# The State & Future of **Frontend Development** in the **Balkan Region**

kutia

# Note to our readers

Frontend is always changing. New libraries, new frameworks, new languages... It's all part of the fun, but it can also feel overwhelming sometimes.

That's where The Now & Future of Frontend Development in the Balkan Region report comes in: this year, we surveyed over 500 frontend developers from Kosovo, Albania, Macedonia, Montenegro, Bosnia and Herzegovina, Serbia, Bulgaria, and Greece to figure out what they're using, what they're happy with, and what they want to learn. The results are a unique collection of stats and insights that will hopefully help us make our own way through the Frontend development.

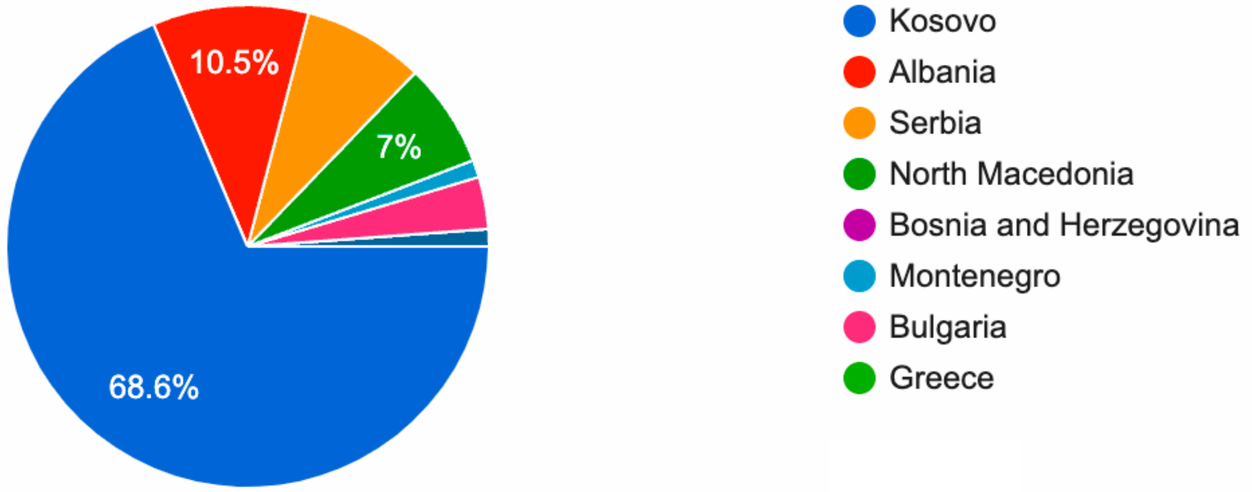
We pooled our resources and worked tirelessly to create an insightful report that would direct us to paving the road for new opportunities, market demands, and for opening new doors for our future generations. We were driven by the need to present new, up-to-date information on trends in Frontend Development and were inspired by several studies of this kind.

Could the figures be higher? Of course, we say. However, every new initiative yields fresh outcomes. As a result, for roughly ten months straight, our survey was distributed to all of the neighboring developers, opening the door for another project. Building new bridges of cooperation across Balkan countries is what the Coders Friendship Gathering is doing.

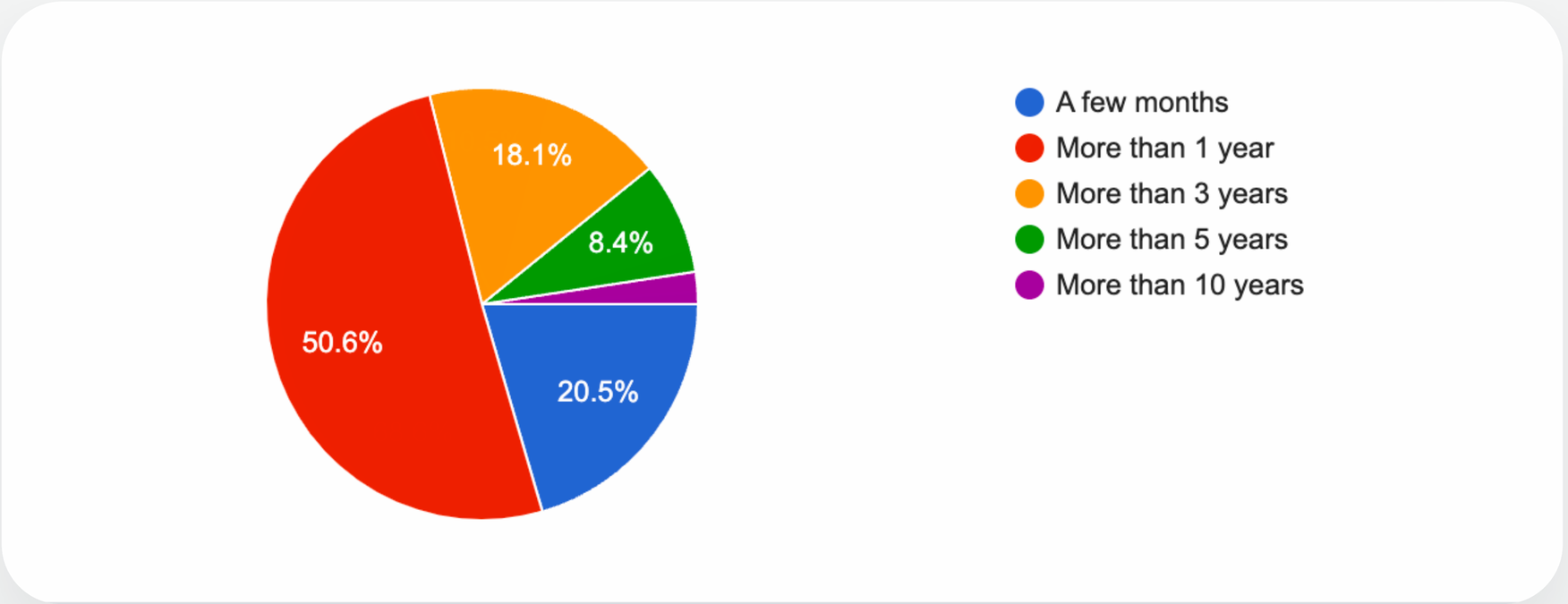
And now, let's see what Frontend has been up to!

**With love,**
Kutia, Prishtinë 2022

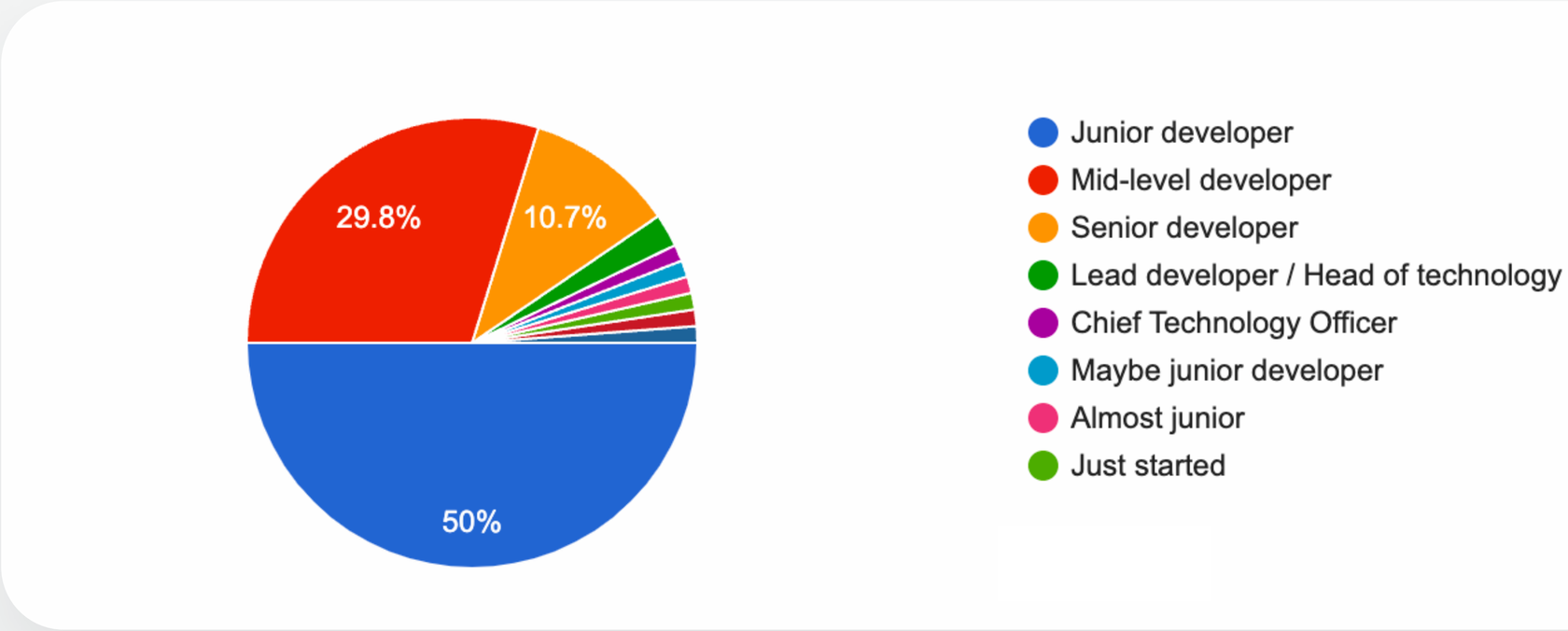
## 1. Which country are you from?



