts:
"I think the relationship between designers and developers would be smoother if..."
(question only appeared to respondents who rated their relationship 6 or lower)

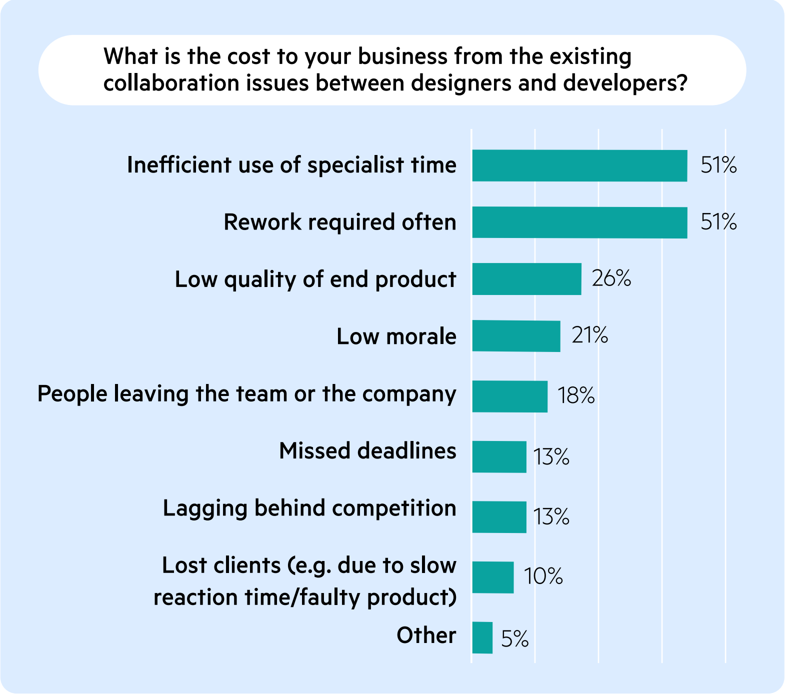


What are the key factors to good collaboration between designers and developers? (question only appeared to respondents who rated their relationship 7 or higher)

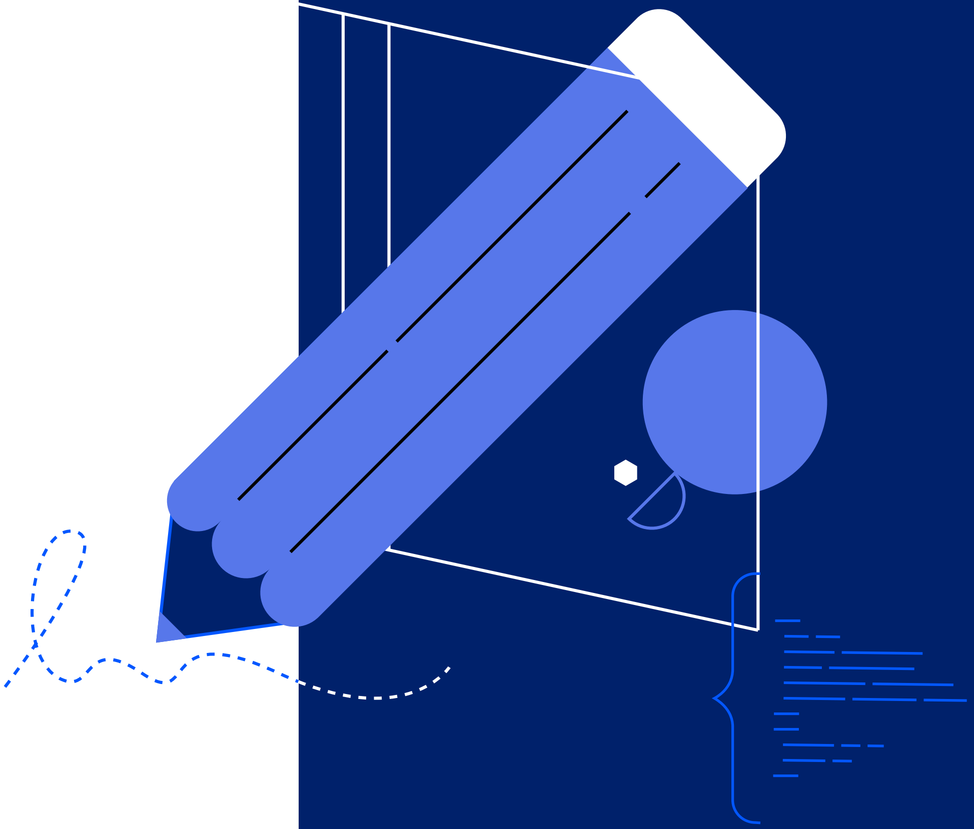
What's the cost of inaction, you ask? So did we.

The top three reported responses about easing the designer-developer relationship all had to do with developers learning more about the goals and processes of their designer counterparts. Whether that's understanding the tools designers are using, the reason why they're making certain design decisions or just becoming an active participant in the design process, it all (clearly) adds up – in this case, to over 50% of issues!

When these things aren't happening, you're not just dealing with a morale issue, you're also wasting valuable time when designers need to re-explain and push back on development solutions that won't work. When both parties fully understand the problem they're solving—and the non-negotiables—working out a compromise is much faster and easier.



If you're a developer looking to get more involved in the design process, check out our free eBook: Foundations of Design for Developers. It will take you through the foundations of design and all the related terminology, giving you a serious step up when it comes to collaboration.



Now the design-development divide in responses hints at some interesting dynamics. Unfortunately, the design side of the data is not statistically significant as many of them had dropped off by this point in the survey. However, it's still super interesting to consider the differences that popped up (consider this anecdotal evidence):

- Many more developers than designers thought bringing in developers into the design process earlier would help. Are designers being territorial or is this based on experience? We'll ask in next year's survey.
- No one's pointing fingers, but designers and developers both think the other should level up their knowledge of the other side. Honestly...maybe they're on to something. More dev knowledge for designers and more design knowledge for devs!
- Designers seem much more convinced that the issues come from misaligned timelines.

An insightful free-form response we got:

"The design tools don't produce things that are usable (...) in the dev tools"

This is a problem we've recognized that many, many teams have. We spoke to lots of them and devised a solution. Learn more about how our family of design- and developer-friendly tools give developers and designers a common language and bridge the gap.

Factors Correlating with the Quality of Designer-Developer Relationships

We cross-sectioned many of our findings with this question and interesting patterns kept emerging: “If 10 is a perfectly functioning relationship with nothing to add and 1 is a complete breakdown where no one talks to the other, how would you rate the relationship between designers and developers working on the same project?”

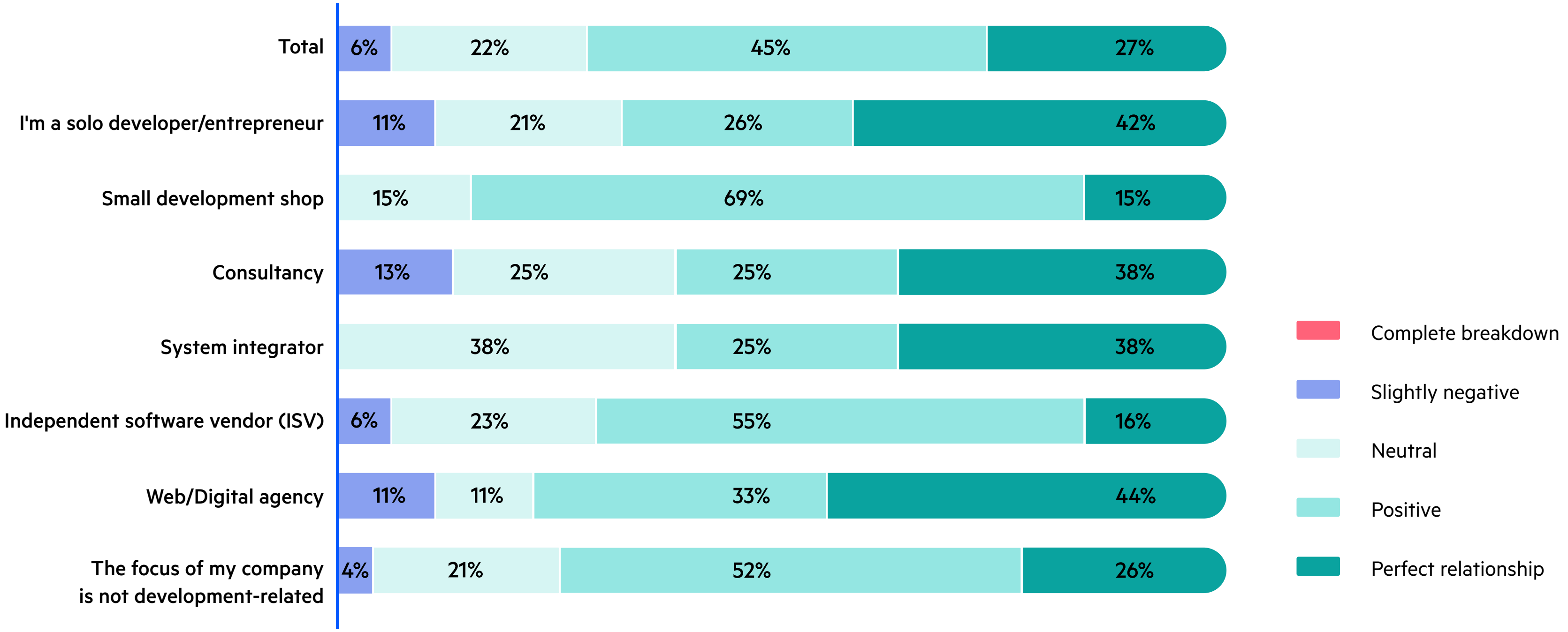
So many interesting observations, in fact, that they earned a section of their own.

1. There’s no one type of company that creates particularly terrible conditions for designer-developer collaboration. So far, so good.

Cross-section:

“Which of the below best describes your company?”*

*Please note that the only statistically significant data in this chart is for people who answered, “The focus of my company is not development-related” and “Independent software vendor (ISV).” We’re publishing the full cross section for completeness and as anecdotal evidence.

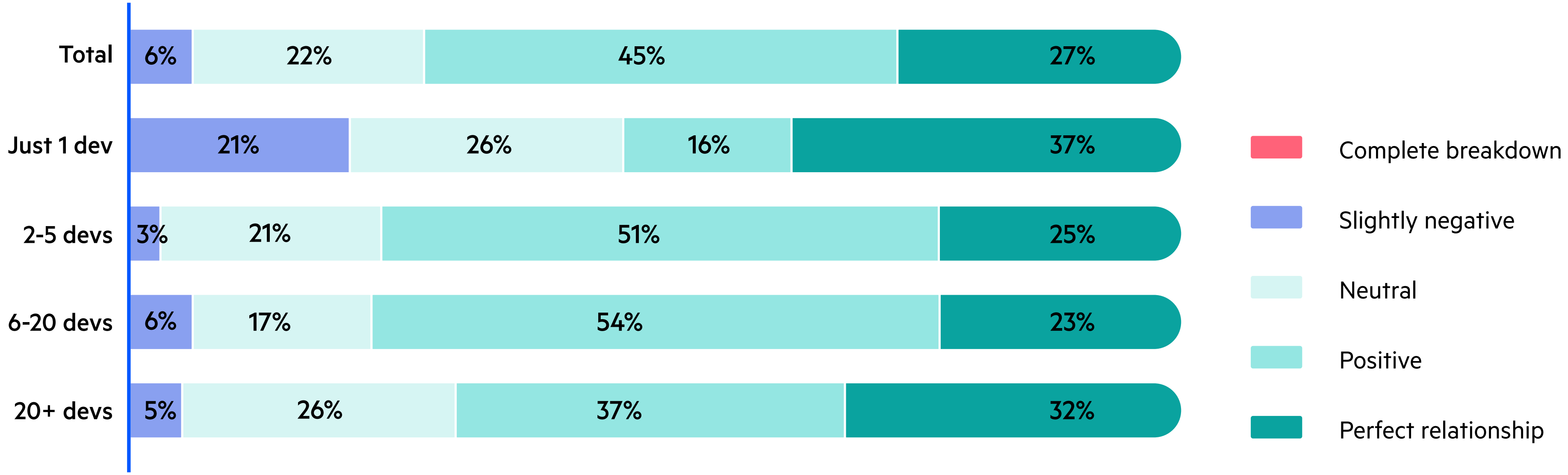


2. Designers working with just one developer are the most disgruntled kind of folk (47% of responses of 6 or lower). Curiously, this is also the situation in which the biggest percentage of designers thrive (37% report a perfect working relationship). Perhaps that makes sense; after all, a one-on-one working relationship is highly dependent on the personalities and preferences of both individuals. When it's good, it's really good—but when it's not, it's a problem.

Working with more developers seems to be the preference of most designers. However, we also wonder how much of that is simply the standard everyone has become accustomed to—after all, developers outnumber designers in most companies. This ratio of 1 designer to 2-5 developers is the most common, so the positive ranking here may be reflecting the respondents' average work environments.

