Tab Tab

The State of Components 2023

Supernova.io surveyed 100+ designers and developers across various industries to understand the current state of components. Explore the results and insights in our "State of Components 2023" report!

+ Button

Badge

Dropdown

SONY

achieve

bumble

WELLS FARGO
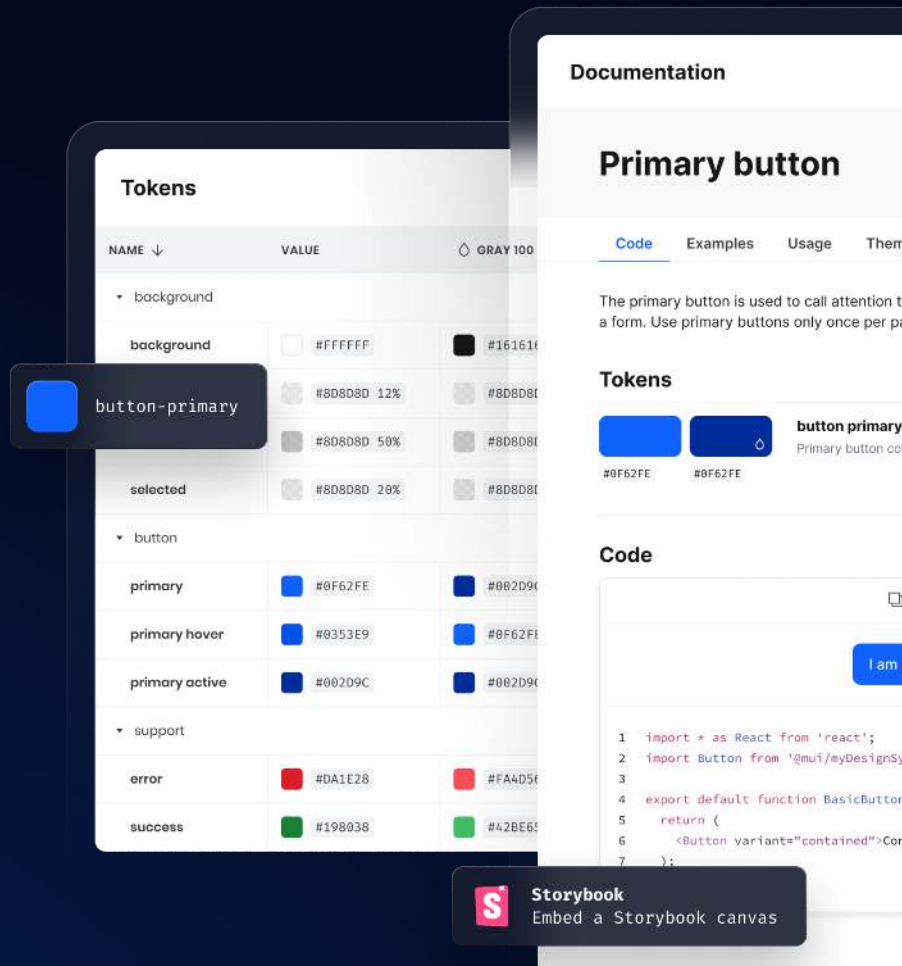
Workiva®

NCR

RAPID7

SEMRUSH

CARVANA

Trustpilot

Introduction

The report is brought to you by:

supernova

Supernova is an end-to-end design systems management platform that empowers your design and engineering teams to work better together.

What are design system components?

Components are the essential building blocks of design systems, which are crucial in creating consistent and efficient user interfaces. They help designers and developers maintain a cohesive visual language, streamline the design process, and ensure that the user experience remains consistent across different platforms and devices. By using components, teams can reduce the need for custom solutions, improve collaboration, and increase the overall efficiency of the product development process.

In a rapidly evolving digital landscape, design systems play a pivotal role in shaping seamless and cohesive user experiences. The "State of Components 2023" survey, conducted by Supernova, discovers trends, challenges, and innovations within the realm of components.
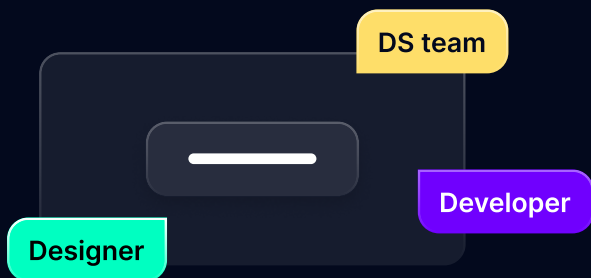
Key Takeaways

The first section of the report walks through the 4 key takeaways from the survey results. We found that components ownership and metrics vary by company size. We also see that teams want to prove the value of components but aren't getting stakeholder input.

Let's dive in.

DS team

Designer

Developer

Ownership of components varies by company size.

We saw a stark difference in who owns components based on company size. 41.3% of companies with 250+ employees had a dedicated design system team that owned and maintained components, compared to just 7.7% of companies with less than 250 employees.

In addition, companies with 250+ employees are more likely to have UX writers. Of the respondents who said they had UX writers to document components, 94.4% of them work at companies with over 250 employees.

It makes sense to see that, as companies grow, roles and ownership become more specialized.

Badge ⚠ 2.84:1
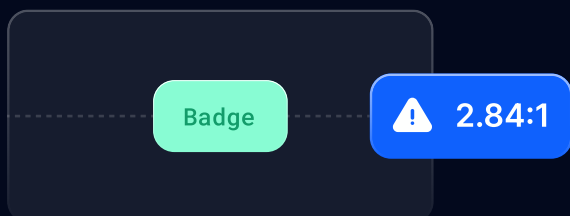
Components aren't as accessible as they should be.