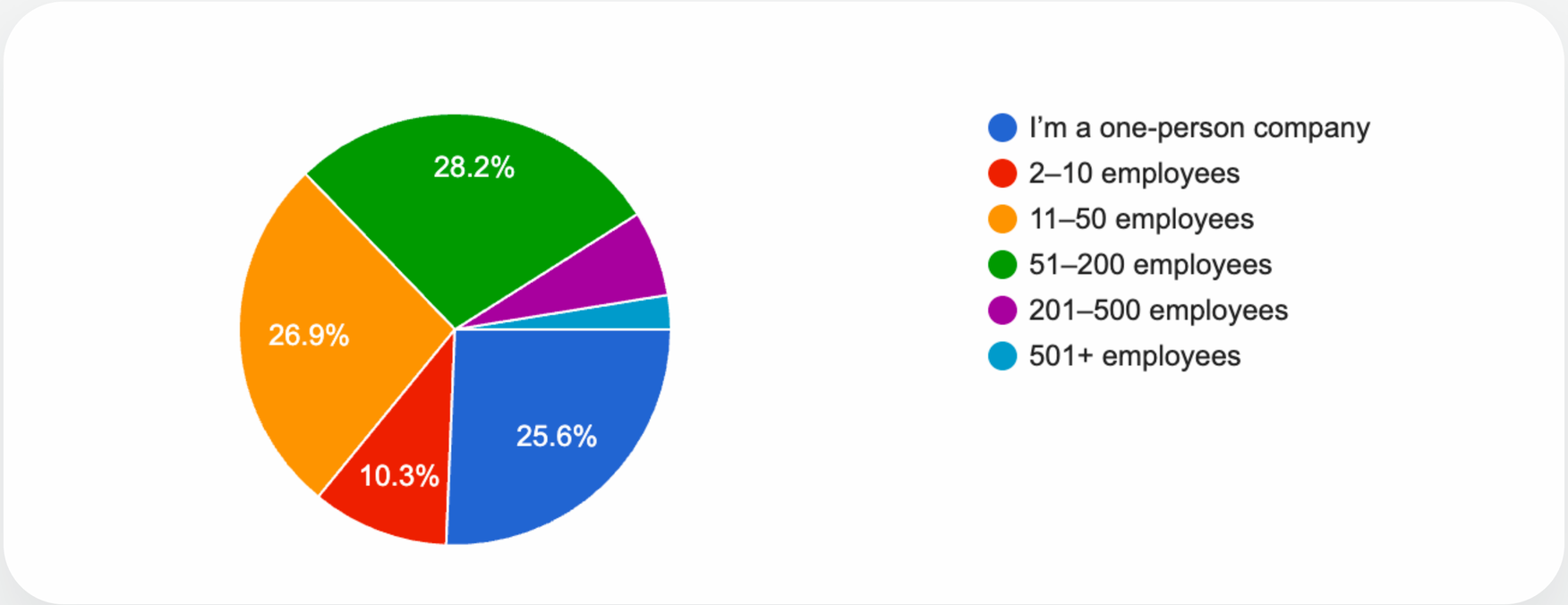
## 2. For how long have you been in the frontend development game?



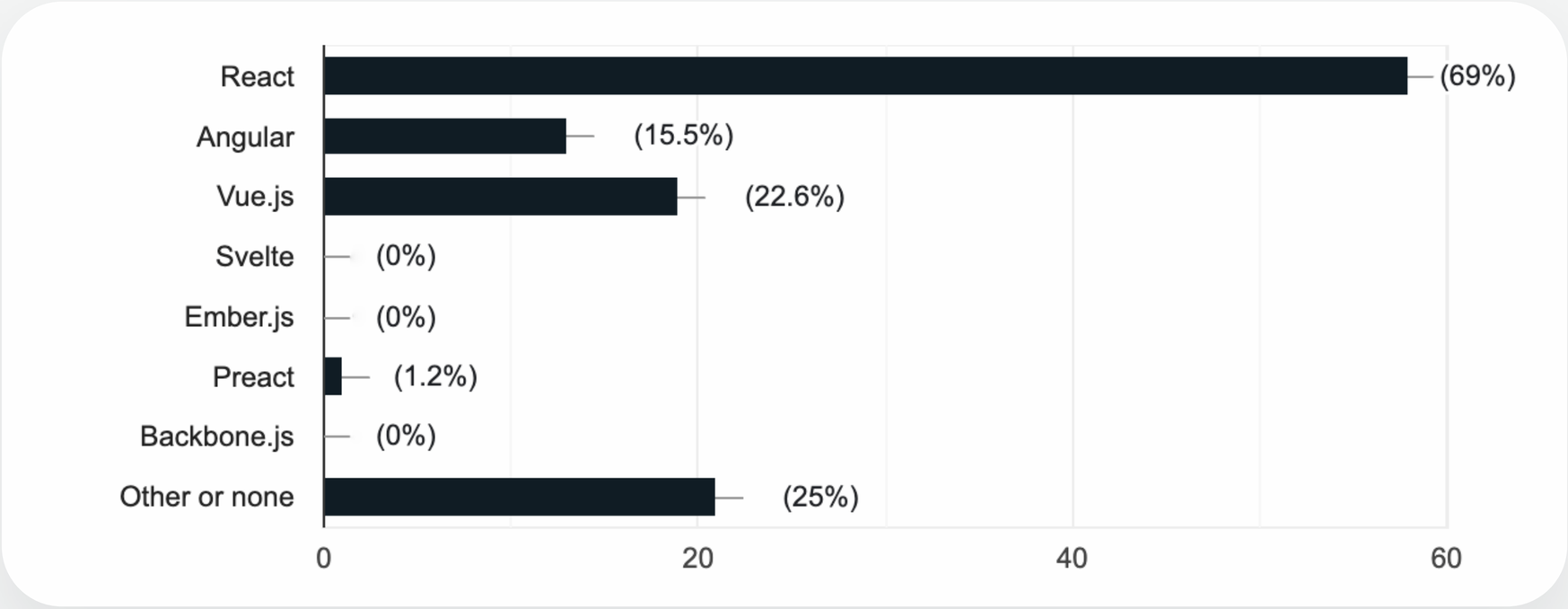
## 3. How would you describe your seniority?



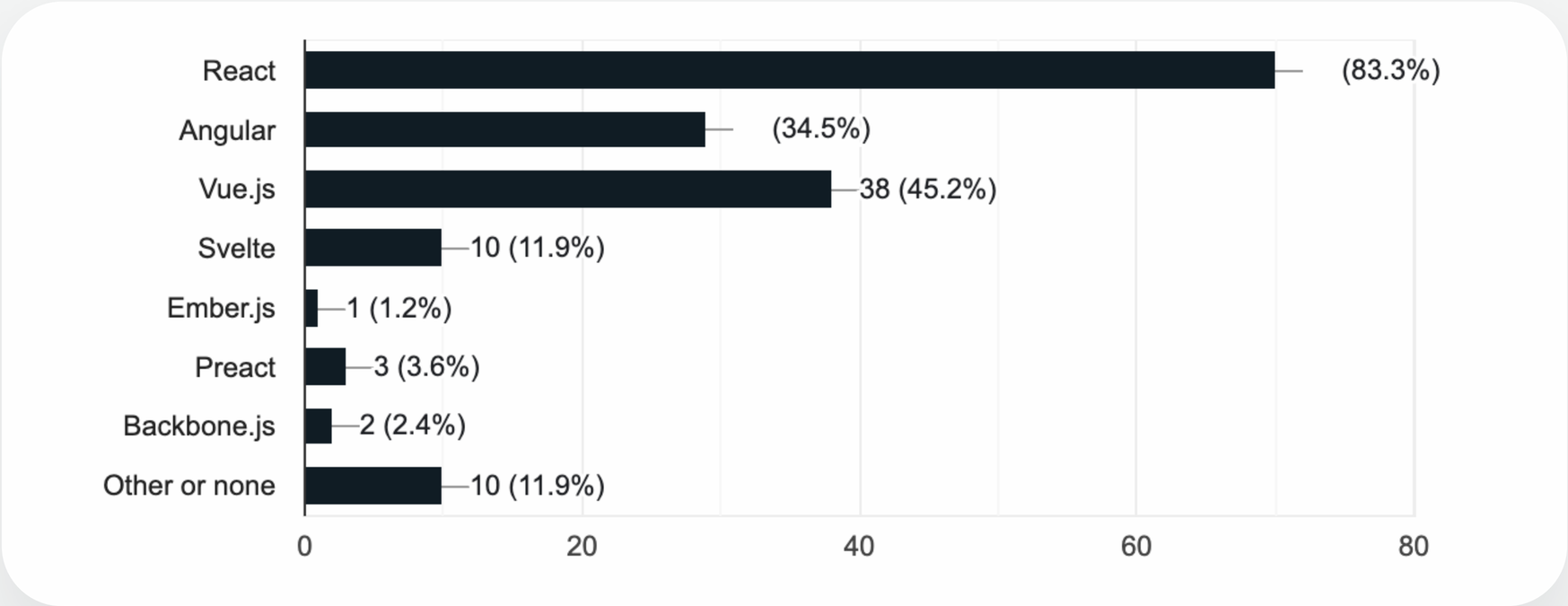
## 4. How big is the company you are working in?



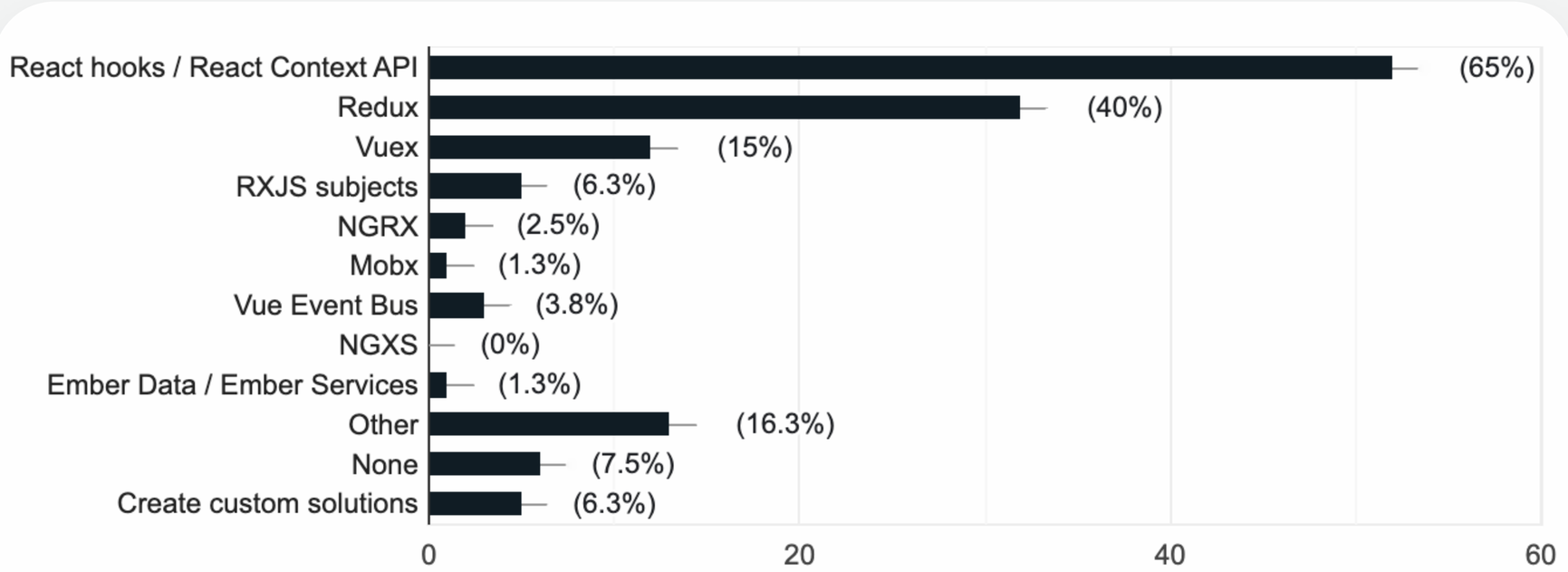
## 5. Which of these frameworks have you used during the last year?