Cross-section:

"How many developers are working on your current project/product?"

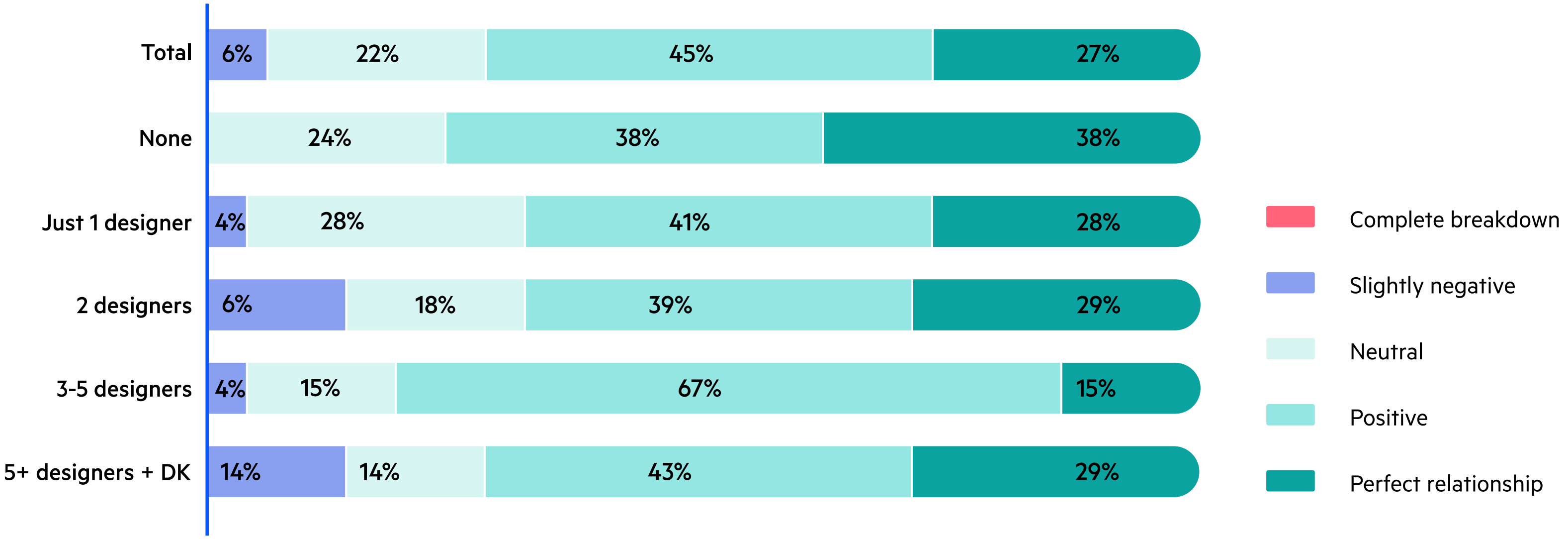


3. Developers working with 3-5 designers are the most cheerful kind of folk (82% of responses of 7 or higher). Perhaps having a few designers on a project helps balance things out—after all, with just one, it can be easy to dismiss suggestions as "just their opinion," but a united front can feel more valid.

Evidently, 2 designers aren't enough and 6 is too many! If you have thoughts or observations on the oddly specific golden middle of 3-5 designers per project, reach out to Kathryn Grayson Nanz on X and let us know. We're open-sourcing the analysis of this data point.

Cross-section:

How many designers are working on your current project/product?

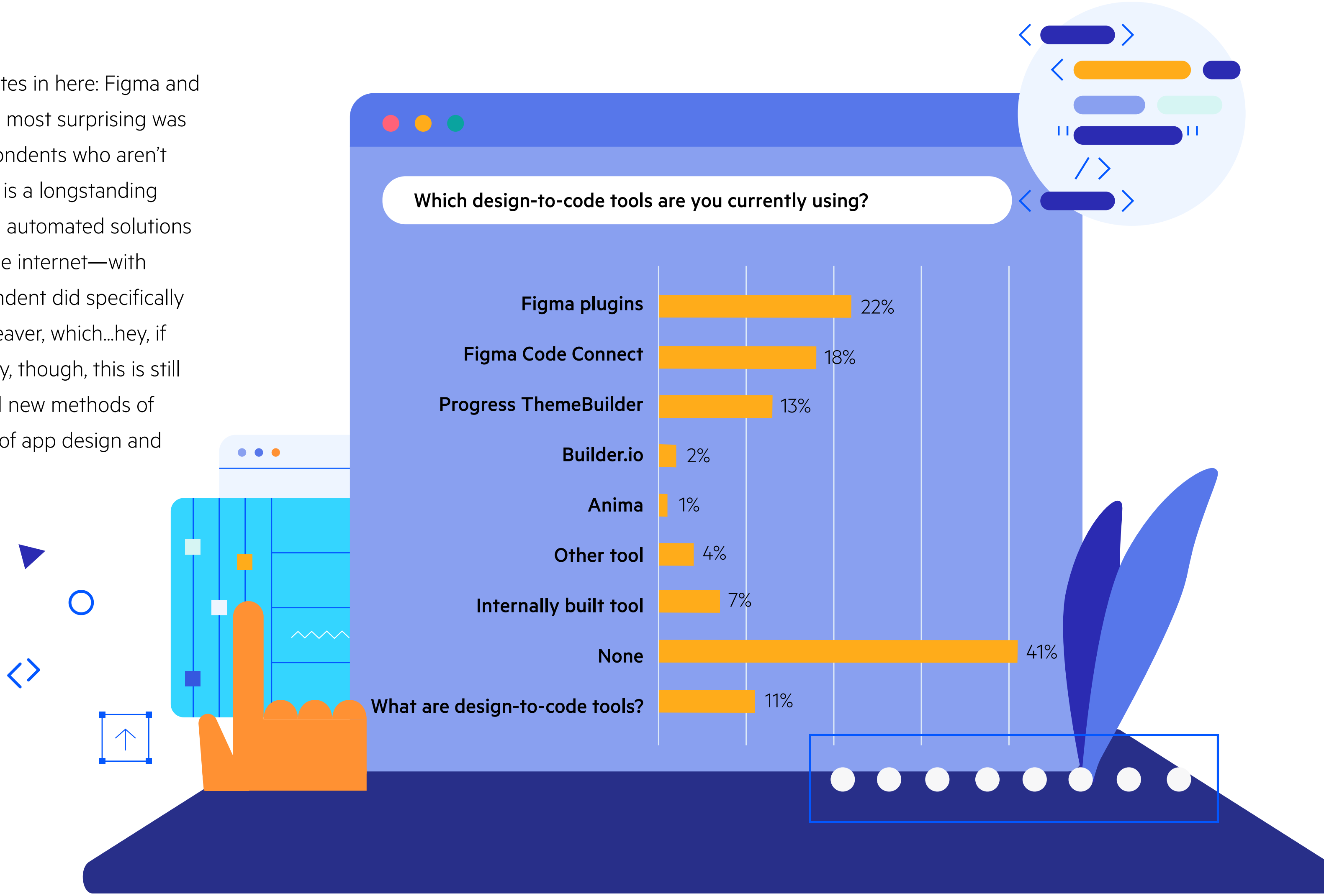


Clarification: You may wonder why developers working with no designers (responding with "None") in this chart give the highest relationship rating. This comes from the phrasing of the questions we asked. We asked, " How many designers are working on your current project/product?" and, "If 10 is a perfectly functioning relationship with nothing to add and 1 is a complete breakdown where no one talks to the other, how would you rate the relationship between designers and developers working on the same project?" So, we have developers who are not currently collaborating with designers but have had a predominantly good experience so far. It's true what they say, sometimes you only realize what you've had once you lose it.

Tooling

Now, for the extra fun part—because who doesn't love to geek out about all the cool tools available for design and dev work these days?

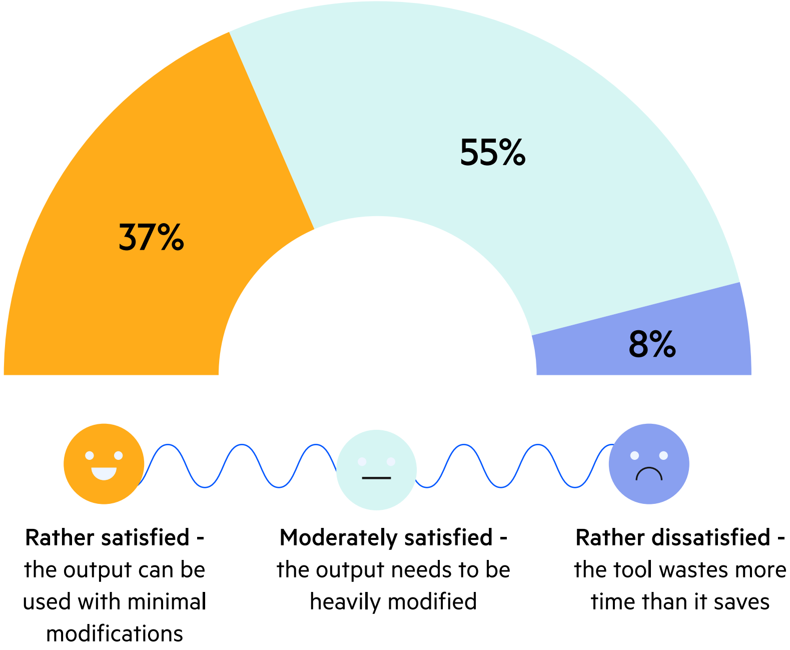
As expected, we saw some fan favorites in here: Figma and Progress ThemeBuilder. But perhaps most surprising was the significant portion (52%) of respondents who aren't using anything at all! Design to code is a longstanding problem that many have tried to find automated solutions for, pretty much since the dawn of the internet—with varying levels of success. One respondent did specifically mention they are still using Dreamweaver, which...hey, if it's working for you, godspeed. Clearly, though, this is still a space that's ripe for innovation and new methods of problem-solving for our modern era of app design and development.



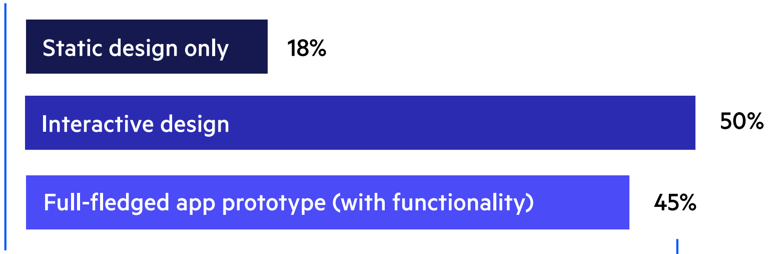
Jumping off that, about half of those folks said they were "moderately satisfied" with the output of those tools, meaning that the output is usable but still requires heavy modification before it is production-ready. Again, this sounds like a space just itching for better solutions.

We were also curious what designers were building with their tools, so we asked about the complexity of their prototypes. Most are building highly interactive interfaces—with a smaller percentage sticking to static designs only. Makes sense, especially for folks building full, complex applications vs. landing pages or simple websites. While a fully interactive prototype is more of an upfront investment, it enables designers to do user testing and get helpful, high-quality feedback—all before a single line of code is written.

? How satisfied are you with the output of your design-to-code tool?



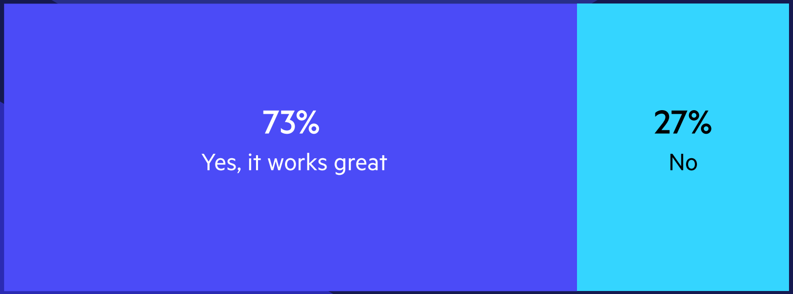
? How complex are your prototypes?



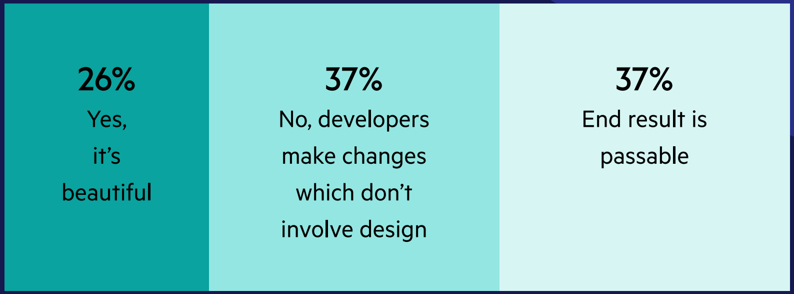
But, of course, the million-dollar question is whether or not they're happy with the finished product. While most are okay with their internal processes, unfortunately only about 1 in 4 are satisfied with the finished work—the rest feel it's not good enough or just "passable."