Components should be built with accessibility baked in. However, only 36.5% of respondents report that 75%+ of their component library is compliant with WCAG 2.1. In fact, 34.8% don't require that components meet accessibility standards at all, and only 55% include accessibility in component documentation.

There is a lot of room for improvement here, and we hope to see these numbers increase in the next State of Components report.

Adoption is the primary metric to determine effectiveness, but small companies care more about consistency and efficiency.

For most respondents (52.2%), the primary metrics for measuring component effectiveness are adoption rate and consistency. However, when we cut the data by company size, companies with less than 50 employees prioritize consistency (37.5%) and efficiency (33.3%).

Only 4.2% said they care about adoption rates. This suggests that smaller teams are more focused on the utility of components and are less concerned with measuring who's using them.
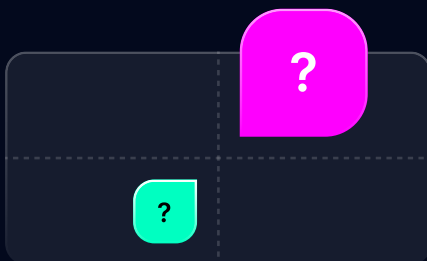
People want to prove the value of components but aren't getting stakeholder input.

73.3% of respondents said it was important to gain quick wins and demonstrate value with their initial components, yet only 28.9% reported they collected input from stakeholders and prioritized accordingly. It's difficult to prove value when you don't know what executives and stakeholders care about.

Just over half (50.5%) of participants create a roadmap for component development, with only 14.4% sharing one publicly. This highlights the need for clearer plans and more communication with stakeholders to make sure everyone is on the same page.

02

Participants

The "State of Components 2023" survey was open from October 26 to November 22, 2023. We received 100+ responses from designers and developers working at companies across industries and company size to provide a comprehensive overview of the current state of components.

What is your role at your company?

51% of the respondents are Designers in a Design System Team.

Developers in design system team 17.5%

Designers 22.8%

Developers 8.8%

Other 8.8%

Designers in design system team 50.9%

What is the size of your company and which industry does it operate in?

37.0% of respondents work at large organizations (1000+ employees), and the most common industry represented is Information Technology (33.3%).

Information Technology 33.3%

E-commerce 13.9%

Finance 11.1%

Education 7.4%

Hospitality, Telecommunications, <5%
Healthcare, Entertainment,
Energy, Logistics, Insurance,
Gaming, Automotive, Other

2 – 49 employees 11.1%

50 – 249 employees 22.2%

250 – 999 employees 29.6%

1 000 – 4 999 employees 23.1%

5 000 – 9 999 employees 4.6%

10 000+ employees 9.3%
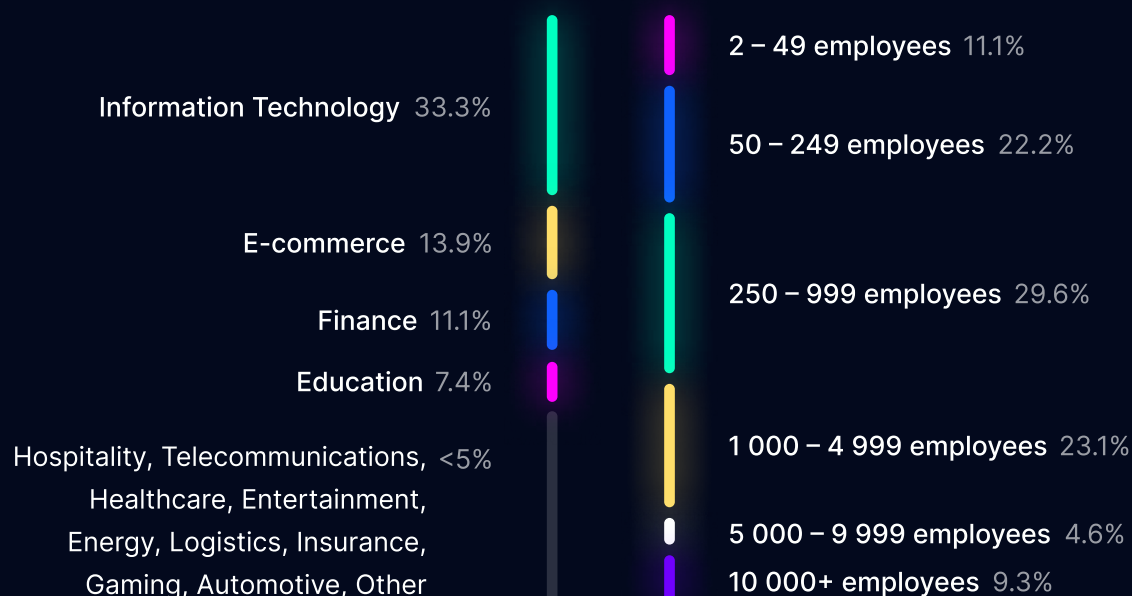
What is your primary source of truth for components?

Before we dive into the results, let's take a look at where everyone's component library lives and how long they've been around.