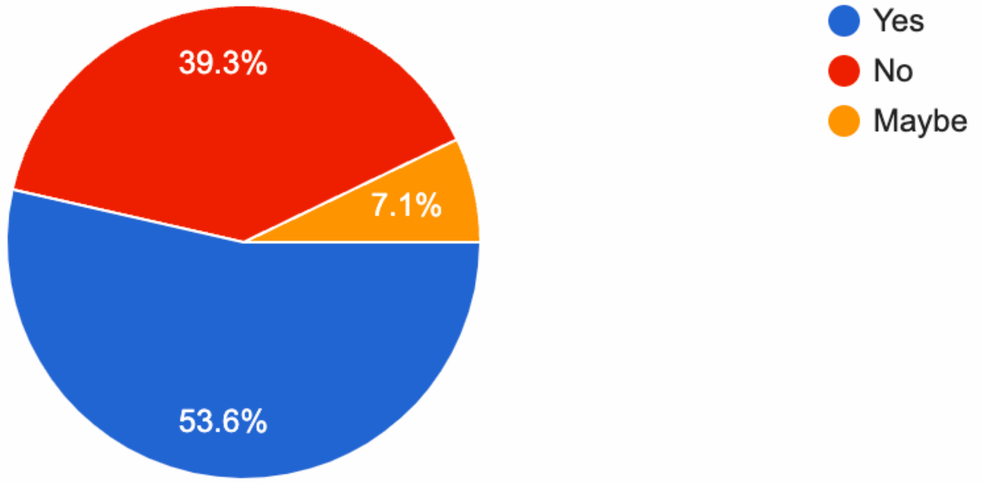
## 6. Which of these frameworks would you like to keep on using or want to learn in the future?



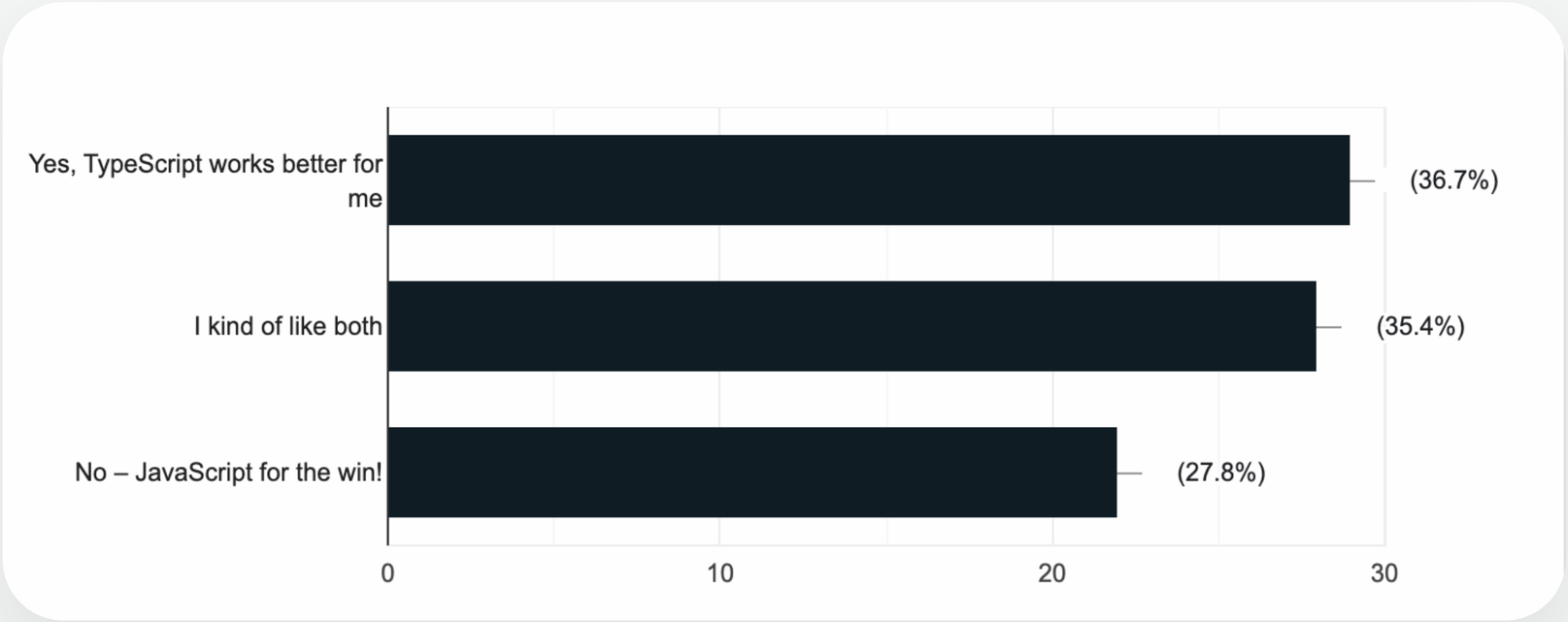
## 7. Which solutions do you use when it comes to state management?



## 8. Have you used TypeScript during the last year?



## 9. Do you like TypeScript better than JavaScript?



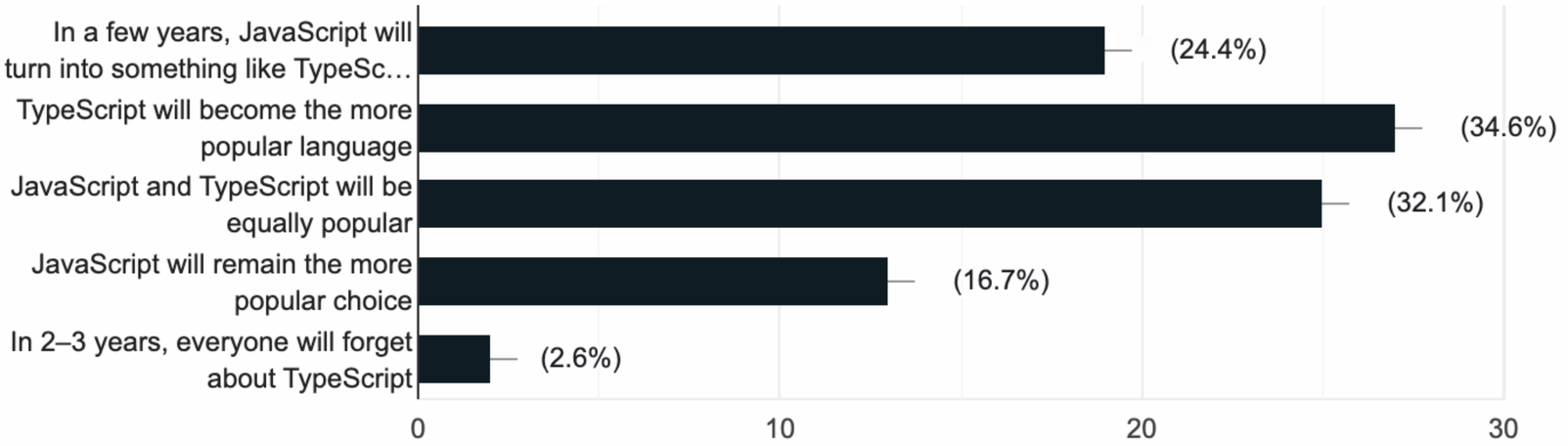

Comparing the advantages TypeScript has over JavaScript is good to see this result that TypeScript is more liked by the Developers, we should always try to adapt and use the best practices and the languages that support our project best.

The popularity of TypeScript, community support, adding some advantages like: Support OOP, optional strong static typing while JS is a dynamically typed language which means that types are checked, and data type errors are only detected at runtime.

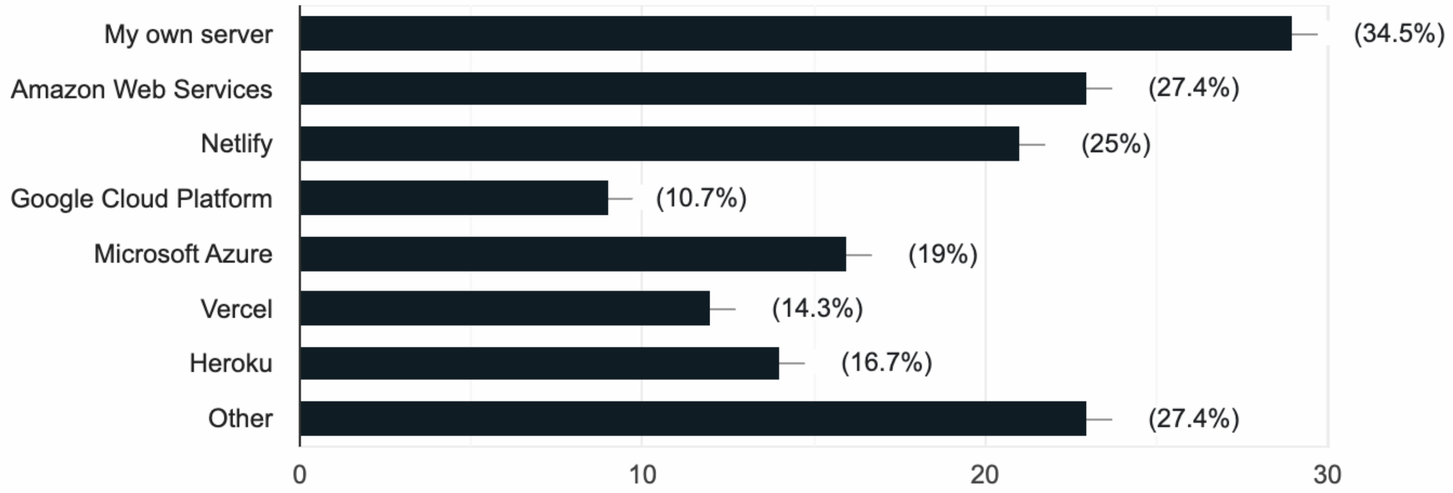
Section commented by:

**Alban Sahiti**
CEO @Tenton

## 10. What do you think about the future of TypeScript?



## 11. Where do you usually deploy your applications to?




In today's software world, many factors impact the decision to choose a hosting infrastructure. While many enterprises have embraced the benefit of the cloud, others are unable to make the leap, instead relying on traditional on-premises infrastructures to do business.

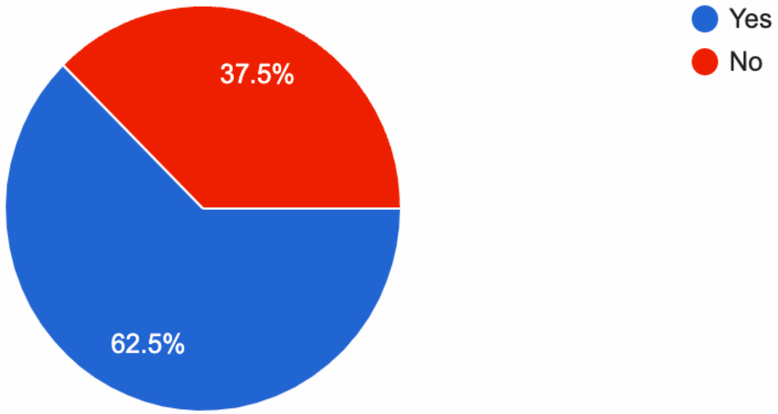
When talking about the developers' choice, the development stack determines where and how to run a platform. With the proliferation of containerization technologies, more and more developers are choosing docker, and similar runtimes, to build and run their software.