Perhaps this reflects how we phrased the questions. Looking at the data, it seems that when the only options are "absolutely yes" and "absolutely no," more folks are the glass-half-full kind of people who will round up into a "yes." However, when presented with the "okay/ passable" option in the final product question, we start to see a little more nuance.

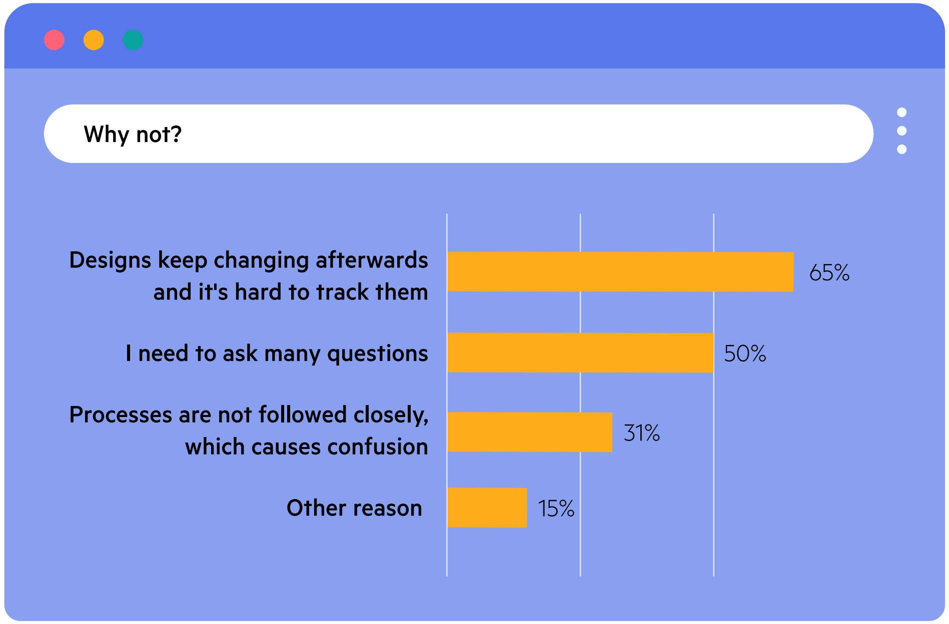
Are you satisfied with the design handoff for your project/product?



Are you satisfied with the way design is implemented?

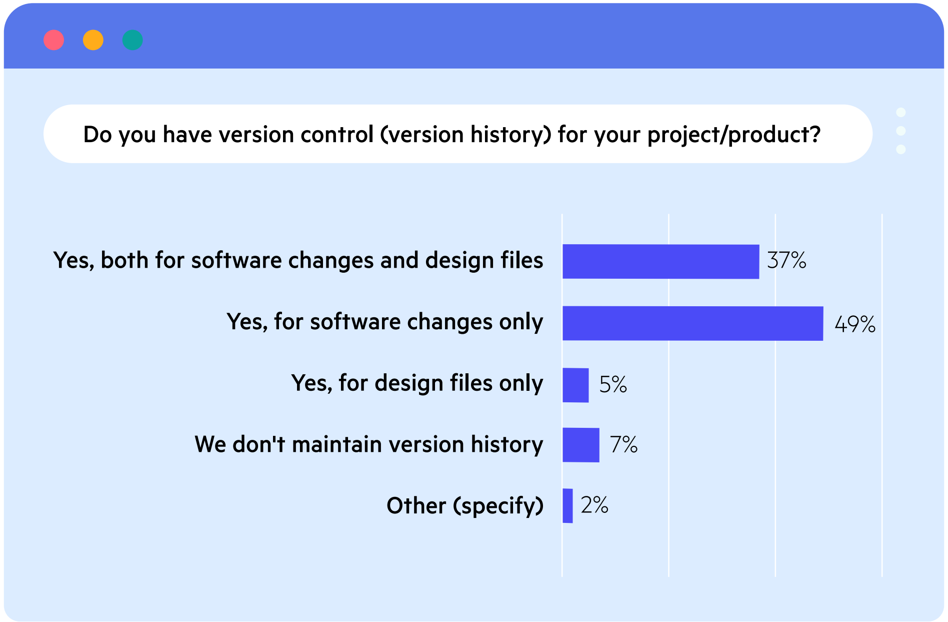


When we dig into why there's dissatisfaction, it seems to boil down to two main pain points: lack of clarity in the initial doc/file leading to lots of follow-up questions (50%) and changing requirements on the design side (65%). Change tracking in design software has always been a struggle – especially for developers, who (generally) are accustomed to the far more granular style of version control systems like git. Lack of clarity could reflect a few things; maybe more detailed notes and files are needed, or perhaps this is one of those problems that calls back to the lack of developer understanding of design basics. What might look clear and obvious to a fellow designer is not necessarily so to a developer without the same foundations and knowledge base.



Other interesting open-response answers from this section included, "Accessibility issues are often overlooked" and "Designs just don't work in the real world"—both cases where something that looks great "on paper" just doesn't quite align with the practical requirements.

Speaking of version control, we asked folks about how that looks for them on both the design and dev side of things.

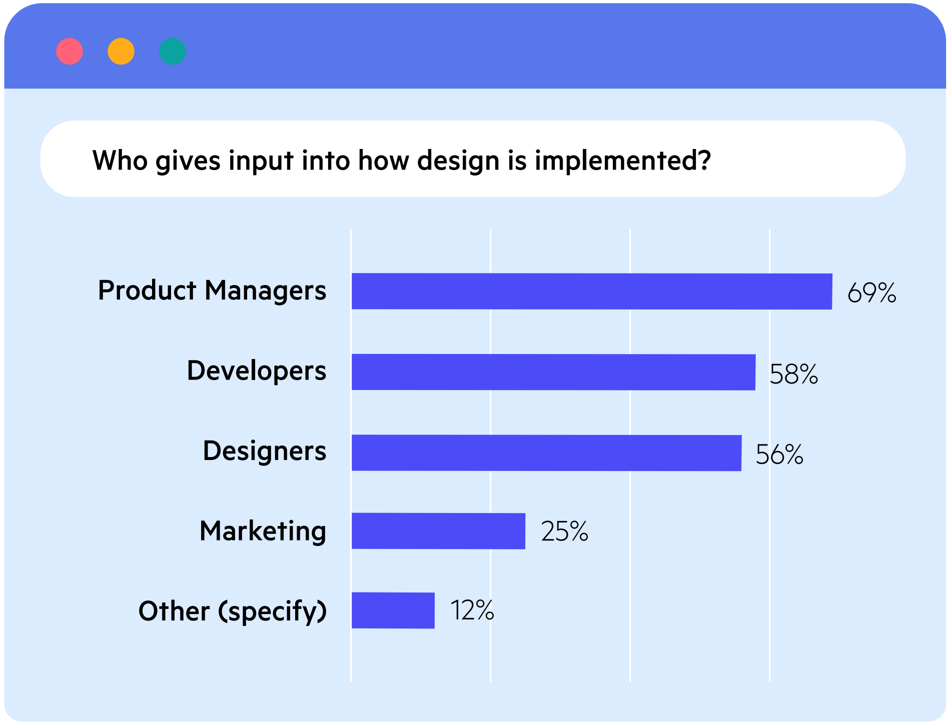


As noted earlier, pretty much everyone is using version control for their software...but it's a little more hit-or-miss on the design side, where that's a (relatively) newer concept. At least, assuming we're talking about something a little more standardized than the infamous "design_file_v2_FINAL_FINAL_FINAL_NO_REALLY" naming systems that almost every one of us has been guilty of at some point in our lives.

Implementation

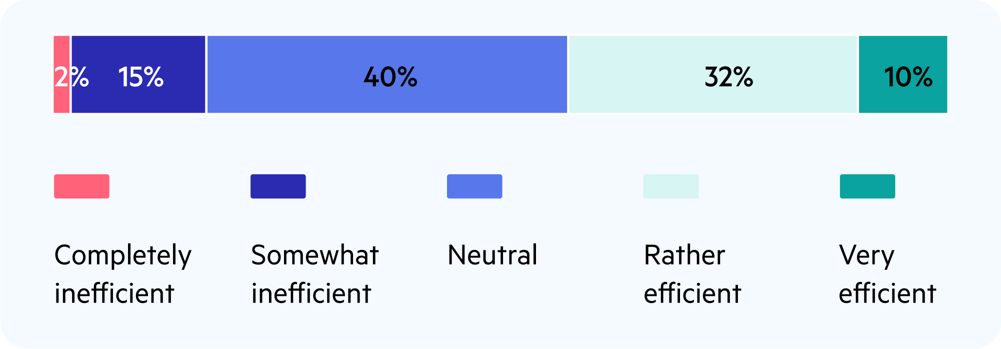
No matter what tools we're using to get there, the end goal is a beautiful and functional finished application or website. While that's awfully easy to type, it is much harder to accomplish. So, let's look at what goes into getting from concept to production.

First, who gets to have a say in how the design is implemented? The results show a fairly even split between designers, developers, and product managers—with just a handful including marketing as well. If we had to guess, this is likely a reflection of the industry where the respondent works. Folks in advertising agencies may see higher involvement from marketers than those building internal apps at a large enterprise company.


Who gives input into how design is implemented?

In terms of efficacy, most folks fall right down the middle here: not amazing, not terrible, just… allright. When we look at the cross-section, there's a clear positive correlation between design implementation efficiency and successful collaboration. With that in mind, "neutral" is starting to look not terribly neutral anymore – how much better could collaboration improve the final, implemented design?

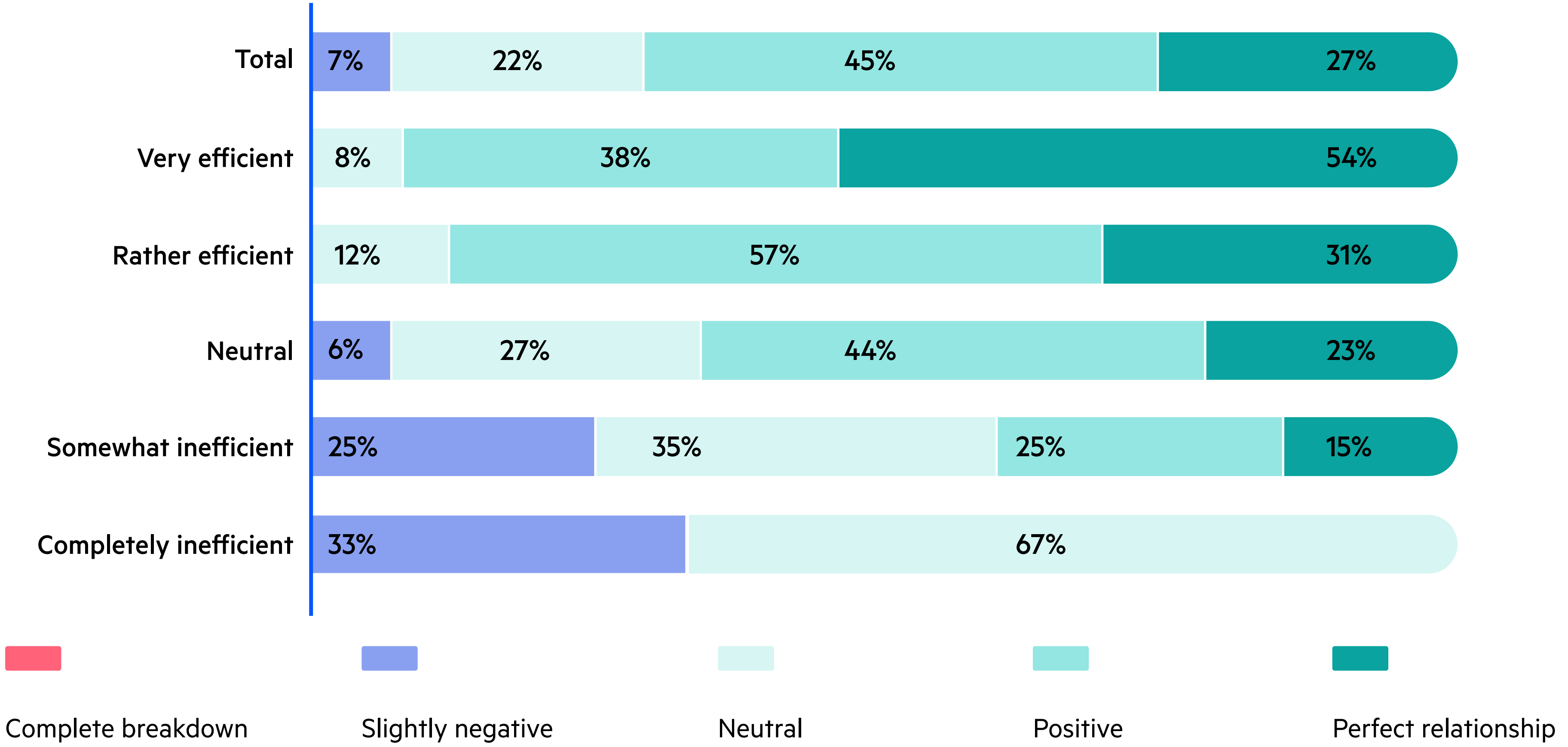
How would you rate the efficiency of the design implementation process?