We're surprised to see that only 14.9% of people leverage a design system documentation or reference site as their primary source of truth for components, considering 68.4% of respondents are on a dedicated design system team. Instead, 44.7% of respondents said that they rely on their design tool as the primary source of truth for components.
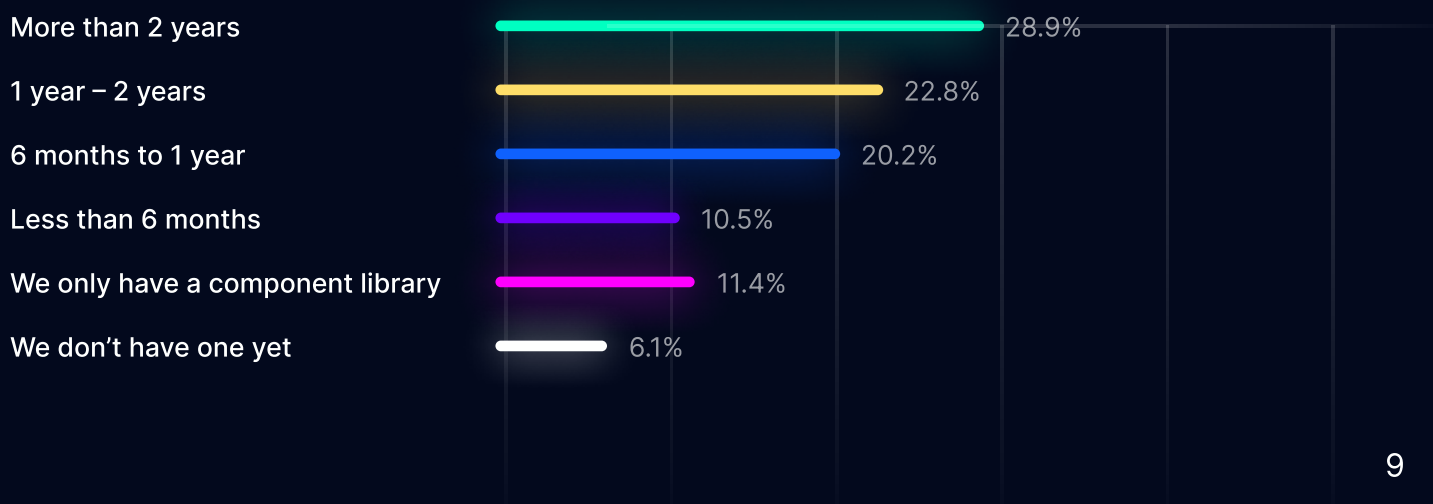
DS documentation or reference site 14.9%

Code repo 15.8%

Storybook 21.1%

44.7% Design tool
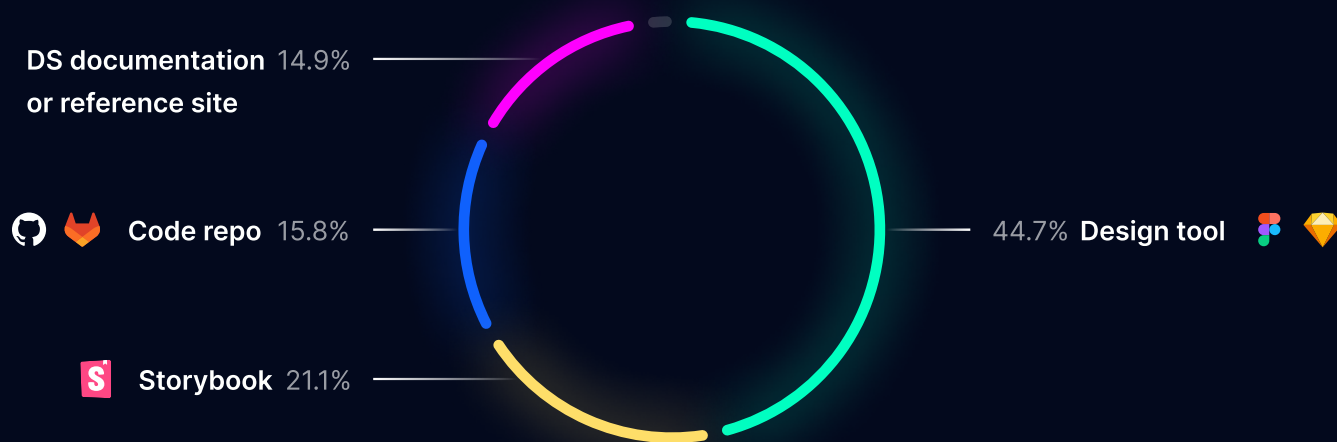
How long has your design system been live?

29% of respondents work on design systems that have been live for over 2 years. However, 17.5% don't have a design system at all or only have a component library.

More than 2 years		28.9%
1 year – 2 years		22.8%
6 months to 1 year		20.2%
Less than 6 months		10.5%
We only have a component library		11.4%
We don't have one yet		6.1%

How many components are currently in your library?

We also found that most libraries (46.5%) have 21-50 components, while 16.7% have over 100 components.