Running containerized applications in a bare metal server is a wise choice. It lets you get many more individual applications running on the same number of servers while having complete control over hardware management and utilization. It also reduces costs and increases speed and agility. Unsurprisingly, more developers are choosing to host their platform in their dedicated hardware versus the cloud, which in many cases, might provide better scalability, reliability, and security.

Section commented by:

**Hekuran Doli**
Founder & DevSecOps Engineer

## 12. Do you use Continuous Integration?




Software delivery is a complex and time-consuming process. A process that involves days of integration, configuration, and testing while the risk of discovering an error threatens to force the process back to the beginning of the lifecycle.

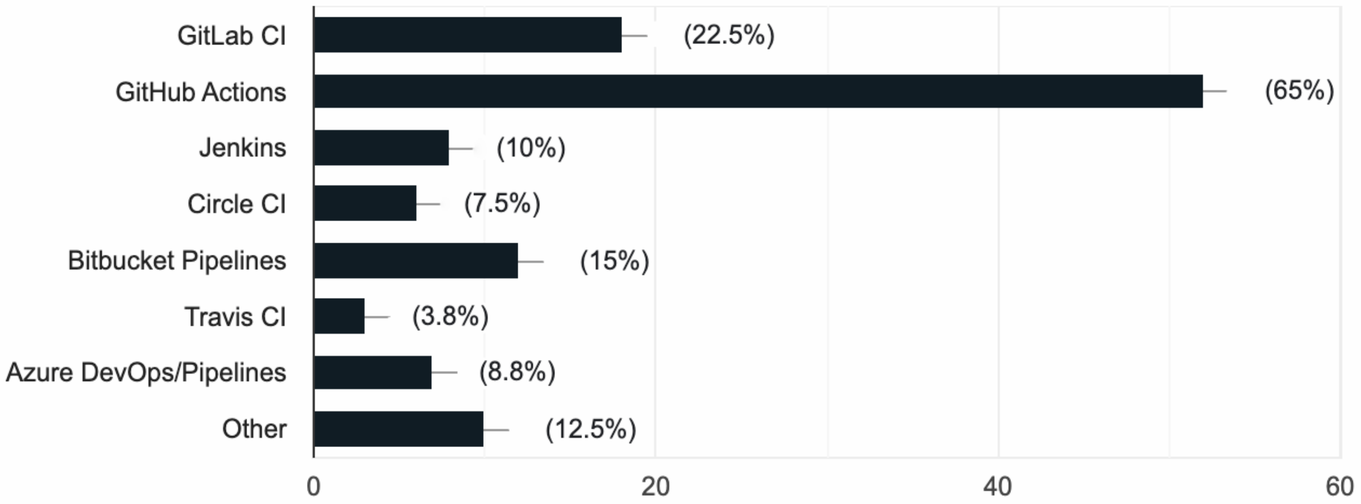
The time commitment involved in making changes means that new releases are delivered less frequently and have a greater chance of being prone to error. Continuous delivery addresses this problem efficiently, enabling developers to release more frequently without compromising the quality of their work.

The CI is so beneficial that once the developers are introduced to the concept, they will love and embrace it. I expect the rage to increase dramatically in the years to come.

Section commented by:

**Hekuran Doli**
Founder & DevSecOps Engineer

## 13. Which CI solutions have you used during the last year?




Over the past decade, DevOps have become a crucial part of the software life cycle. This fueled the birth of many tools and practices to support the CI/CD process, where Jenkins and GitHub Actions outstandingly stand among them.

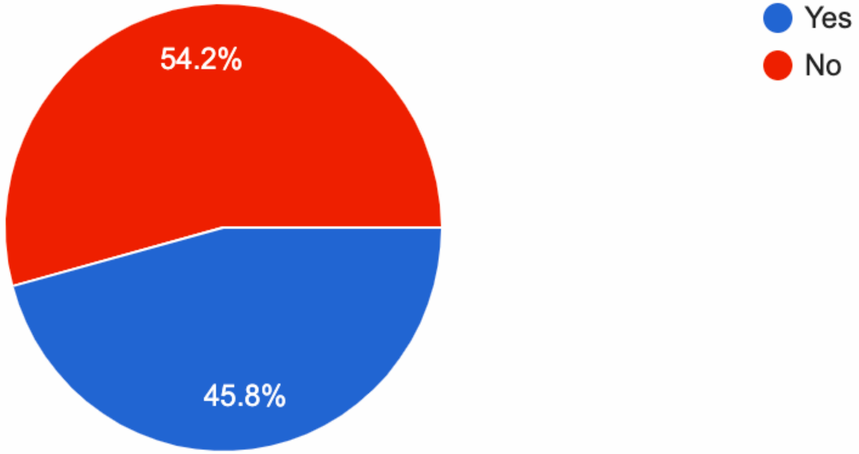
Jenkins, like many other tools, makes it easier to automate the process of building, testing, and deploying software projects. Setting up and maintaining Jenkins requires a level of expertise. GitHub Actions, on the other hand, is a fully managed service by GitHub.

You don't need to know how to scale and operate the infra-structure to run it, and for those who use GitHub as the source control platform and already feel confident with it, GitHub actions is the preferred choice.

Section commented by:

**Hekuran Doli**
Founder & DevSecOps Engineer

## 14. Do you make use of containerization?




Virtualization has emerged as a way to minimize idle computing resources by distributing hardware resources across multiple containers.

Containers are a streamlined way to build, test, and deploy applications in various environments, from local computers to testing environments and even the cloud.

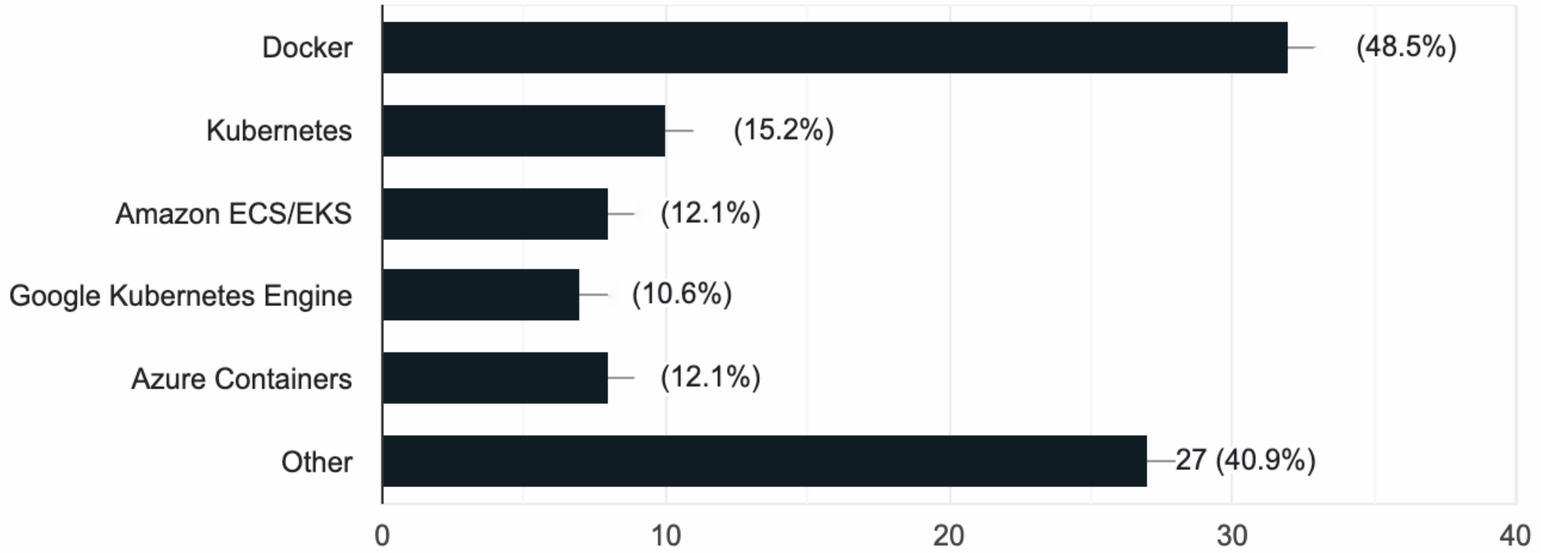
It gives the developers the confidence that applications in containers will run the same, regardless of where they are deployed.

Containerization has made traditional deployment obsolete. Surprisingly, the majority of participants are not taking advantage of such a benefit in the application deployment process. I expect the rates to grow as the developers are introduced to the containerization trend.

Section commented by:

**Hekuran Doli**
Founder & DevSecOps Engineer

## 15. Which container management solutions have you used during the last year?



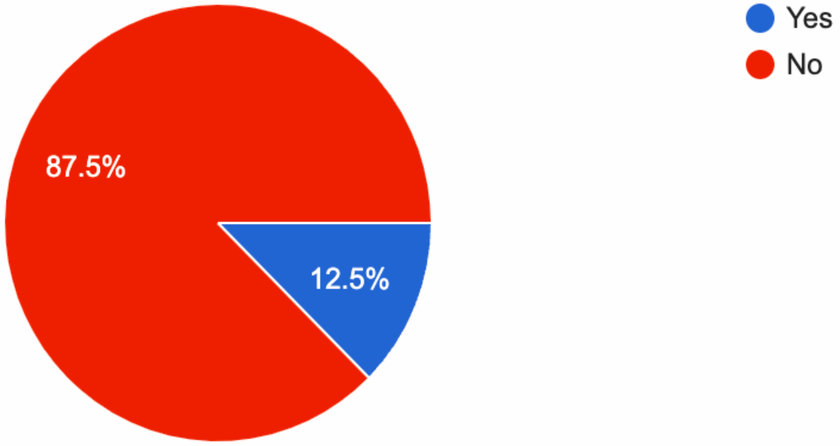

Kubernetes is a suite of services that automate many of the DevOps processes, which were previously handled manually, and simplify the work of software developers. It provides container management, simplifies the scaling process, and provides load balancing. Kubernetes is well suited for complex applications distributed across multiple servers. Because it enables automated scaling, healing, and monitoring, it has a significant advantage over its competitors. Therefore, Kubernetes has quickly become a standard way of deploying complex applications.

On the other hand, Docker is designed for ease of use, making it a preferable choice for running simple applications. Docker is not the best option for deploying applications that require complex automated scalability and High Availability. That's why I expect an increase in favor of Kubernetes as the applications start getting more complex.