How would you rate the efficiency of the design implementation process?

Crossed with

If 10 is a perfectly functioning relationship with nothing to add and 1 is a complete breakdown where no one talks to the other, how would you rate the relationship between designers and developers working on the same project?

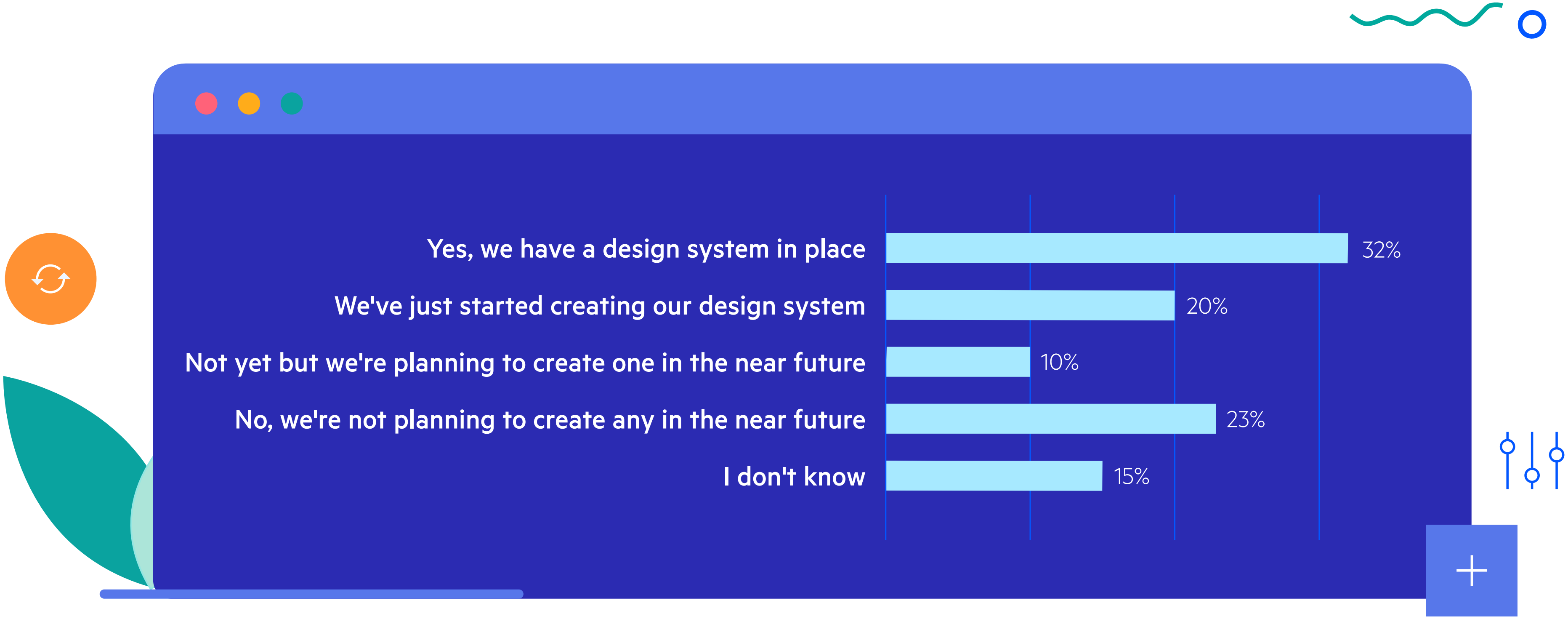


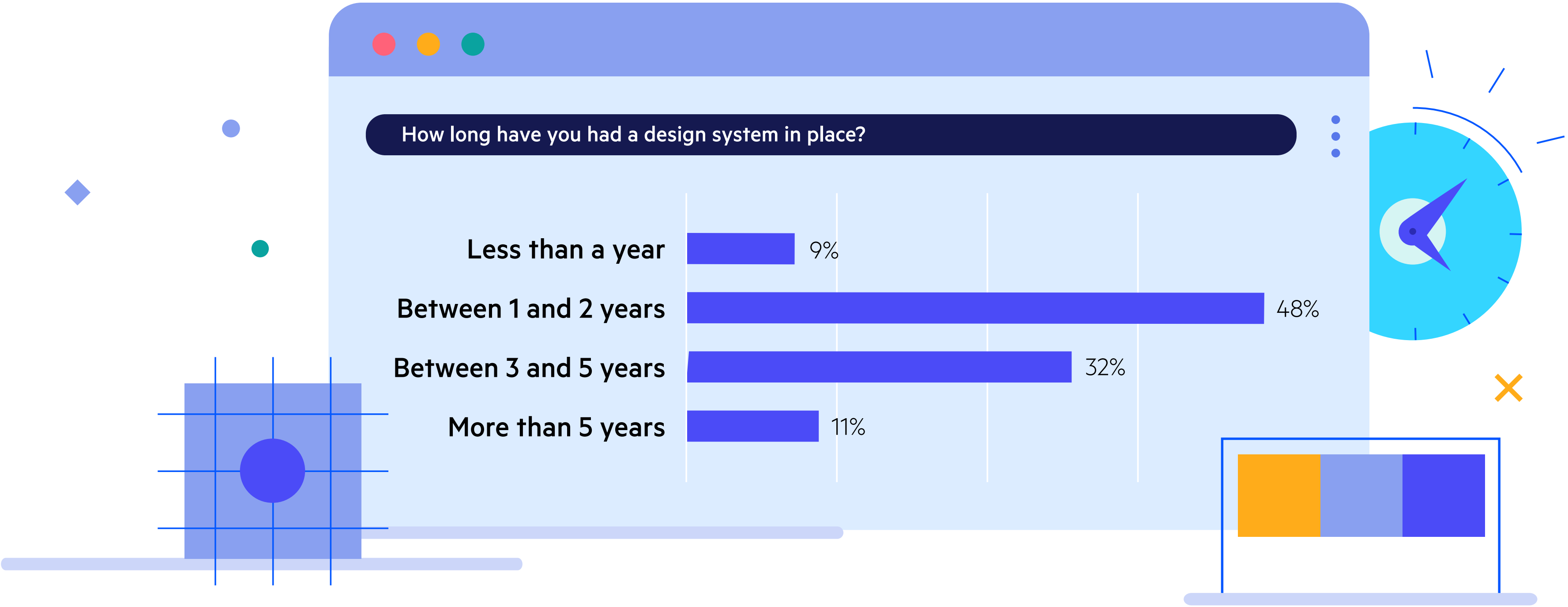
Design Systems

For our final section, of course we would be remiss if we talked about the designer-developer relationship and didn't ask about design systems. After all, a true design system is a blend of both worlds: a place where design and dev naturally overlap. We don't think there's been either a design or development conference in the last decade where there hasn't been at least one talk about design systems. Let's dig in:

Design Systems 101

First, the obvious: do you have a design system?





About half of all respondents did—whether it was one they had just started or a more mature and established system. Of those who didn't, about one-third were planning to create one in the future. We did find it interesting that 15% of folks simply...didn't know! Our guess would be that those are developers that work in rather siloed workplaces and don't have a lot of communication with their designers, which is a shame.

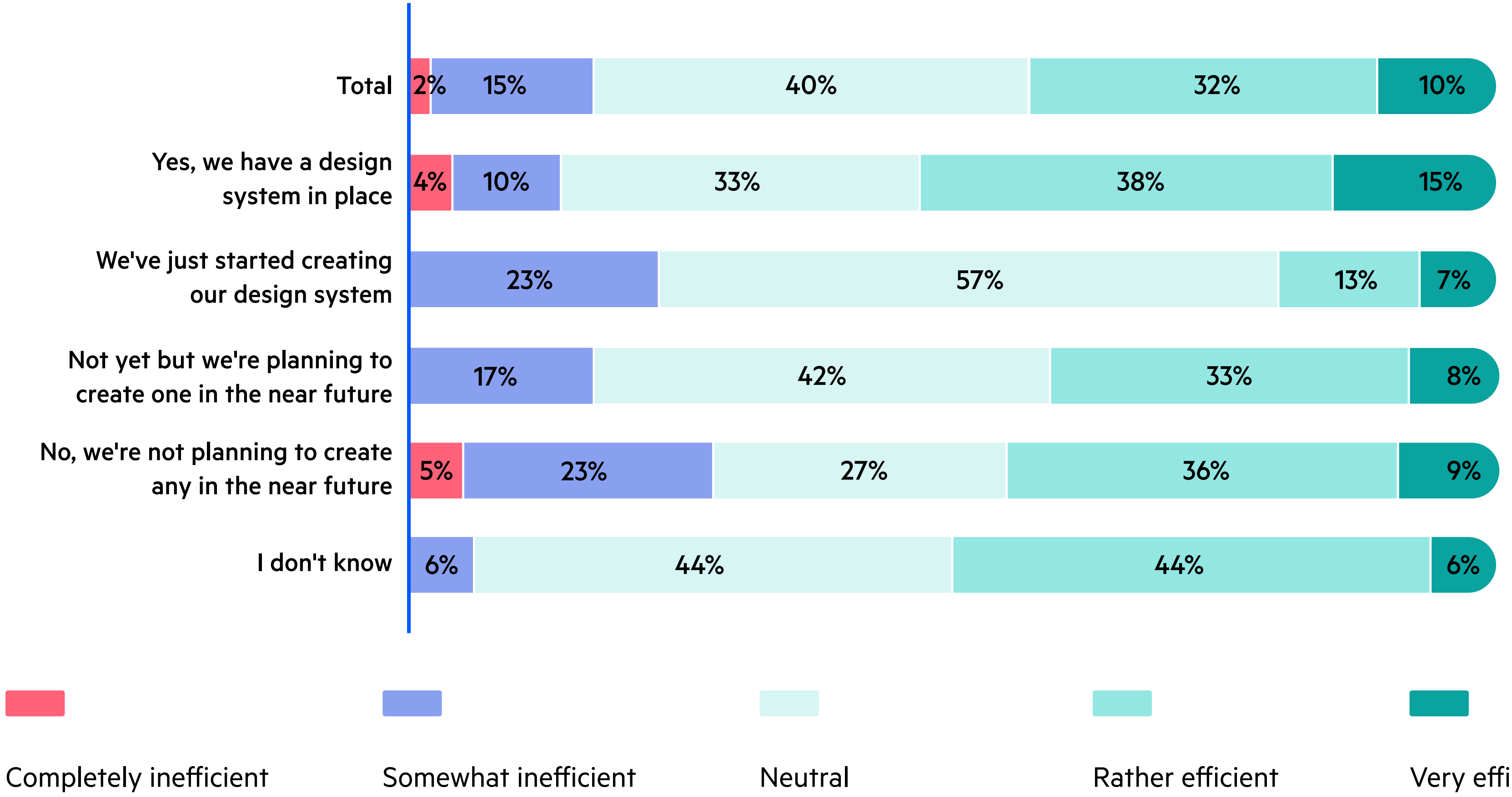
> A good design system makes everyone's lives easier!

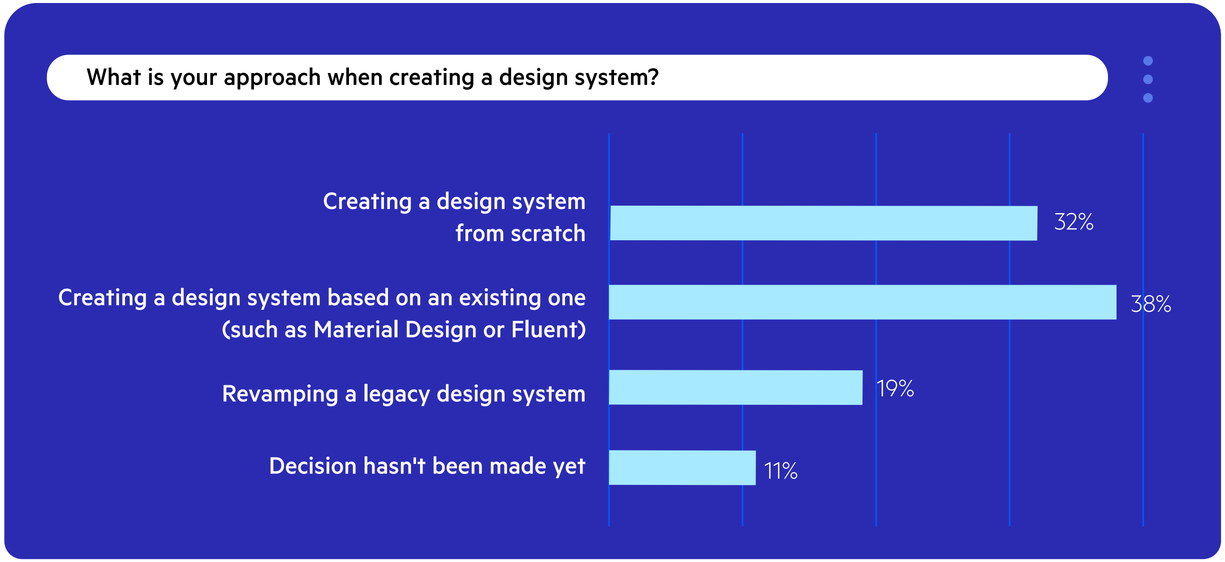
The survey results show a nearly even split between those who had recently created design systems (less than 2 years) and those with more mature systems (3-5+ years). In fact, when we looked at the cross-section, we saw that having a design system makes design implementation more efficient, and the effects grow stronger with time. It seems like a good sign that so many folks are finding enough success to keep their systems going for so long!