Less than 20	14.9 %
21 – 50	46.5 %
51 – 100	21.9 %
More than 100	16.7 %
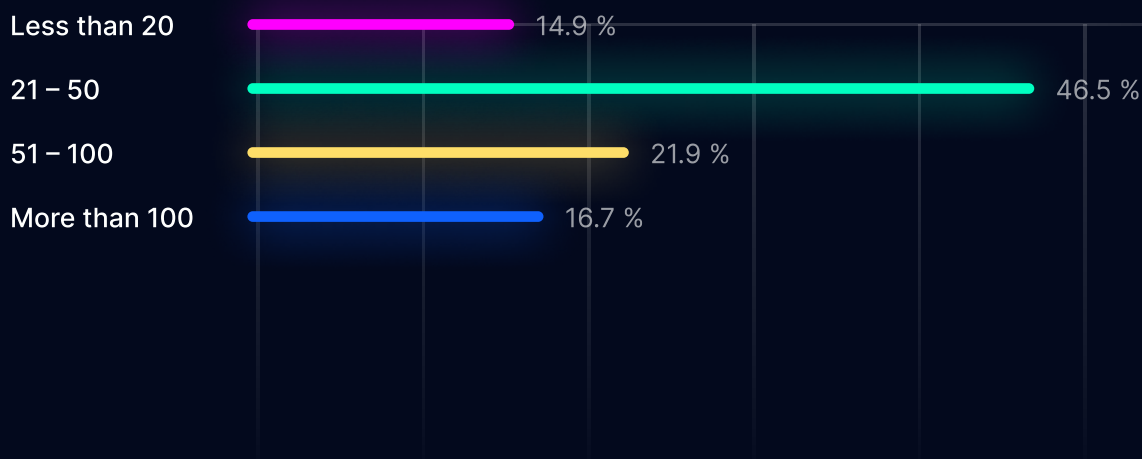
Does your design system include documented design patterns?

Patterns are well established across design systems. The majority of respondents (58.4%) have patterns, alongside their components.

No 34.5%

58.4% Yes

I'm not sure 7.1%
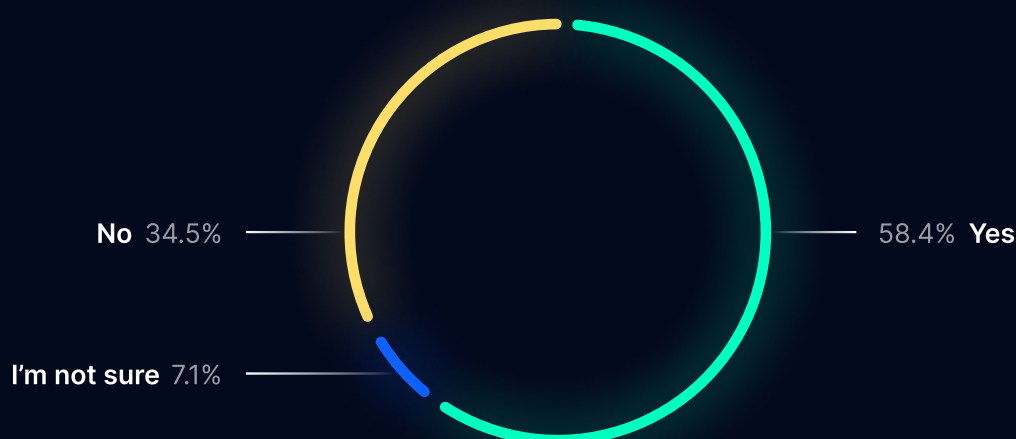
Creating Components

In this section, the report explores how the development of components is prioritized and planned. We also look at what technologies are used and what quality standards components need to meet.

What do you prioritize in the initial phase of design system development?

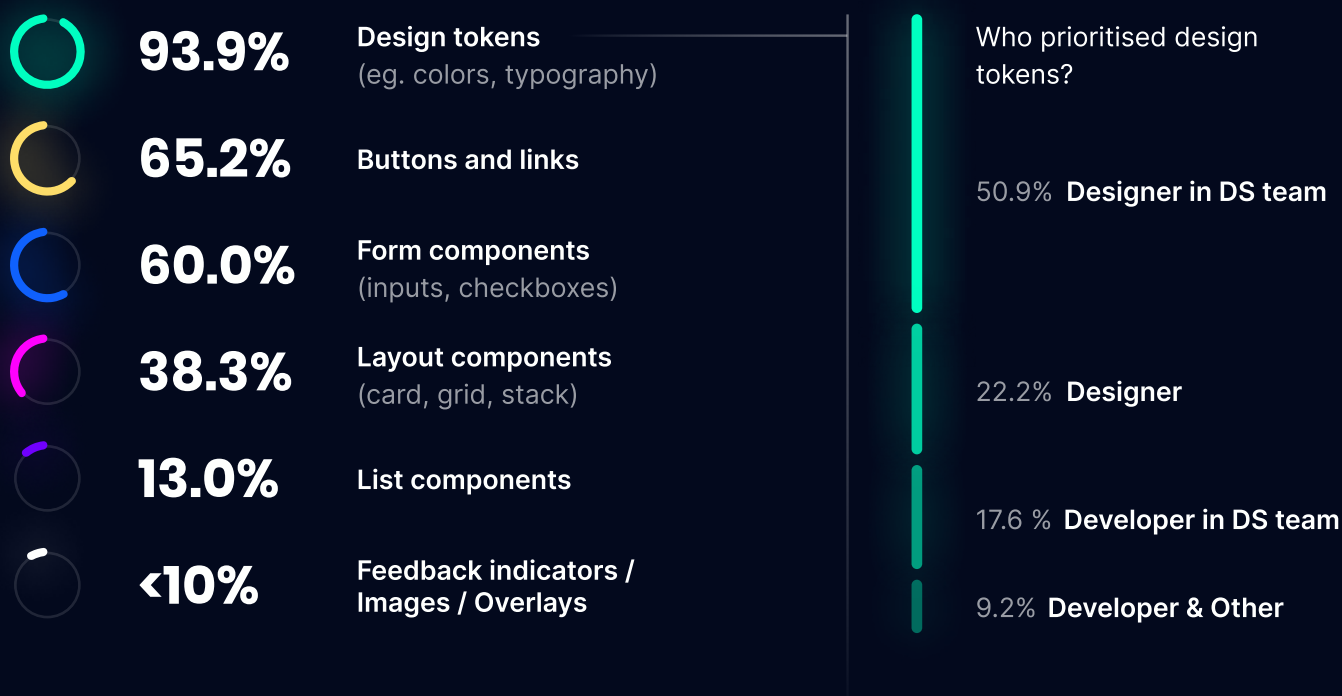During the initial stage of design system development, 94% of the participants prioritized design tokens. When we looked at which positions focused on design tokens first, we found that it was a priority across the board.