Section commented by:

**Hekuran Doli**
Founder & DevSecOps Engineer

## 16. Have you built JAMstack websites?




Despite the innovation JAMstack is bringing, more than 80% of the developers have not yet put this architecture into use. Statistics of JAMstack usage in our region indicate that this technology has been introduced to the market quite recently.

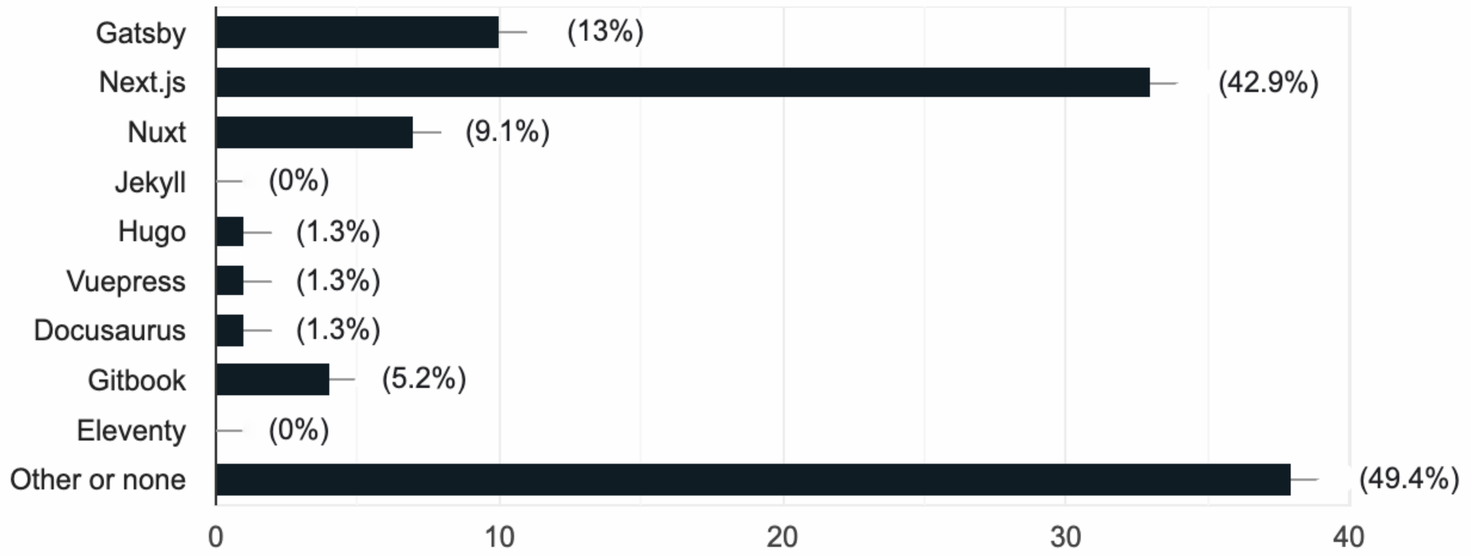
However, looking at the trends, the current results do not exclude the possibility that the JAMstack's popularity will quickly change.

From our experience at Sogody, we have managed to tailor this architecture design for a category of projects which resulted in better performance and faster load time, we avoided infrastructure setbacks and large server resources required on a daily basis.

Section commented by:

**Lorik Mullaademi & Arbena Musa**
CEO and Web Developer @Sogody

## 17. Which static site generators have you used during the last year?




Next.js and Gatsby together, cover more than 50% of the developers' responses. Their use is distinguished compared to other SSGs because they have prioritized the developer experience at a level that makes development much easier and attractive.

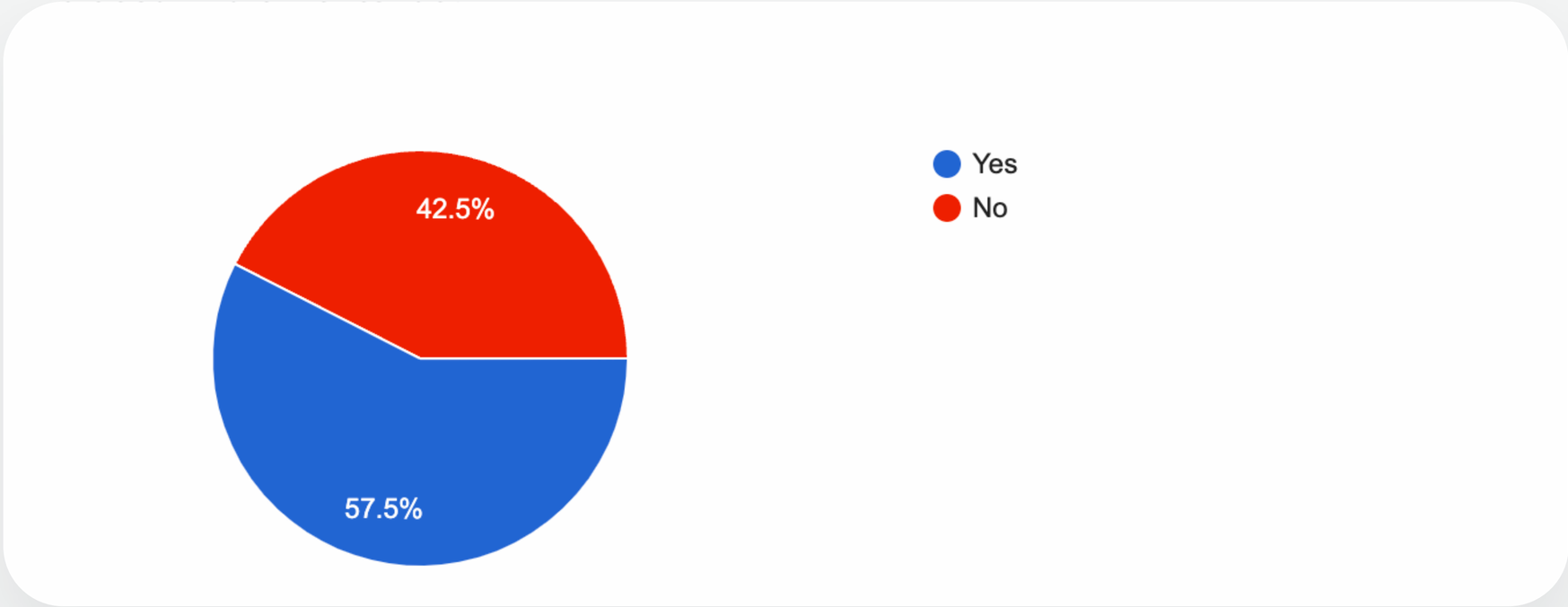
Recently, we at Sogody are mostly focusing on Next.js and Gatsby to generate static sites and content-enriched markup.

These solutions, along with many headless stacks, are focused on built-in performance, integration and optimization, which make e-commerce technology shifts and transitions easier due to support for integration with external services via plugins and third-party platforms.

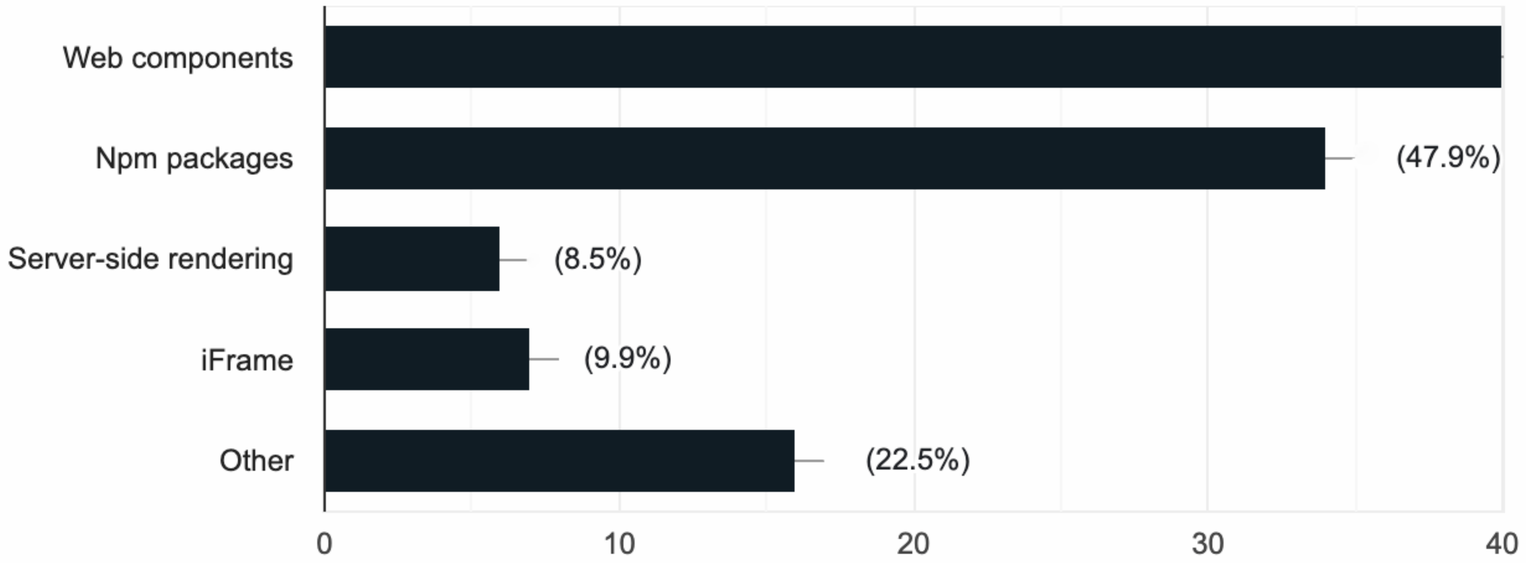
Section commented by:

**Lorik Mullaademi & Arbena Musa**
CEO and Web Developer @Sogody

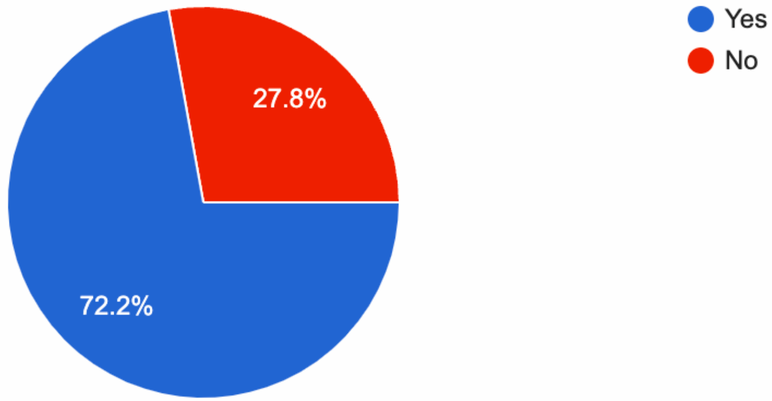
## 18. Have you used micro frontends?