Do you have a design system in place?

Crossed with

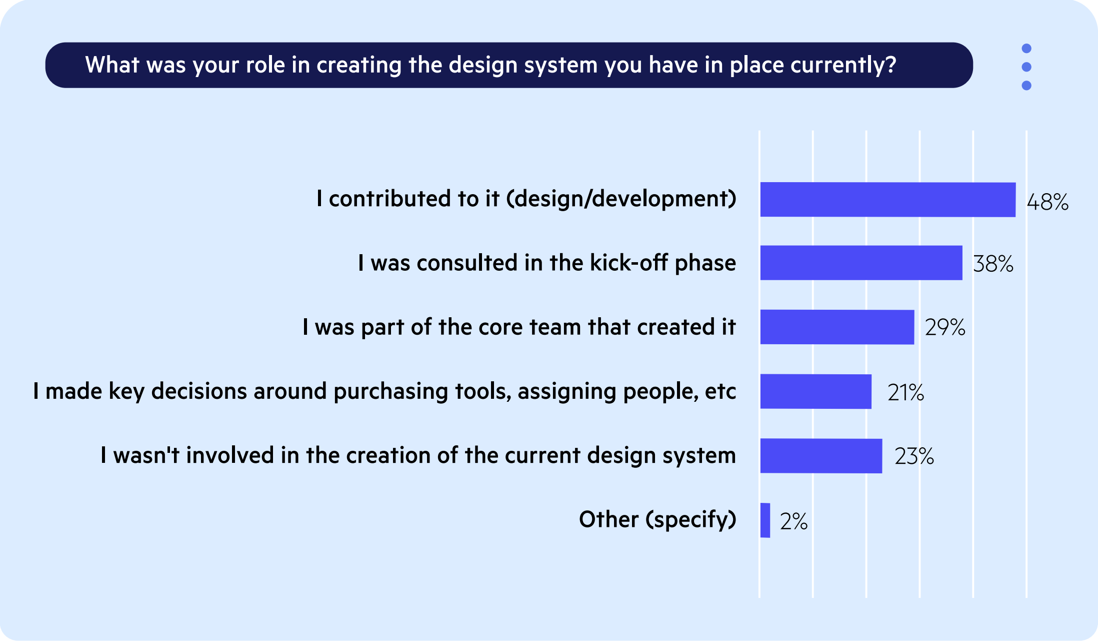
How would you rate the efficiency of the design implementation process?



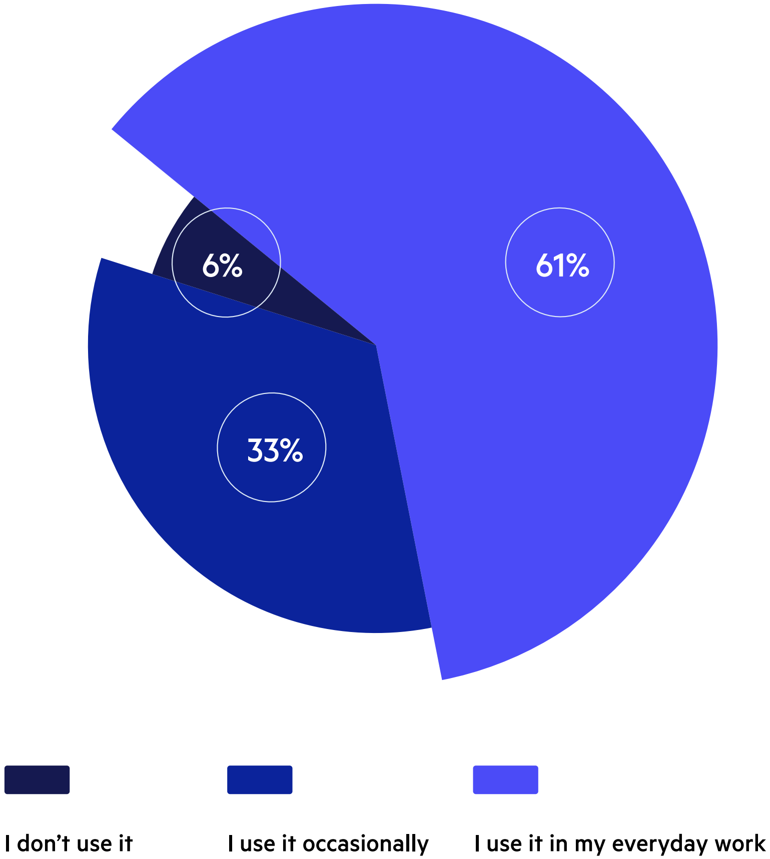


Of the folks that had or were planning to create a design system, most were working from the basis of a larger, established design system (like Material Design or Fluent). This makes a ton of sense – after all, why reinvent the wheel? Personalizing an established system can give you a huge jump start if you don't need to make significant changes. Of course, if you do need something extremely custom, there's a high investment – but also a huge benefit if you build from scratch.

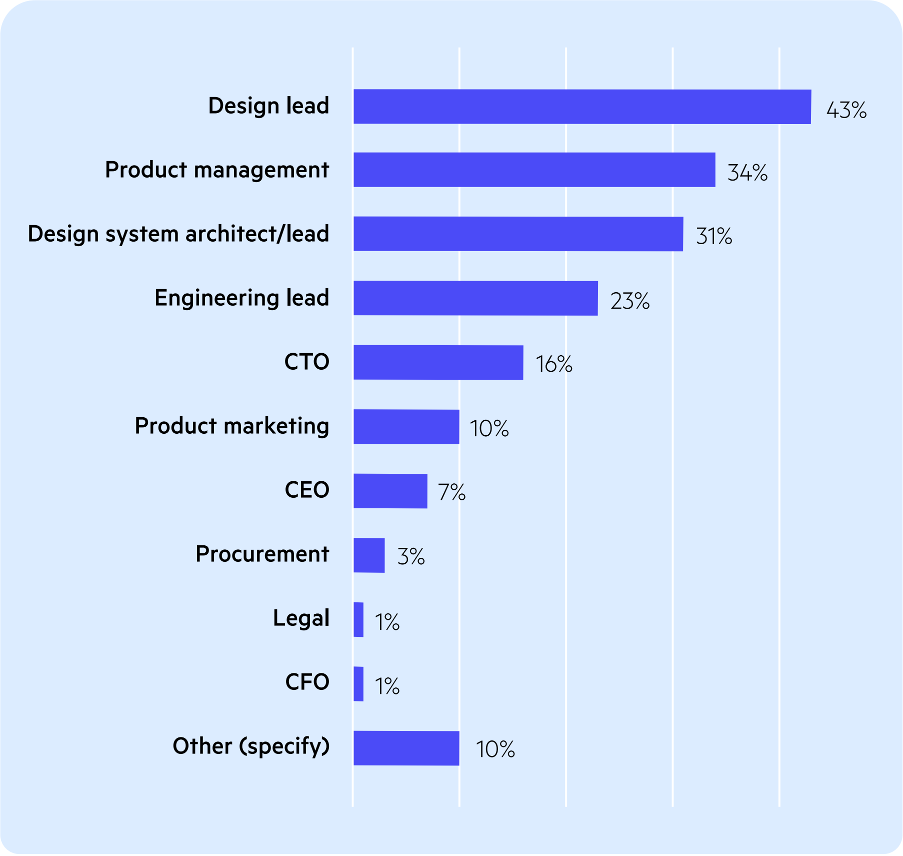
Naturally, we were also curious about each respondent's role in shaping and using their design systems.



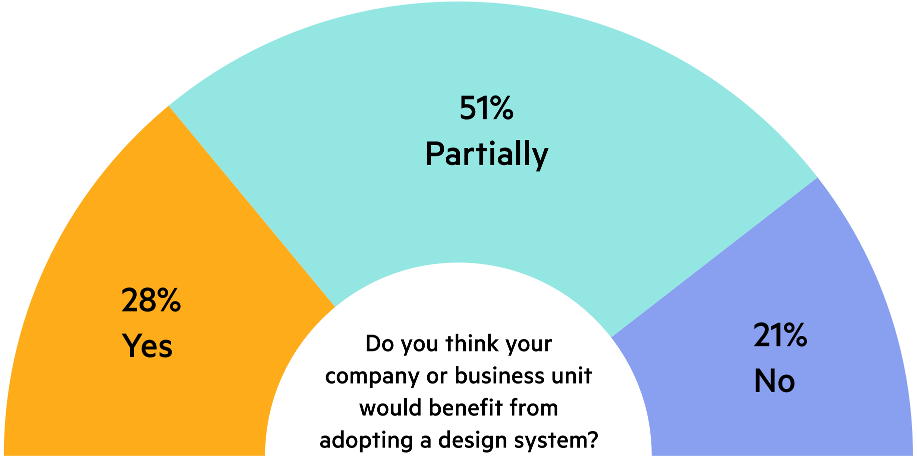
How often do you use the design system you have in place currently?



Who decided that you'll be creating an internal design system?



Most folks who made it this far in the survey were either direct contributors or consultants in some way to the creation of the system (whether that was in the past or actively ongoing), with 23% not involved at all. Similarly, most of the respondents with established design systems used them regularly in their day-to-day work, with only 6% not using them. However, when it came to who got the final say on design system creation, it was overwhelmingly the design leads – with almost half of respondents saying that the Design Lead or Design System Lead was the one who decided to create the system.

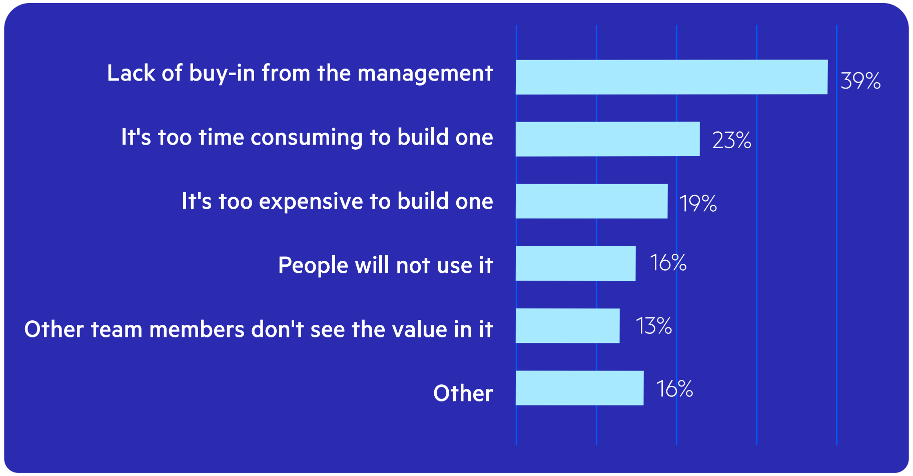


For those without a design system, the vast majority (79%) felt like it would either somewhat or definitely benefit them to adopt one. However, we all know that's easier said than done…

If you are creating, revamping or updating your design system, the Progress Design System Kit gives you the tools and resources you need to implement your own design system in drastically less time. It will save you months of work as compared to building everything from scratch.

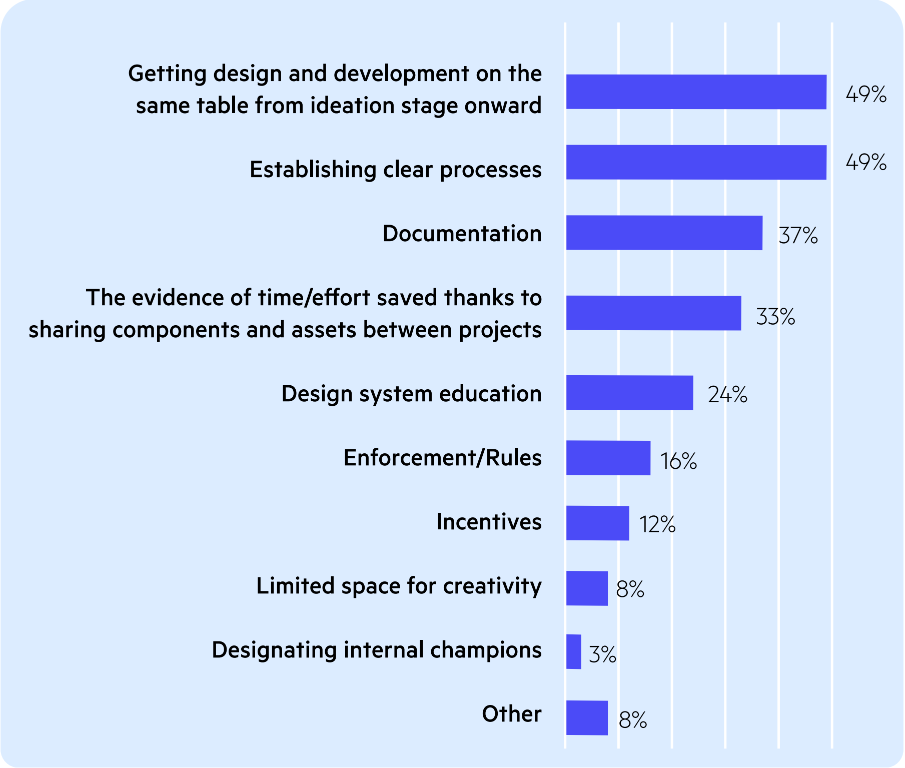
Design System Challenges & Benefits

? Why don't you have a design system in place if you think your company would benefit from it?