After design tokens, buttons and links are the most prioritized components. This aligns with the data we saw from Supernova users who use buttons as the most common design system component. However, our data showed that cards were the second most common component, while survey participants prioritized form components instead.

93.9%	**Design tokens** (eg. colors, typography)	
65.2%	**Buttons and links**	
60.0%	**Form components** (inputs, checkboxes)	
38.3%	**Layout components** (card, grid, stack)	
13.0%	**List components**	
<10%	**Feedback indicators / Images / Overlays**	

Who prioritised design tokens?

50.9% **Designer in DS team**

22.2% **Designer**

17.6 % **Developer in DS team**

9.2% **Developer & Other**

"Focus on the most valuable component. So if you're really lagging behind, obviously only create what is going to be really valuable. And the way I would go about doing that, if you had an existing product, make a list of every component that's inside the product. See which ones are duplicated the most, see which ones have the highest level of technical effort and correlate those and focus on the ones that are the top of those lists at the end. So focus on value and really manage your expectations there."

Henry Daggett

Design Systems Lead & Product Designer / Societe Generale
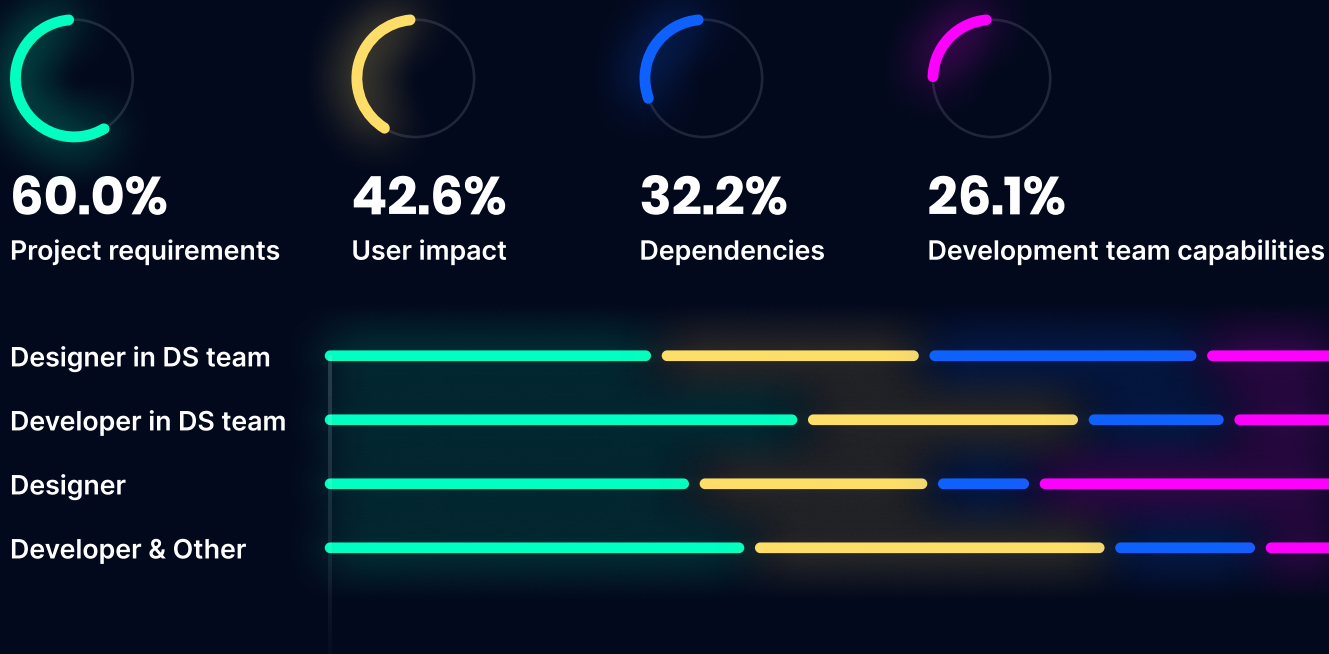Beyond Launch: Nurturing & Sustaining Design Systems

Deciding which component to build first is no small feat

Q: What are the main factors you consider when deciding which component to build first?

Project requirements (60%) and user impact (42.6%) are the main factors considered when deciding which component to build first. Striking a balance between these factors optimizes development efficiency. It ensures that the team focuses on building components that provide maximum impact with minimal friction.

When we look at the breakdown by role, project requirements continue to be the main factor that all designers and developers (regardless of whether or not they were on a dedicated design system team) looked at.

The second most common factor varied. Engineers prioritized user impact, while design system designers considered dependencies and product designers considered development team capabilities.