## 19. How do you compose your micro frontends?



## 20. Do you take care of Search Engine Optimization?



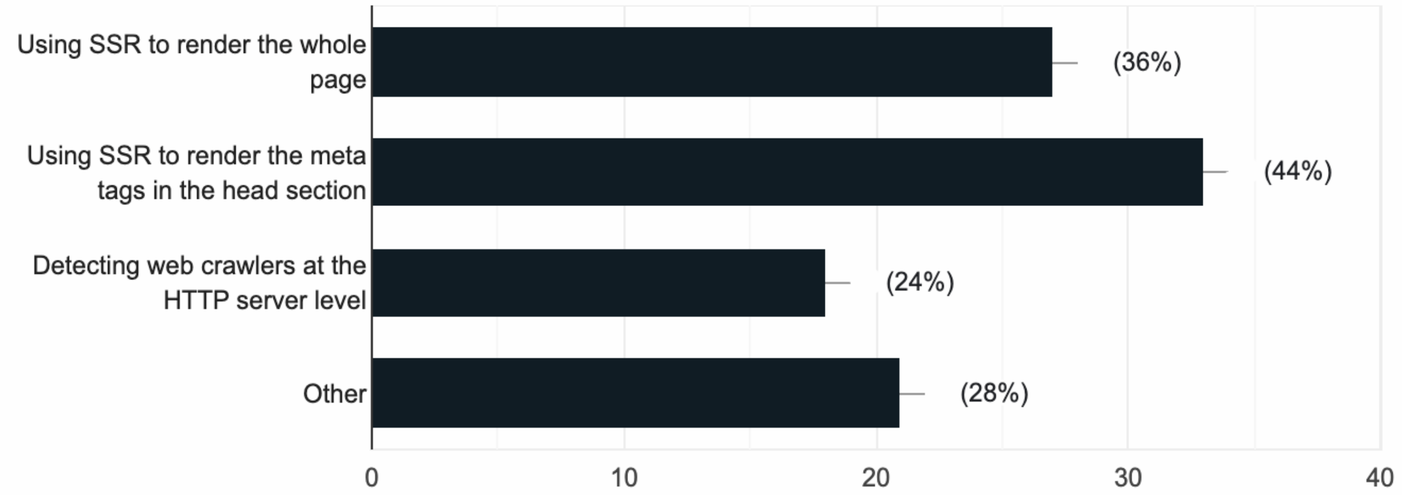

Seeing that close to 30% of front-end developers have responded 'No' is concerning. The idea is to code something that will bring value to the businesses that hire you. Having visibility in search results is very valuable/important to every consumer/B2B facing company out there.

SEO, at its core, is about following dev best practices. If you've answered 'No' to taking care of SEO, it shows that you're not good at your job, and this is something that must be addressed.

Section commented by:

**Granit Doshlaku**
Co-founder & SEO Director @Manaferra

## 21. How do you approach the subject of SEO?




Google is able to crawl and render JS pages, but they won't do that immediately. Once they see there's JS on a page, they'll take a note and come back later (days or weeks) to render it, and this will have a negative impact on that page's performance (and the whole site if there are lots of pages that need rendering).

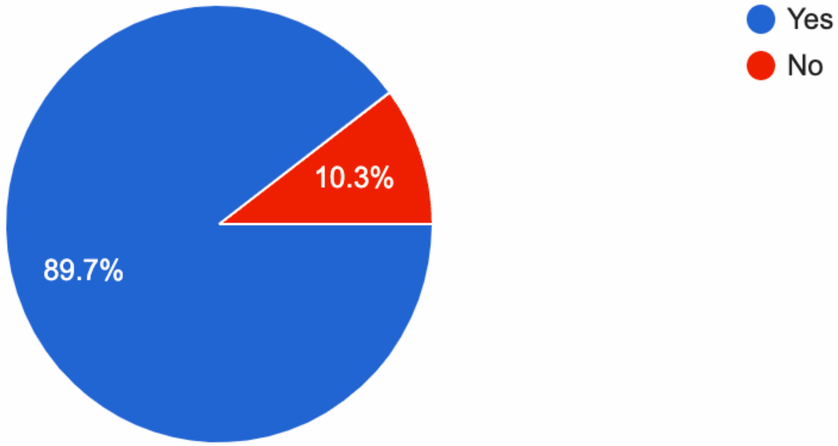
44% of the participants are doing it wrong. There's very little value in using SSR to render just meta tags when the most important part of the page, the main content, isn't rendered. Using SSR to render the whole page works great, but at scale could become quite pricey.

Dynamic rendering is the way to go, SSR for Google's crawler (Googlebot), and CSR for other bots and visitors. This way, the site does well in search results, and the client doesn't spend a fortune on rendering bills.

Section commented by:

**Granit Doshlaku**
Co-founder & SEO Director @Manaferra

## 22. Do you take care of application accessibility?

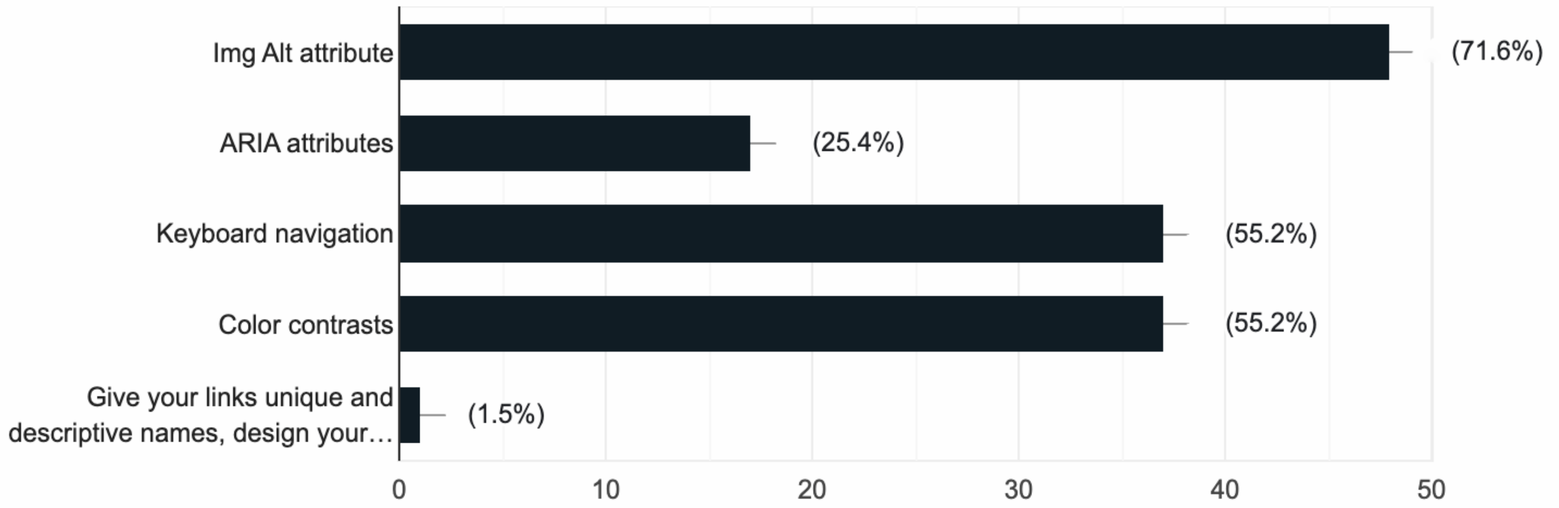



The % of YES is a bit higher compared to reality right now, knowing that most of the apps are built not so accessible, but this should and is changing daily and more apps are becoming more accessible to people which means more usage and more uniqueness.

Section commented by:

**Milot Hajredini**
DevOps @BBros

## 23. How do you take care of application accessibility?

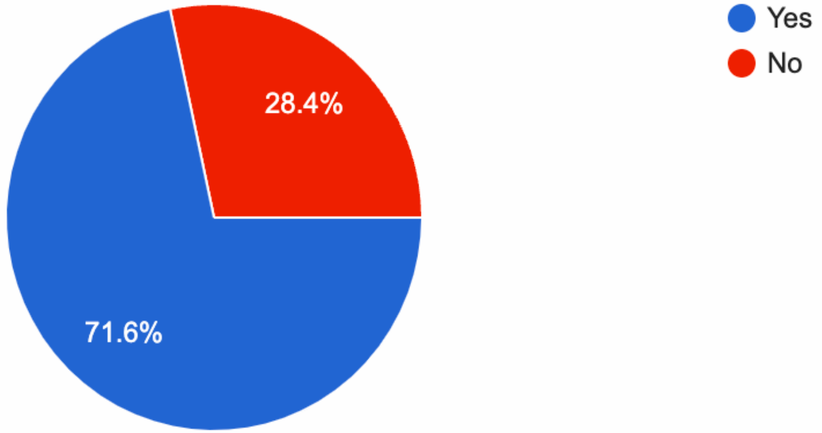



Developers should be more aware of using tags to make the apps accessible, because as we can see most apps that differentiate from another being not accessible is by using the right tags, and thus gain more audience. To make one app completely accessible, it means that the developer should take care of the details and add or use the right tags at the right element.

Section commented by:

**Milot Hajredini**
DevOps @BBros

## 24. Have you worked as part of a development team during the last year?

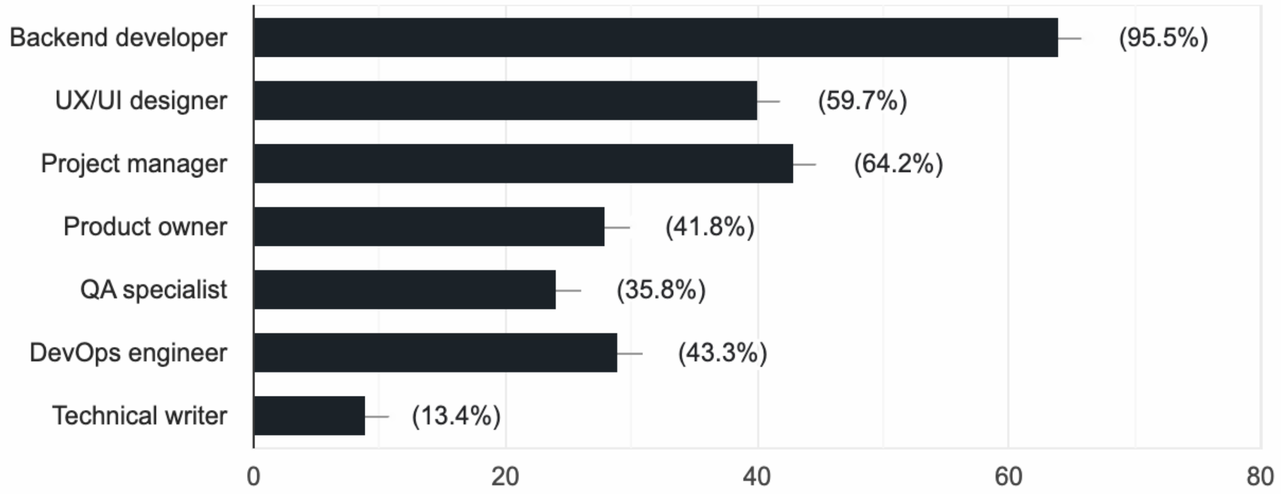



A graph like that gives an indication of a market contraction that has been reflected in 2022, showcasing bench vs people in projects. In 2020/2021 pandemic times, the situation with the bench was quite different as there were less than 5% of people in companies that were not part of development teams as everyone was running to go digital.