This next set of questions sheds some light on the primary inhibitors to design system creation. While resource dedication (time, energy, finances, etc.) all play a role, lack of buy-in at the management level was reported as the primary factor standing in the way.

? Which of these factors impacted the adoption and usage of design system the most in your company?

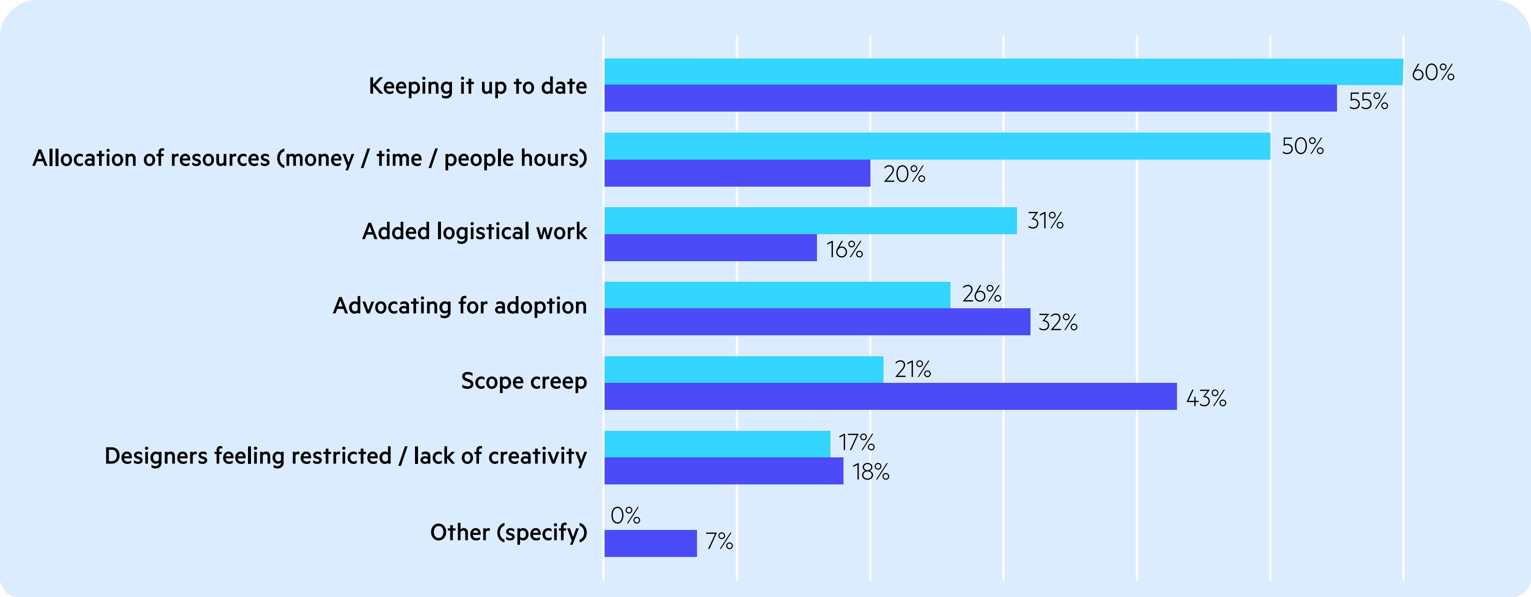


Of course, once you've cleared the hurdle of creating the design system, that doesn't mean it's all sunshine and roses. There are different pain points when we're talking about adopting a design system. For our respondents, the biggest struggles were getting design and dev aligned and establishing clear processes – things that naturally go together. After that, documenting everything and proving ROI were the next biggest challenges.

Comparison chart

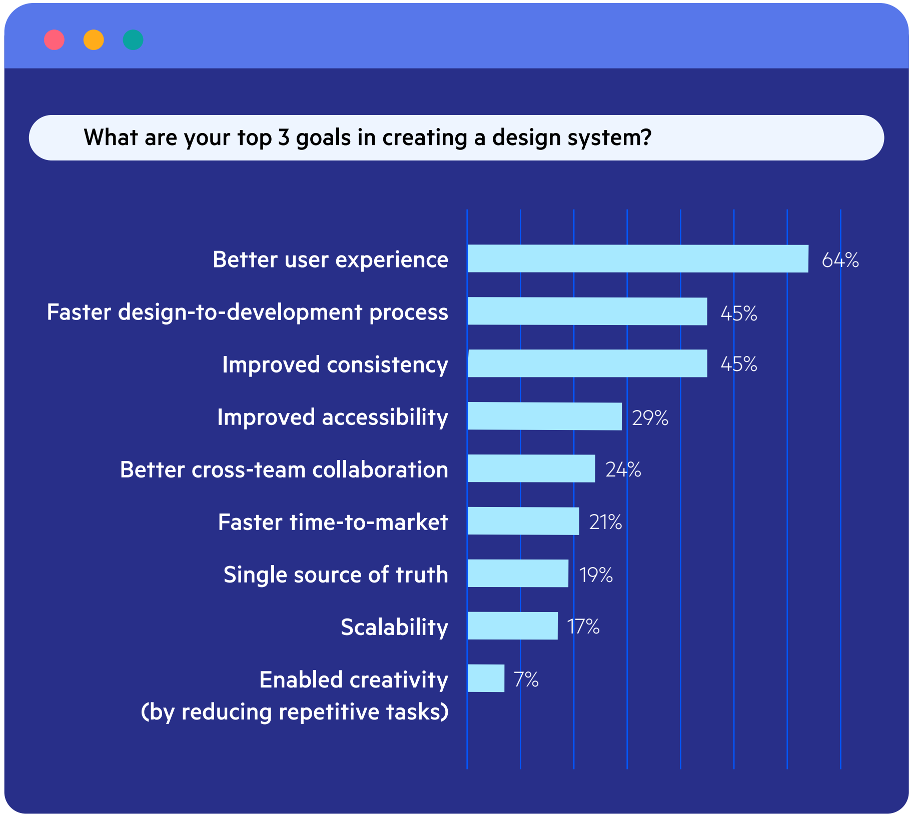
What are the main challenges of maintaining a design system? (this question only appeared to respondents who answered "Do you have a design system in place?" with "Yes, we have a design system in place")

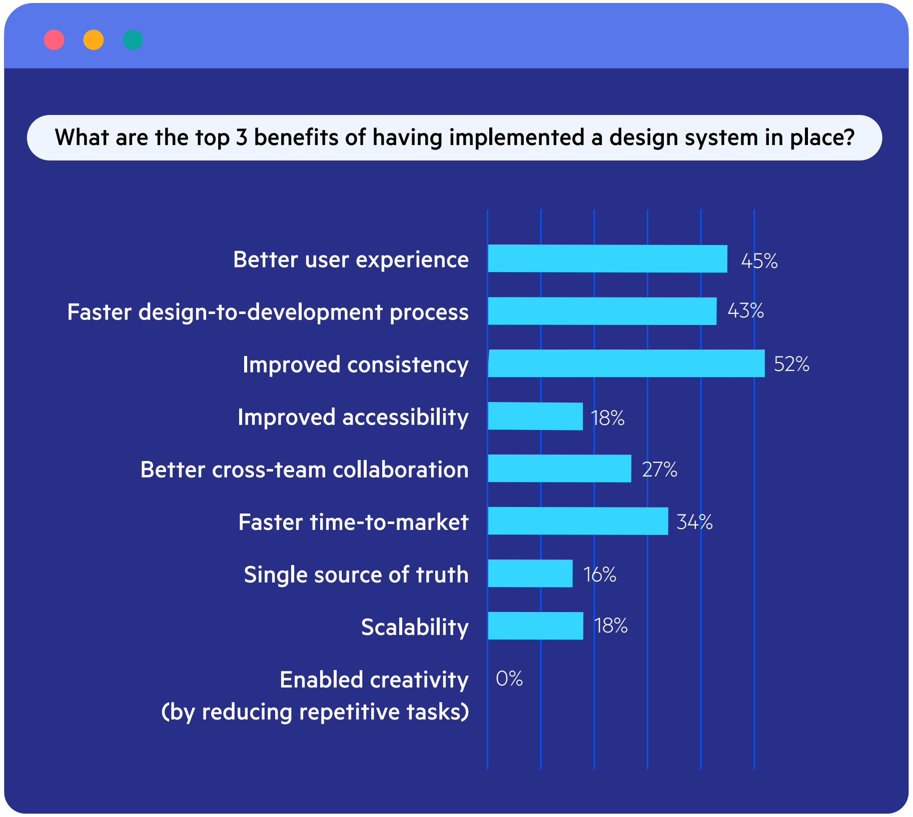
What are the main challenges in the process of creating a design system? (this question only appeared to respondents who answered "Do you have a design system in place?" with "We've just started creating our design system" or "Not yet but we're planning to create one in the near future")



Some other common challenges? Scope creep, allocation of resources and maintenance. No big surprises there—for better or worse, that's the stuff that comes with any large, long-term project.

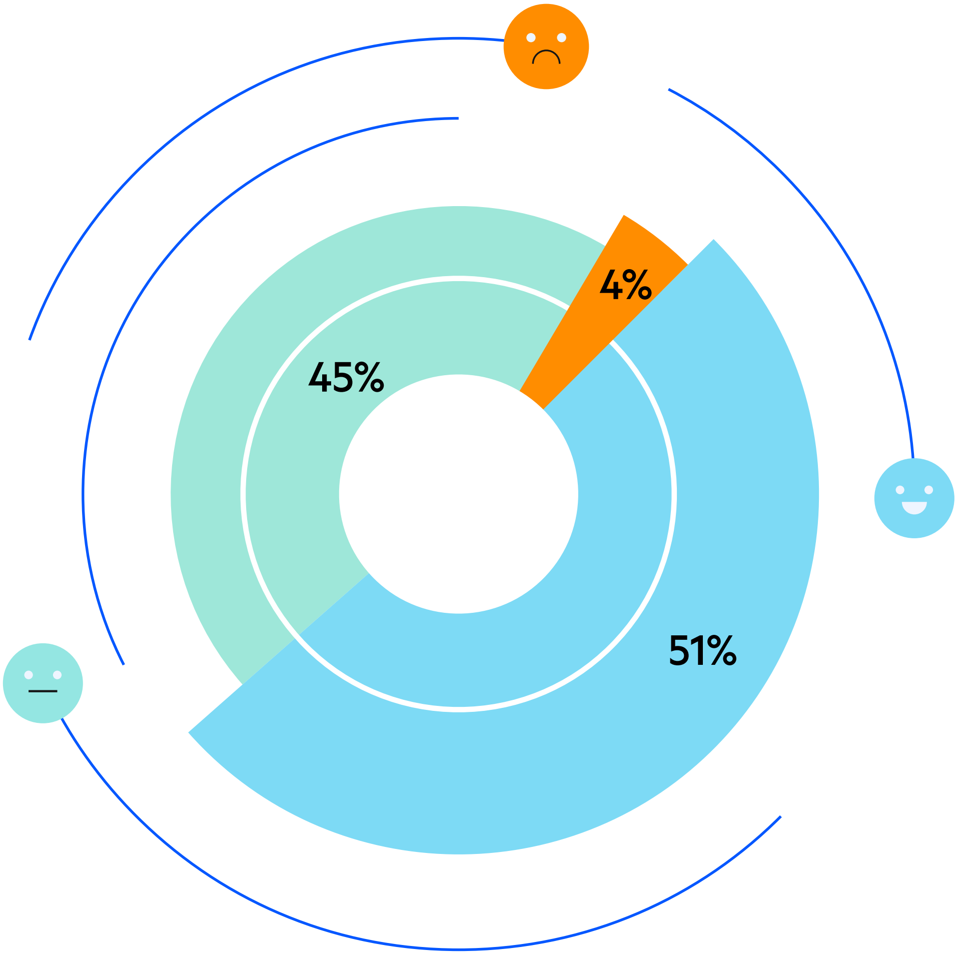
Now, it's not all bad—after all, if it was, nobody would bother. A design system requires significant upfront time and energy (and no small amount of maintenance afterwards), but it pays back those investments in spades.


What are your top 3 goals in creating a design system?


What are the top 3 benefits of having implemented a design system in place?