60.0%
Project requirements

42.6%
User impact

32.2%
Dependencies

26.1%
Development team capabilities

Designer in DS team

Developer in DS team

Designer

Developer & Other

"The first rule we implemented for our team, when we started scaling from a very small team serving just one product to team serving everyone in the company, was that we only build what's requested by their teams."

Dmitry Belyaev
Principal Frontend Engineer / Booking.com
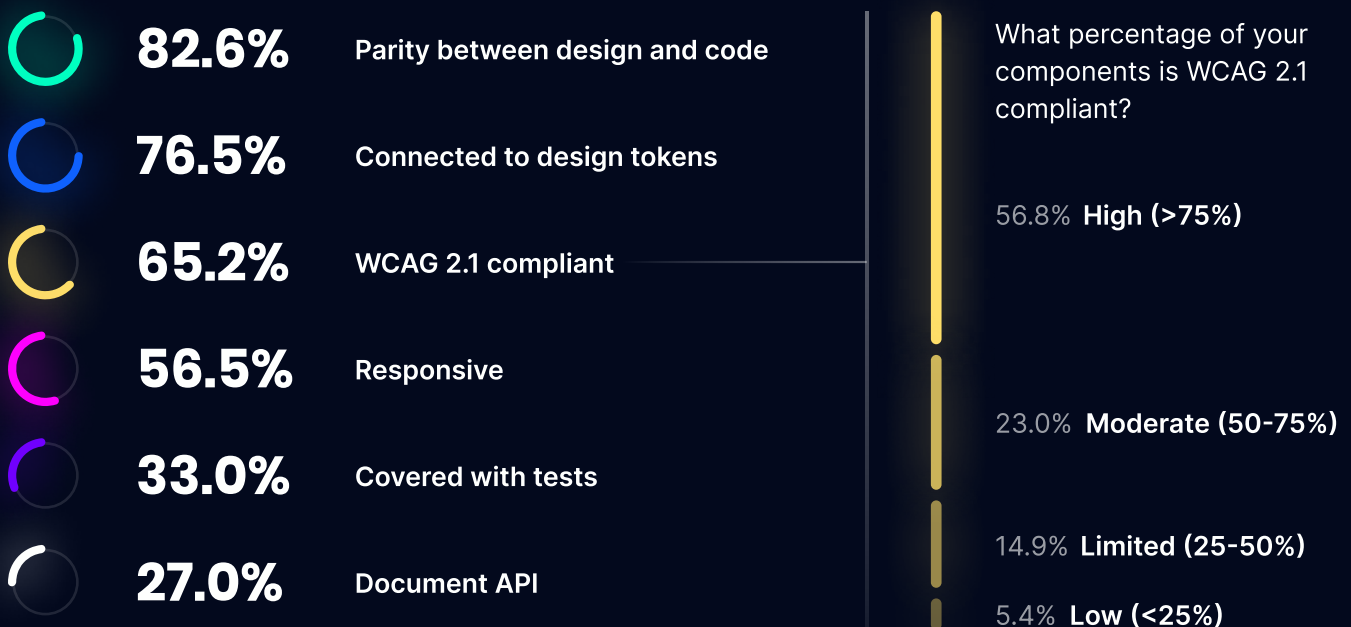Components: Building the Building Blocks

13

What quality standards are you required to meet for your components?

When creating components, key criteria include ensuring there's parity between design and code (82.6%) and that components are connected to design tokens (76.5%). Meeting accessibility standards (65.2%) is crucial for inclusivity.

While it's great to see WCAG 2.1 compliance in the top 3 quality standards, there's still a lot of room for growth from an accessibility perspective, as only 36.5% of respondents report that 75%+ of their component library is compliant with WCAG 2.1.

We were surprised to see that responsiveness and test coverage were far less important, though this may be because most of the participants are designers, not developers.

82.6% — Parity between design and code

76.5% — Connected to design tokens

65.2% — WCAG 2.1 compliant

56.5% — Responsive

33.0% — Covered with tests

27.0% — Document API

What percentage of your components is WCAG 2.1 compliant?

56.8% **High (>75%)**

23.0% **Moderate (50-75%)**

14.9% **Limited (25-50%)**

5.4% **Low (<25%)**

"In a lot of places where I was, it was actually that it was more like developers and designers trying to push accessibility and clients saying, what's that exactly? And then you come, you're like, yeah, you know, there's a European law that is supposed to be in different countries in 2025, something like that, that's going to make it even kind of try to regulate a little bit more."

Stéphanie Walter
UX Researcher and Designer
Scaling Accessibility Through Design Systems

React is the most commonly used technology

The widespread usage of React (39.1%) in design systems can be attributed to its extensive community support, robust ecosystem, and flexibility, making it a popular and versatile choice for building components that cater to diverse project requirements.

Some of the other technologies teams used include Android, Compose, Swift, Flutter, Stencil.js, and Tailwind.