Section commented by:

**Çelik Nimani**
CEO @Frakton part of KIN + CARTA

## 25. Which of these people were part of your project development team(s)?



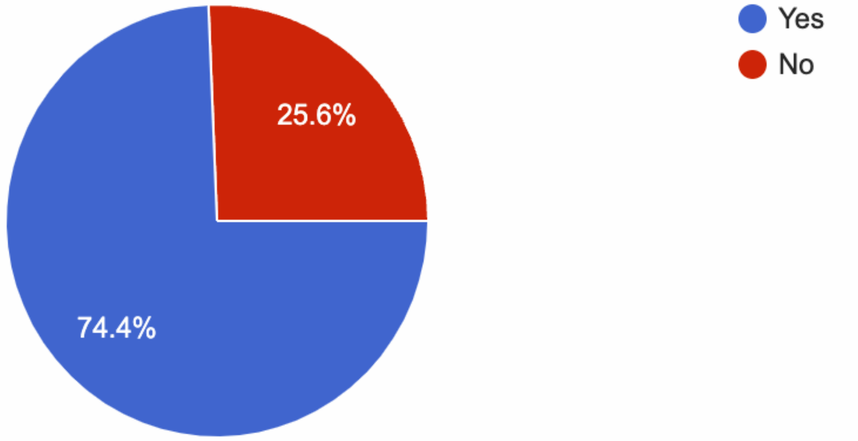

Based on the responses, one can see that the trend for Backend Engineering is growing and that there is more demand for Engineering Level on the Backend Requirements rather than Front-End.

This could also be a reflection that more companies and teams are getting involved with complex projects, as well as getting their hands in digitalizing solutions that used to be manual or old-software moving to new and modern ones.

Section commented by:

**Çelik Nimani**
CEO @Frakton part of KIN + CARTA

## 26. Have you used any handoff tools when working with designers during the last year?

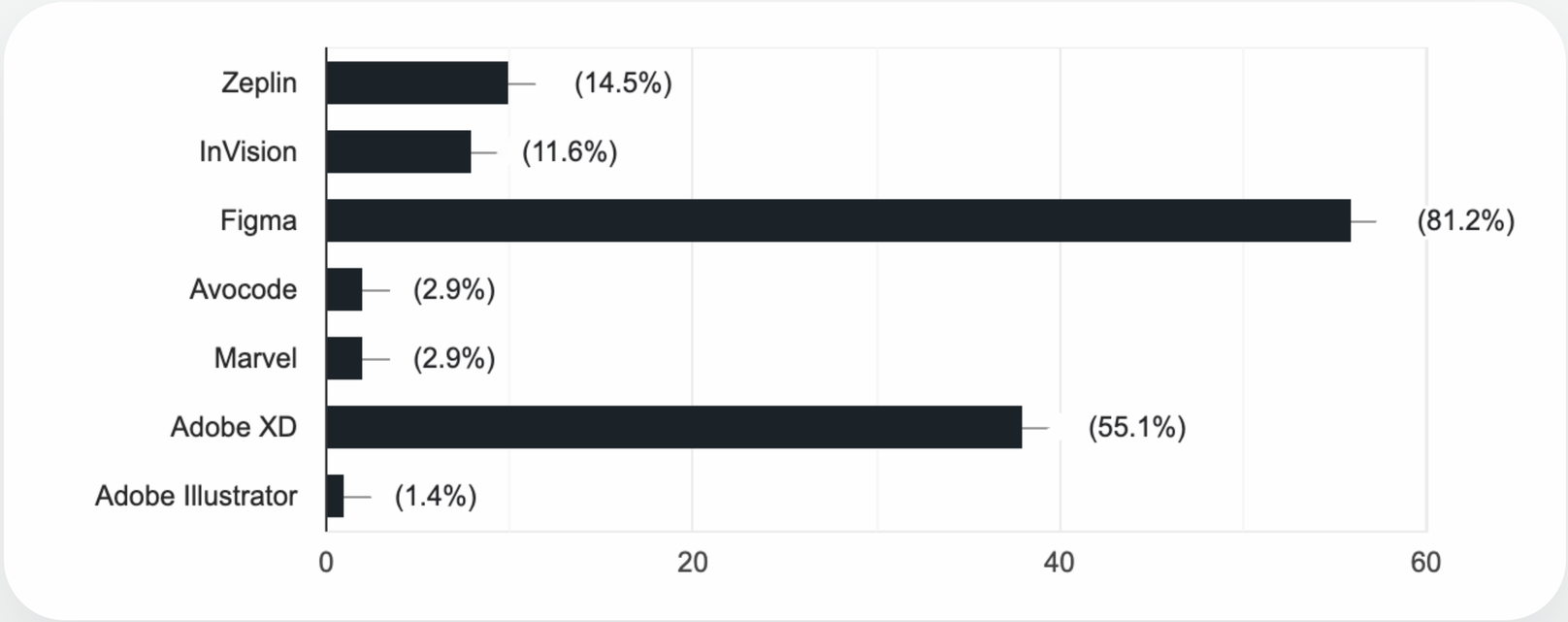



The answer is Yes, we use handoff tools, and they are life savers. Some of the designers use different tools from other colleagues but in the end, the results are fine.

Section commented by:

**Urim Karakashi**
Co-founder & Designer @Honedon

## 27. Which handoff tools have you used?

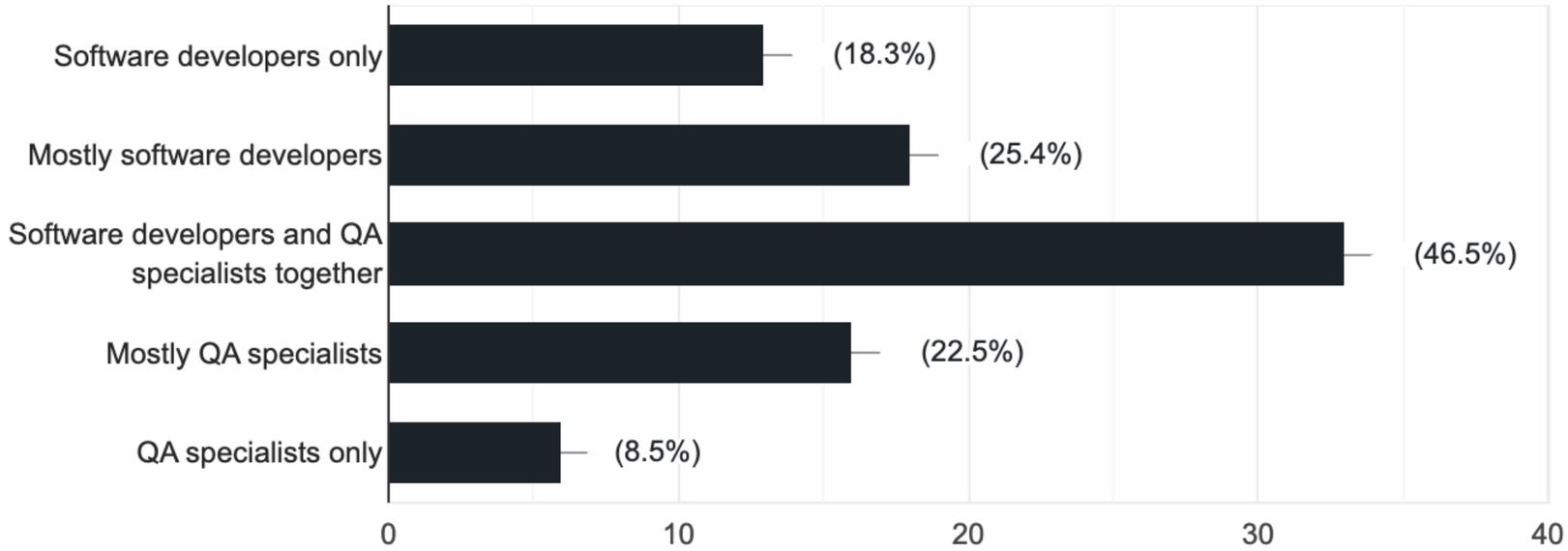



During the last years, the software trends, especially in the UI industry, changed and developed so fast that it was really challenging to adapt and choose which one is a great fit for us. But if we have to specify 2-3 software, they are Figma, Illustrator, and Photoshop.

Section commented by:

**Urim Karakashi**
Co-founder & Designer @Honedon

## 28. Who was responsible for testing in your software development teams?




Section commented by:

**Albana Berisha**
Test Manager @Imbus Peja and Vodafone DE

SEE THE COMMENT ON THE NEXT PAGE

## Developers only:

This causes a lack of quality and user experience. Using only unit testing and exploratory testing is not sufficient on projects/products.

Such an approach can cause errors/failures in the integration process (when components are tested together) and affect the pre-release.

## SD and QA:

Together Such an approach is highly preferred if the company follows the methodology of SDLC and STLC.

Starting from testing branches individually (taking the developer's branch to test on a certain environment), moving to testing environment (where the dev's branch merges into master) and reaching to demo environment in which system/integration/acceptance testing is all passed before the prod-release.

## Mostly software developers

Partly same opinion as option 1.

## Mostly QA Specialists

I do believe that QA can manage the testing on all phases, but in my opinion, collaborating with developers can produce bug prevention rather than bug finding.

## QA Specialists only

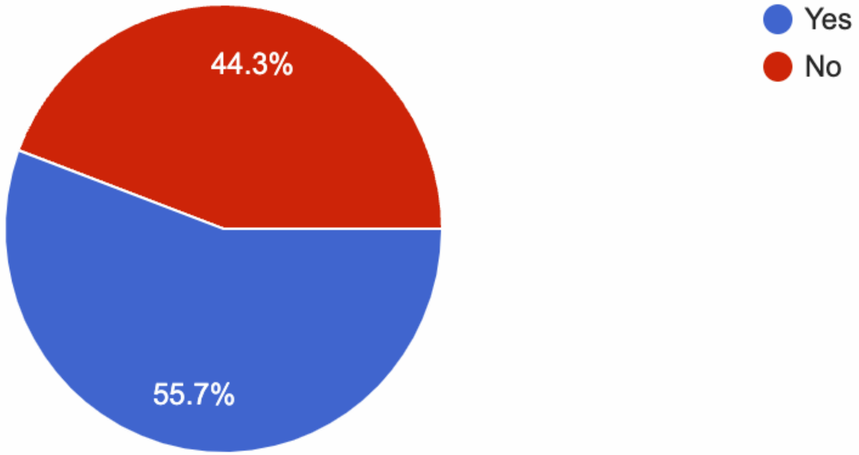
The so-called 'Software Testing'. The QA (Tester) gets involved only in the testing phase, ignoring all the previous phases (Business requirement, Design, Development). I do believe this is not the right approach of companies if Agile is the work methodology chosen to follow.