Even better, the results show that expectations are matched by results in many cases. For the folks that don't yet have a finished design system – better user experience, improved consistency and faster design to dev time are the top 3 reasons why they want one. And great news: those are also the top 3 benefits listed by folks who do have them! It's reassuring to see that expectations and results are aligned regarding the power and impact of a design system.

What is the general feeling in your team regarding the adoption/usage of design systems?



🟦 Design systems are great - they solve so many problems!

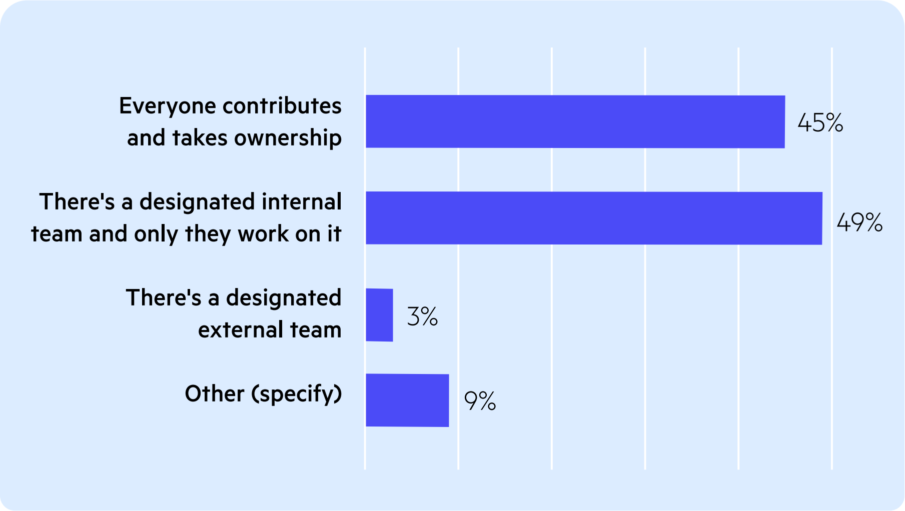
🟩 The adoption of the design system solved some problems but created others

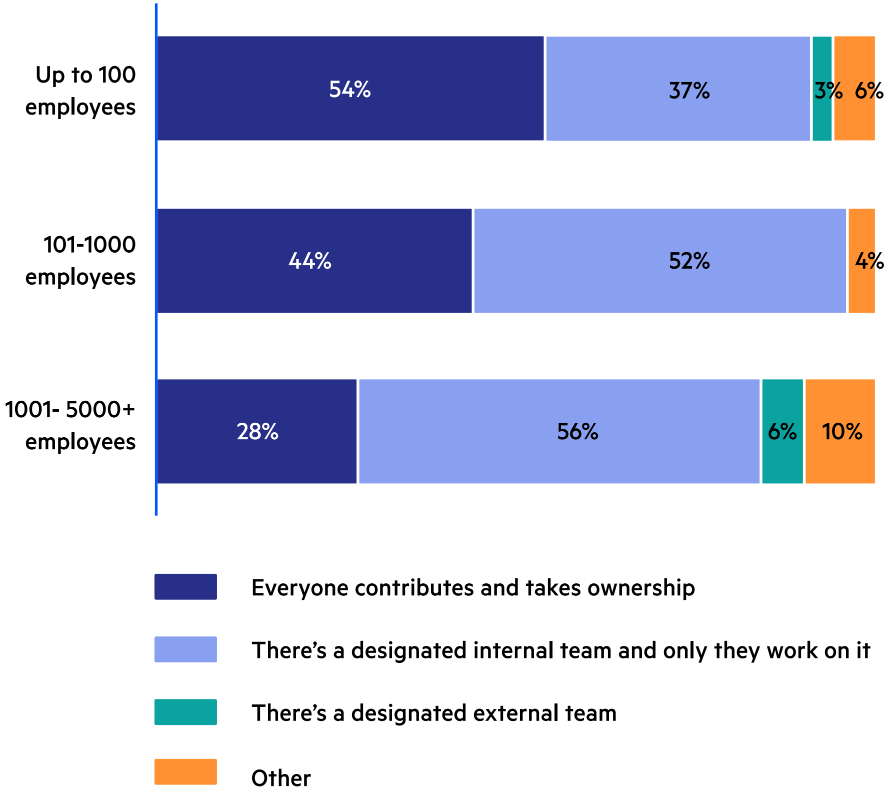
🟧 The adoption of the design system created too many issues

Overall, about half of users felt that design systems solved their problems, and the other half felt that it solved some problems but also created some new ones. Only 4% of users felt that it was a net negative, creating too many issues to be worthwhile. It's always reassuring to see that such a huge project isn't just wishful thinking; it does have real, tangible benefits.

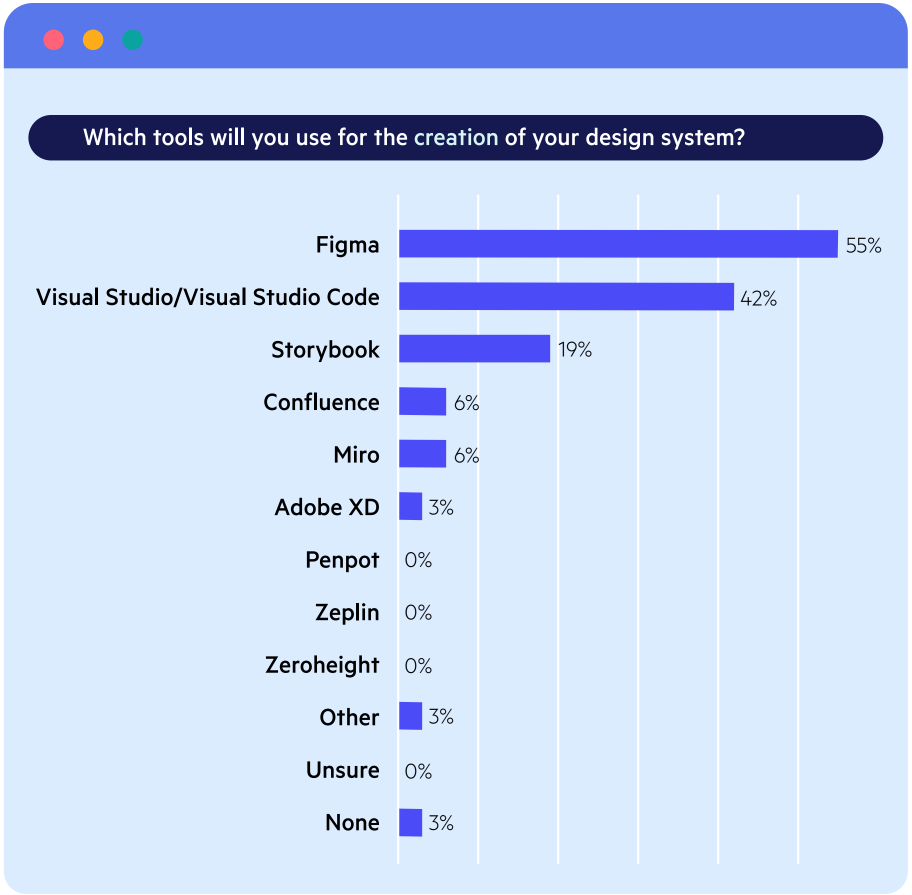
Design System Tools and Maintenance

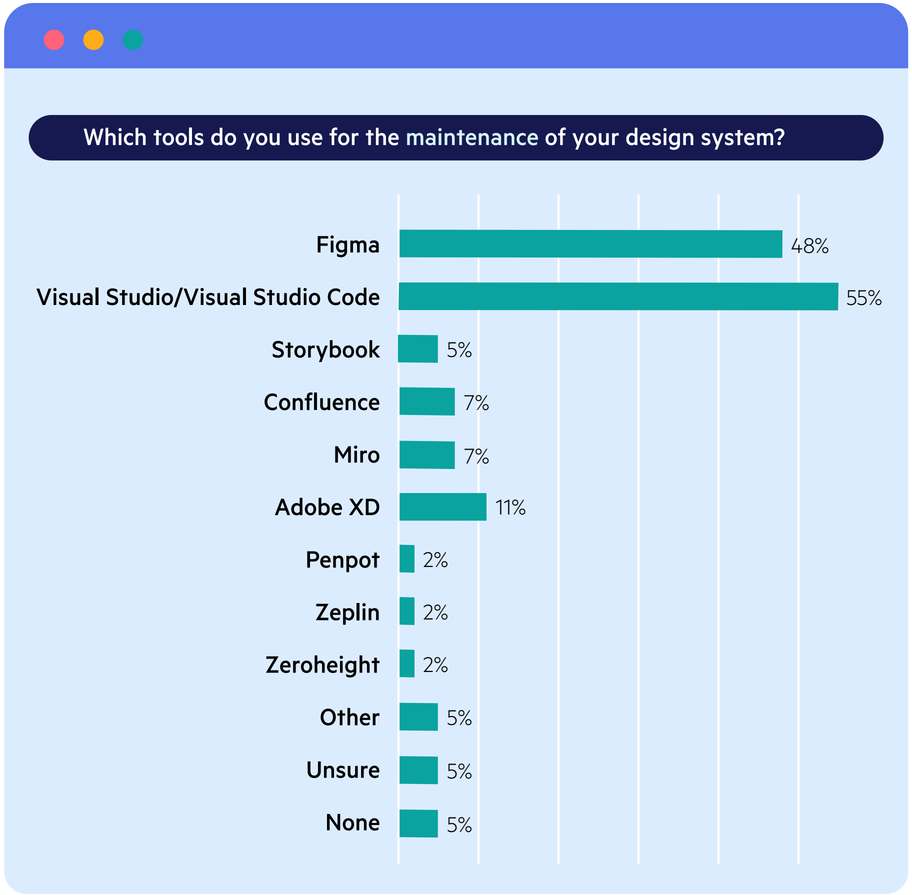
Who’s responsible for maintaining and/or contributing to the design system in your company?





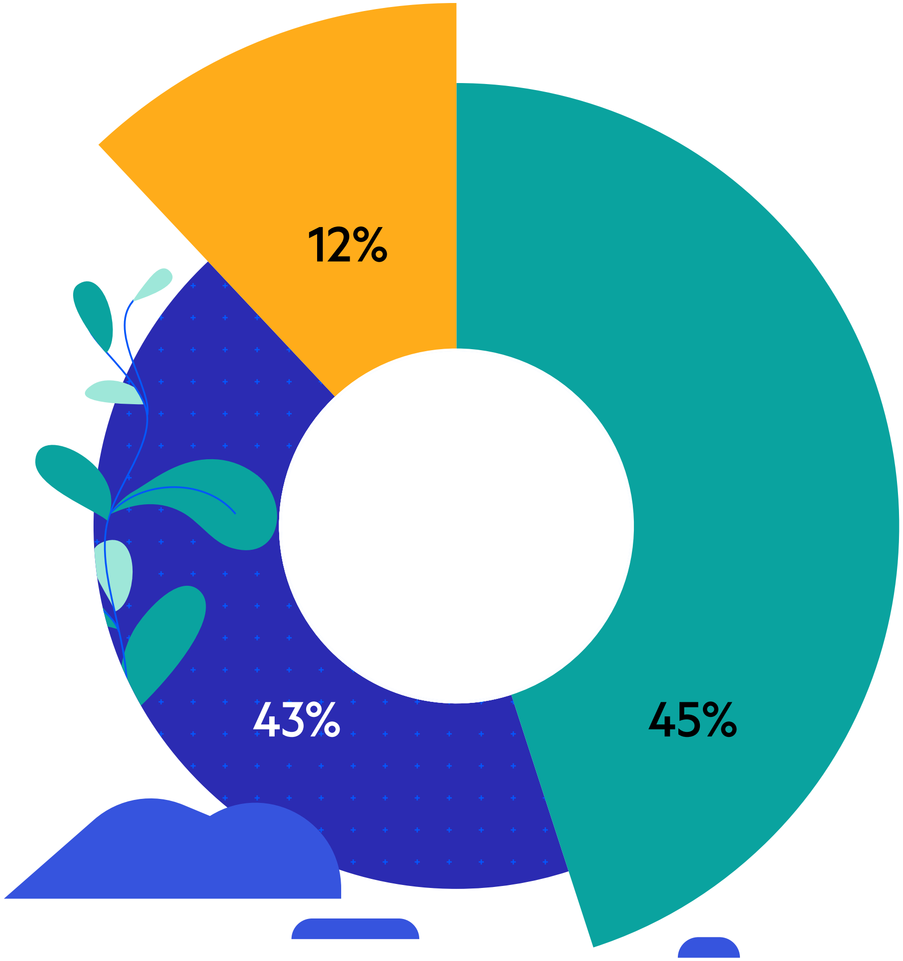
When it comes to maintenance and continued contributions to the design system, our survey found that there are two main approaches: everyone contributes equally or a dedicated design system team owns the system. Looking at the cross-section, we found that this reflected the company’s size; for smaller groups, a design system would need to be all-hands-on-deck, while a larger company often allows for more specialized roles. Only 28% of respondents report using an “everyone contributes” model in a 1000+ employee company, as compared to over half in a less-than-100 employee company.