Svelte 1.4%
Vue.js 5.8%
Angular 7.2%
Web components 10.6%
HTML/CSS 24.6%
39.1% **React**
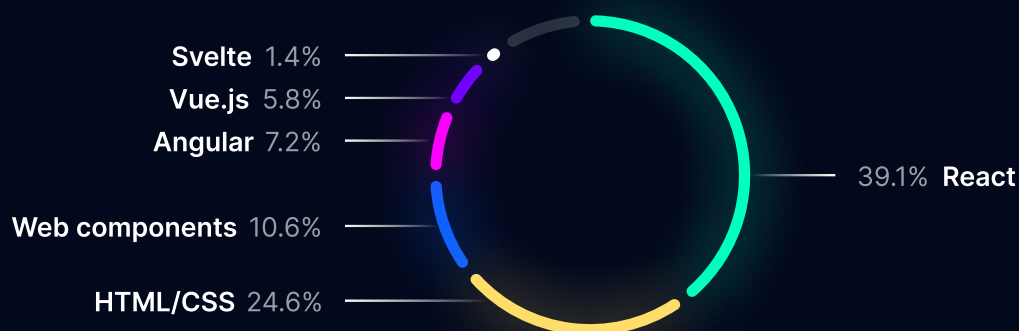
CSS-in-JS is the most often-used approach in defining the visual appearance of components

Our respondents have shared the usage of other ways as well like Sass/ SCSS, Tailwind, Vanilla CSS, etc. This diversity in styling approaches reflects the adaptability of design systems, allowing teams to choose methods that align with project requirements, team expertise, and the evolving landscape of web development.

Vanilla CSS 5.0%
CSS modules 12.6%
Tailwind 15.7%
21.4% **CSS-in-JS**
20.7% **Sass / SCSS**
19.5% **I don't know**

How do you test and validate design patterns?

The majority of respondents (71.3%) test and validate design patterns by getting feedback from designers and developers on their team. Some peop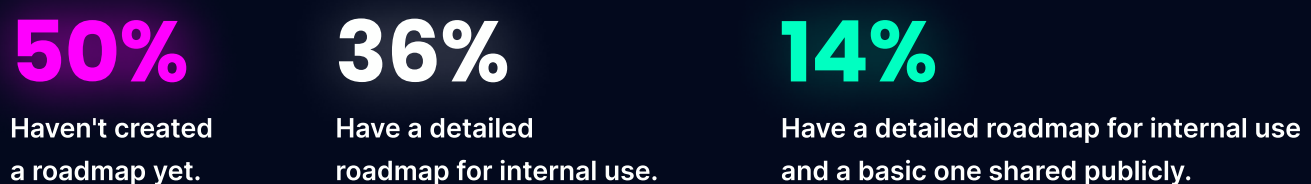le also test patterns to verify effectiveness in different contexts (48.7%) and to ensure patterns meet user needs (47.0%).

71.3%   Solicit feedback from designers and developers within your team

48.7%   Test patterns in different project contexts to verify their effectiveness

47.0%   Conduct usability testing to ensure patterns meet user needs

28.7%   Use analytics and user feedback to asses pattern performance

14.8%   Use automated testing tools to identify issues
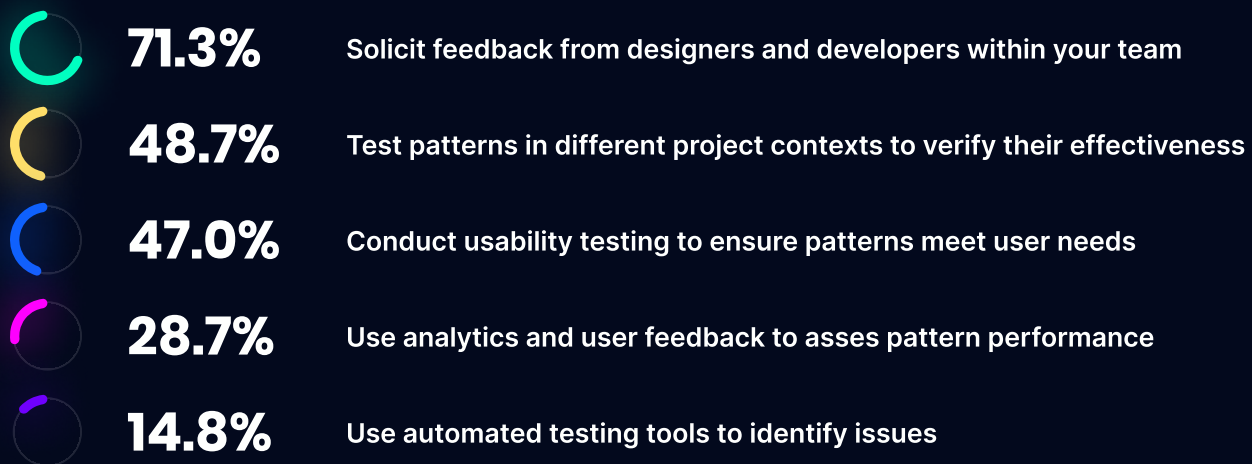
Roadmaps for developing components

Q: Do you have a roadmap outlining the sequence of component development?

Creating a roadmap for developing components is a critical aspect of a design system. A well-defined roadmap helps guide the team, aligns stakeholders, and ensures a systematic and strategic approach to building and evolving the design system.

50%
Haven't created
a roadmap yet.

36%
Have a detailed
roadmap for internal use.

14%
Have a detailed roadmap for internal use
and a basic one shared publicly.

> "Absolutely share your roadmap — it's a great way to validate you're building what people want, and hopefully build some excitement for people to adopt."
>
> **Maya Hampton**
> Senior Product Manager, Design System / REI Co-Op
> Navigating Adoption: Driving Design System Usage & Engagement Q&A

Demonstrating value with initial components

Q: How important is it to gain quick wins and demonstrate value with the initial component?

The importance of quick wins in gaining stakeholder buy-in and demonstrating the value of the design system was evident. 73.3% of respondents emphasized the need to showcase tangible results early in the development process, with 27.6% saying it is extremely important.