Section commented by:

**Albana Berisha**

Test Manager @Imbus Peja and Vodafone DE

## 29. Have you performed software tests yourself during the last year?




The lead/manager should have a role in guiding the team. It does not mean specifically on executing a test case.

Performing the software testing can imply:

Section commented by:

**Albana Berisha**
Test Manager @Imbus Peja and Vodafone DE

SEE THE COMMENT ON THE NEXT PAGE

## Asking the right questions regarding the product/project/feature

There is a high change of PMs and Developers misunderstand parts of a business requirement.

A QA should always be involved in product refinement meetings. The ability of the QA to catch details at an early stage of project/product scope increases the assurance of developing proper functionalities as requested from clients.

## Advice the team for best approach/tools to use

There are a lot of tools for easy documentation, testing and reporting.

Finding the right tools and approach can reduce the time consuming of the testing process and reporting on time before the deadline of the sprint. Efficient usage of the right tools = more time on testing or retesting bug-fixes.

## Analyzing the acceptance criteria and reviewing them while task assigning/development with PMs

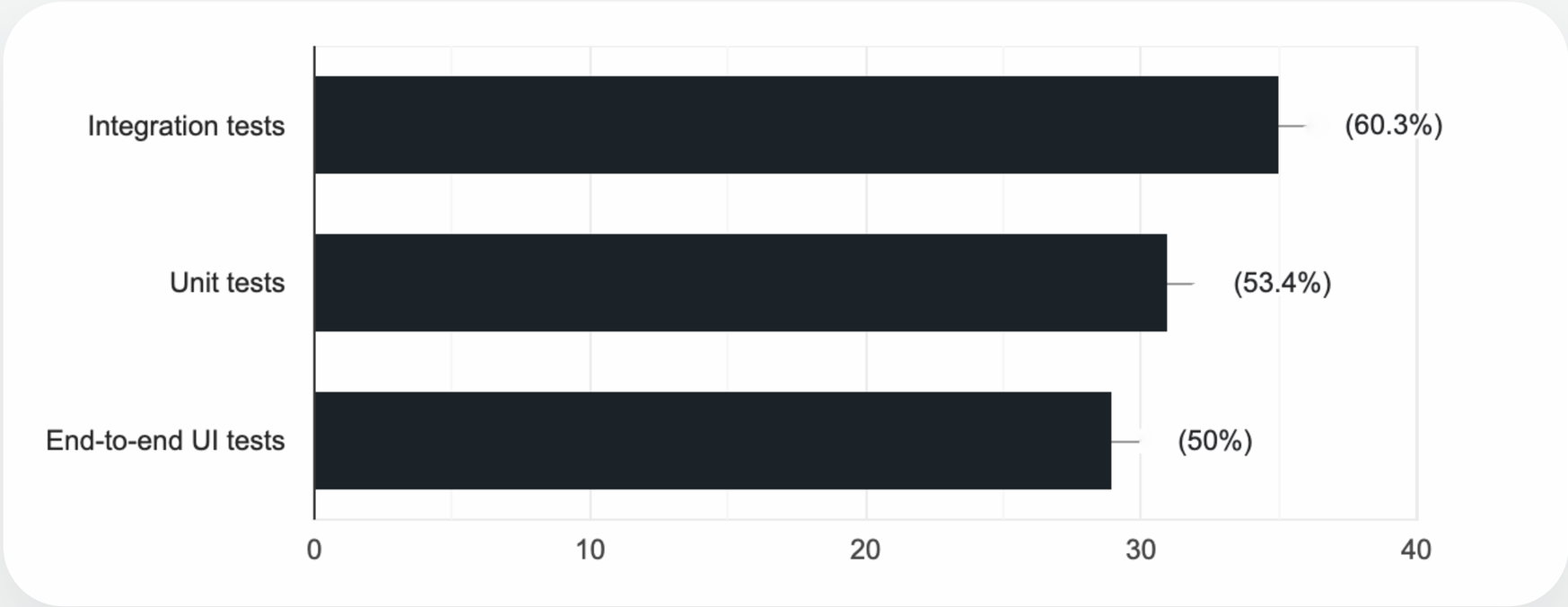
While PMs are more business-driven, Developers are focused specifically on the technical side. The responsibility of a QA is to follow up all the tasks description, requirements and most important:

Acceptance criteria. If the acceptance criteria are not listed properly, there is a chance of a partly-developed task. Such cases lead to creating new out-of-scope tasks, returned tasks into development again and overloading the sprint.

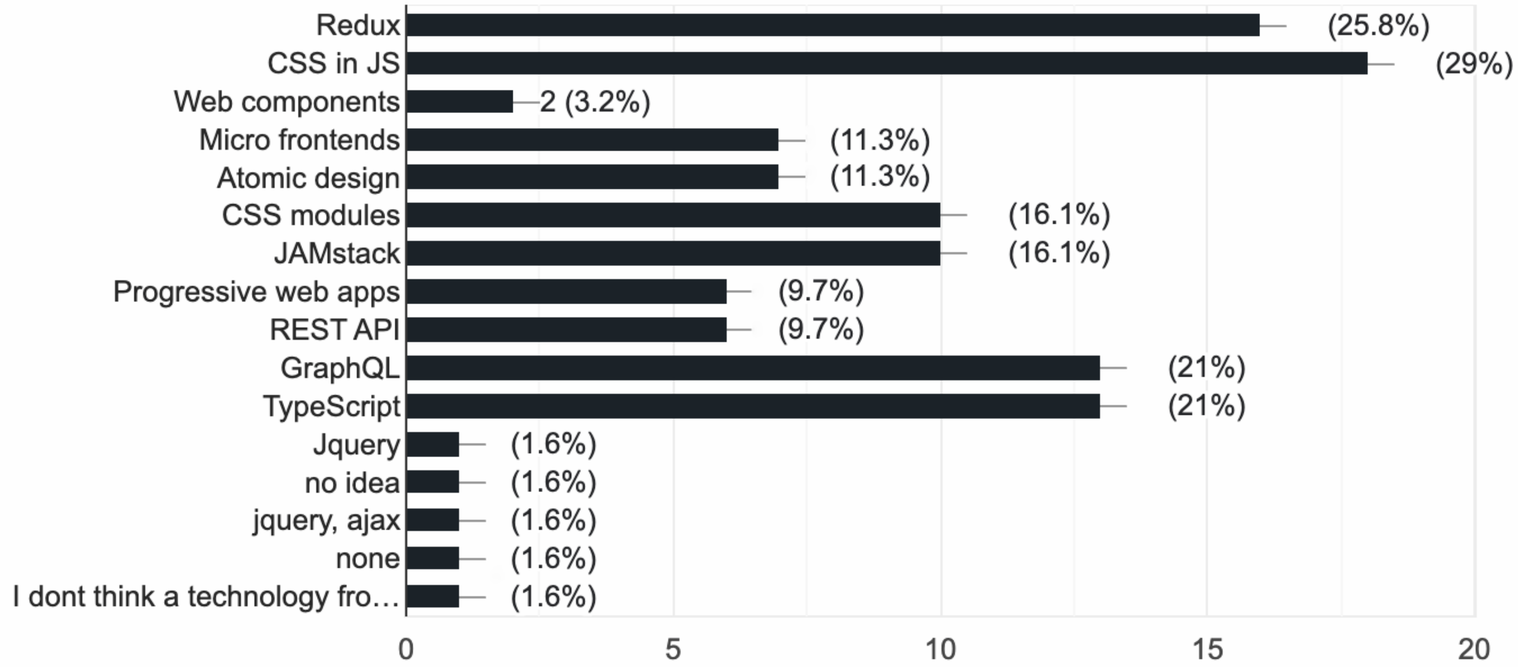
Section commented by:

**Albana Berisha**
Test Manager @Imbus Peja and Vodafone DE

## 30. What kinds of tests have you performed yourself?



## 31. Which of these trends/solutions will be pretty much dead in 3 years from now?




Section commented by:

**Lend Kelmendi**
Software Engineer & Team Leader at Kutia

SEE THE COMMENT ON THE NEXT PAGE

## Typescript

Since Typescript has become a staple in the front-end game, the only way for this trend to go this way in the next 3 years, is if JavaScript adopts Typescript features.

## CSS IN JS:

While looking at the performance side CSS in JS might sound like a good idea, changing the way we used to work with styles might be counterproductive and add some extra development time, that might be one of the reasons why the trend is in this way.

## Redux:

When React started, redux was the go-to tool for global state management even as the only tool available it was a hassle to set up and use effectively that's why we saw the rise of hooks + context API or something more modern like react query.

Even the creator of Redux tells us to not use it, this is reflected in the chart. .

Section commented by:

**Lend Kelmendi**
Software Engineer & Team Leader at Kutia

# A big THANK YOU to our collaborators who gave their outstanding insights on this report!

Alban Sahiti, CEO at TENTON

Albana Berisha, Test Manager at Imbus Peja and Vodafone DE

Arbena Musa – Web Developer at Sogody

Çelik Nimani, CEO at FRAKTON • Part of Kin + Carta

Granit Doshlaku, Co-founder & SEO Director at Manaferra

Hekuran Doli, Founder & DevSecOps Engineer at Cingerr.com

Lend Kelmendi, Software Engineer & Team Leader at Kutia

Lorik Mullaademi, CEO at Sogody

Milot Hajredini, DevOps at BBros

Urim Karakashi, Co-founder & Designer at Honedon

# Team

**Liridona Syla**
Executive Coordinator &
Content Writer

**Rona Lohaj**
Designer &
Marketing Manager

**Faton Selishta**
CTO

**Arianit Fazliu**
CEO

**Shpat Paçarada**
Creative Director

# Grateful for our sponsors and their support

This report was presented at Coders Friendship Gathering 2022 – an event organized by Kutia during the annual KosICT event organized by STIKK.
The goal of this event is to build new bridges of collaboration between Balkan countries in the ICT sector.

## Coders Friendship Gathering was sponsored & supported by:

USAID
NGA POPULLI AMERIKAN
OD AMERIČKOG NARODA

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

**Swiss Agency for Development and Cooperation SDC**

EYE Enhancing Youth Employment