When it comes to tooling, we see some familiar old favorites made an appearance again: Figma and Visual Studio/VS Code. Axure was a common write-in – we'll make sure to include it next year – but otherwise, no real surprises here.

Do you believe your team/company is using the tools that best enable designer-developer collaboration?



🟩 Yes, absolutely

🟦 Partially - there's room for improvement

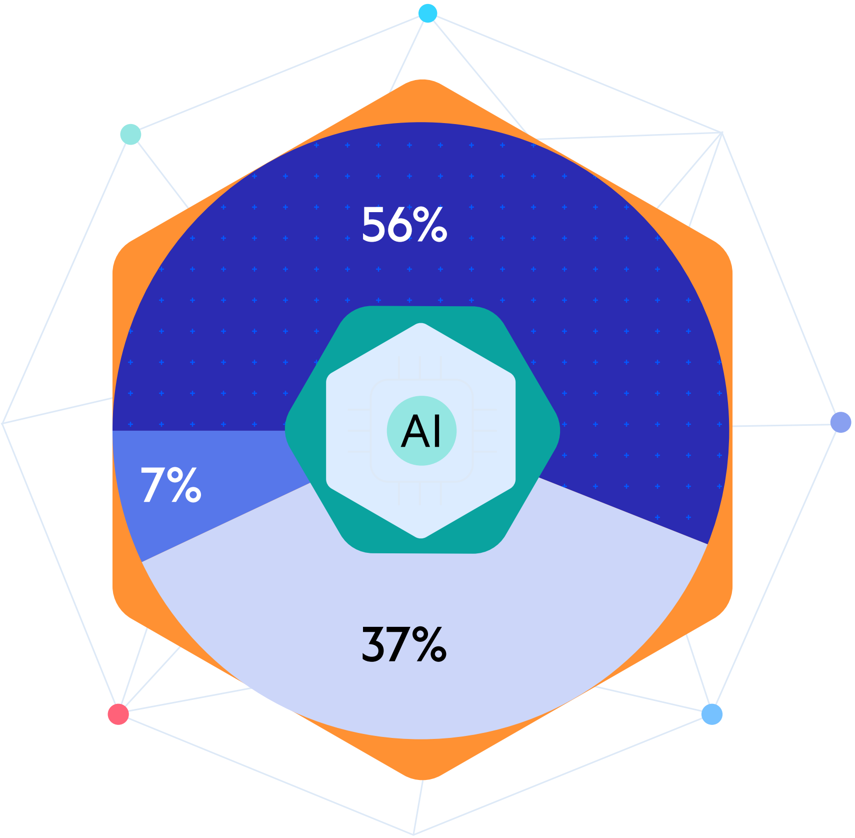
🟧 Not really

The other good news is that folks are, on balance, happy with those tools. While just over 40% feel that there's room for improvement, only 12% are outright unhappy with their options. With so many specific use cases when it comes to design systems, I'm not surprised that tooling isn't 100% meeting every need – but room for improvement means room for innovation. I'm excited to see what's coming for design systems in 2025.

The Future of Design Systems

What's a survey without a good predictions section? Let's see what folks are planning, excited for and thinking about with their design systems in the years ahead.

Are you experimenting with AI in your design system?

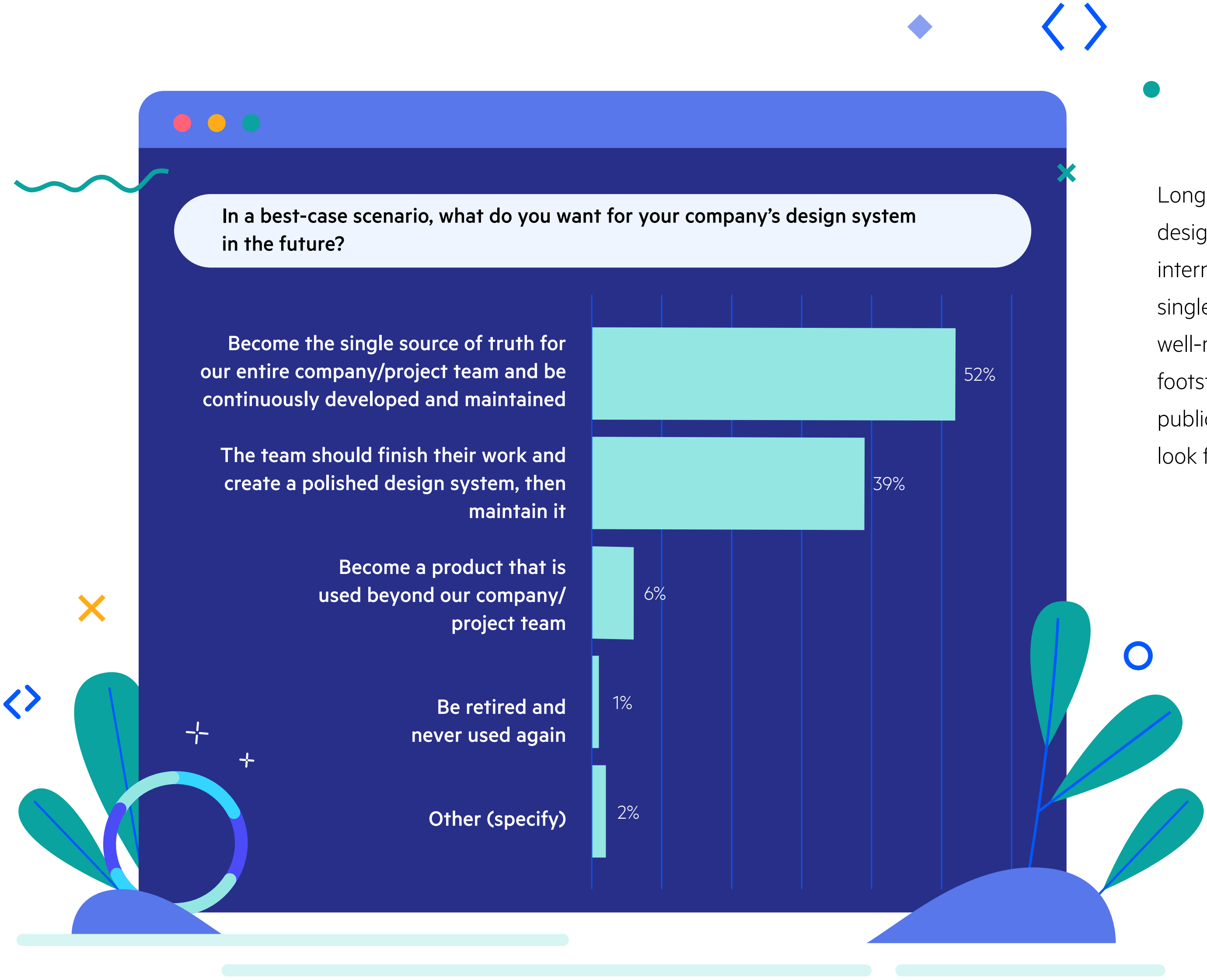


When it comes to AI in design systems, most folks are still feeling hesitant. Only 7% have implemented AI in their design system currently, and (from our open response follow-up question) they're primarily using GenAI to answer questions or create placeholder elements. 37% consider using AI to make suggestions, create templates or help brainstorm ideas. It will be interesting to see how – or if – any of those ideas end up playing out as AI develops. While there are some, like Brad Frost, who are feeling extremely positive about the incorporation of AI into design systems it seems that for most, it's not quite at the tipping point just yet.

■ **Yes**

■ **Not yet, but I'm considering it**

■ **No and I'm not considering it for the time being**



In a best-case scenario, what do you want for your company's design system in the future?

Long term, most respondents are hopeful that their design system becomes an integral aspect of their internal design-to-development process, serving as a single source of truth that's continually updated and well-maintained. Only 6% are hoping to follow in the footsteps of giants and launch their design system for public use. To them, we wish the very best of luck and look forward to seeing what they create!

Conclusion

When we look at these results compiled together, it's easy to feel optimistic about the future of designer/developer collaboration. Sure, there's work to be done—no doubt about it. But it's reassuring to see that none of the problems are insurmountable or unsolvable.

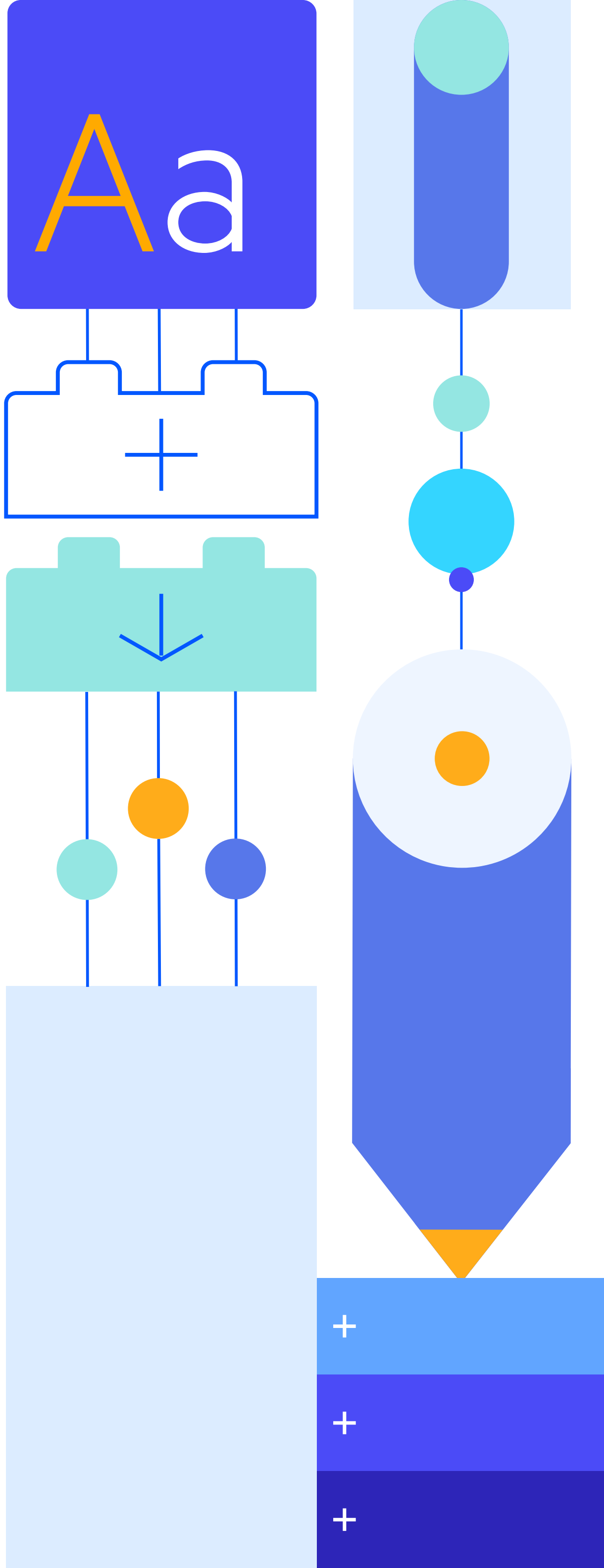
Many of the challenges hinge on communication: not speaking the same language, getting involved in the process too late, not checking in often enough, not clearly communicating expectations, not sharing a tool set. These aren't problems that will magically solve themselves. They require a commitment from both parties to work together towards bridging the gap. But as they say: "We have the technology. We have the capability…"

If you're looking for ways to help improve the design to development flow at your job, check out the Telerik and Kendo UI tools. Our Design System Kit, ThemeBuilder, Figma UI Design Kits, building blocks, page templates and 100+ beautiful, accessible UI components can help get everyone unified and working towards the same goal—faster and more smoothly than ever before.

And, of course, this isn't the end of the conversation. If there's something we left out, something you disagree with or just something that sparked an idea, we'd love to hear about it.

Keep working hard! We can't wait to hear about everything you've accomplished as designers and developers working together.

Until next year,
Kathryn & Nora



About Telerik and Kendo UI

UI component libraries that care about the design process

The Telerik and Kendo UI component libraries and UI customization tools give developers and designers a common language. This toolkit removes many of the obstacles presented by the design-to-code process and significantly reduces tedious CSS, guesswork, and frustrating back and forth—and, ultimately, time to market.

→ I want better designer-developer collaboration

About Progress

Progress (Nasdaq: PRGS) empowers organizations to achieve transformational success in the face of disruptive change. Our software enables our customers to develop, deploy and manage responsible AI-powered applications and experiences with agility and ease. Customers get a trusted provider in Progress, with the products, expertise and vision they need to succeed. Over 4 million developers and technologists at hundreds of thousands of enterprises depend on Progress. Learn more at www.progress.com

Worldwide Headquarters

Progress Software Corporation
15 Wayside Rd, Suite 400, Burlington, MA 01803, USA
Tel: +1-800-477-6473

- facebook.com/progresssw
- twitter.com/progresssw
- youtube.com/progresssw
- linkedin.com/company/progress-software
- progress_sw_

Progress®