8.6%	11.2%	6.9%	45.8%	27.5%
Not important at all	Somewhat unimportant	Neutral	Somewhat important	Extremely important
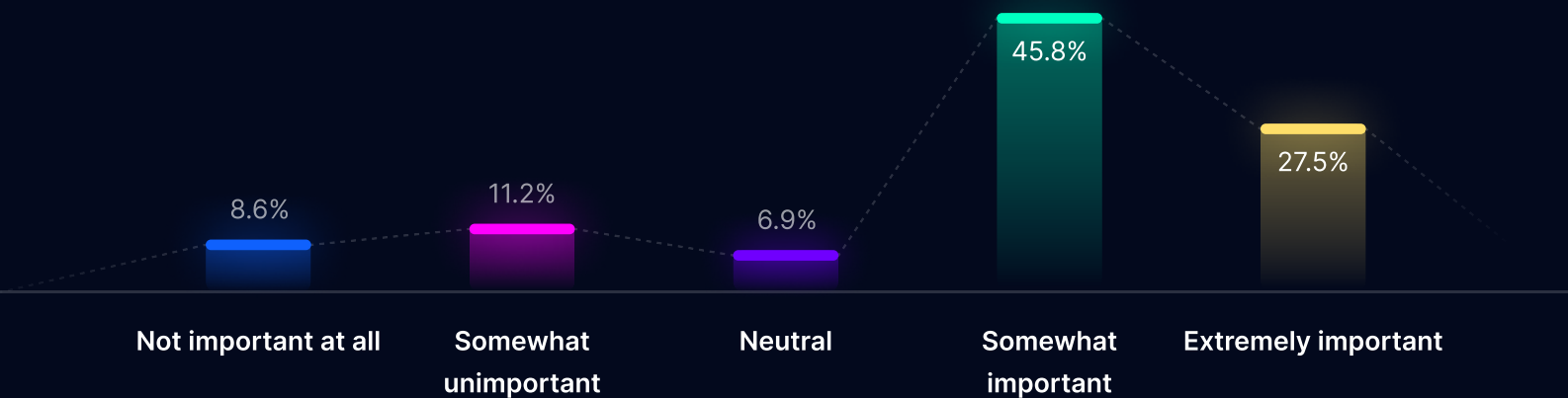
Have you gathered input from stakeholders on component priorities?

~ 51.3%

We've collected some but will primarily rely on our internal assessment.

✓ 28.7%

Yes, we've collected input and will prioritize accordingly.

✕ 17.4%

No, we haven't collected stakeholder input.

"Design systems are meant to reduce friction for designers and engineers. Design systems are there to create consistency and to make it easy to create designs and products. To do that, we need to make sure that our system meets the needs of our product friends and is built in collaboration with them."

Lauren Beatty

Staff Engineer / Zapier

Guide on how to foster better collaboration and communication.

04

Documenting Components

We examined how decisions about components are documented in the next section, "Documenting Components." The report looks at who documents components and what's included in component documentation, with a deep dive into accessibility, code examples, and variants.

Who documents components?

The vast majority of designers (92.2%) document components, while 61.7% of respondents say front-end developers document components. We also see that 15.7% have UX writers who contribute to documentation. Digging into the data shows that UX writers are more prevalent in companies with over 250 employees. This suggests that as companies grow in size, there's an increased need for more specialized roles.

Role	Percentage
Designers	92.2%
Front-end developers	61.7%
Ux writers	15.7%
Product managers	7.0%
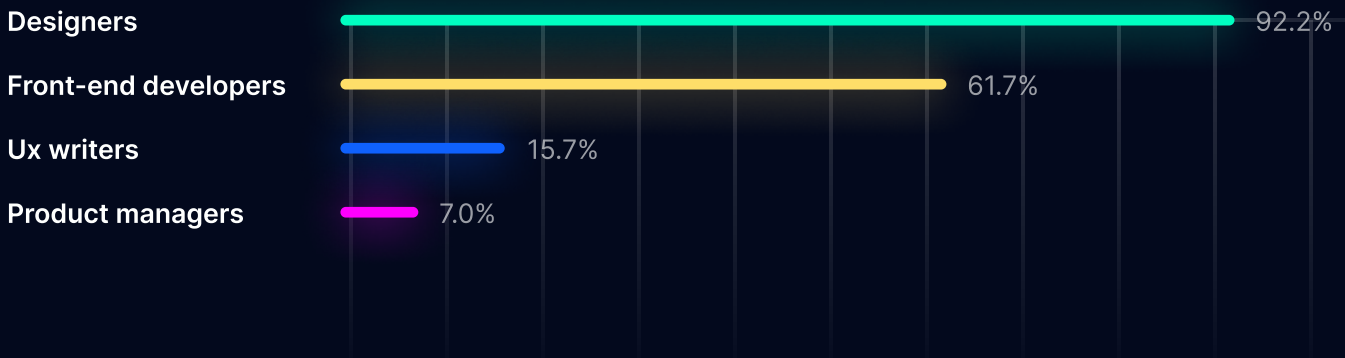
Q: What information do you include in component documentation?

Component documentation is comprehensive, but most teams prioritize sharing best practices (72%), do and don't guidelines (68%), accessibility (55%). Only 17% include dependencies and 21% include demos (eg. video tutorials or walkthroughs).

72%	Best practices	37%	Referencing related components & patterns
68%	Do and don't guidelines	35%	Code comments and annotations
55%	Accessibility information	35%	Visual representation (chart, diagram, etc.)
50%	Interactive components	28%	Versioning / Changelog
43%	Content guidelines	21%	Demos (video tutorials or walkthroughs)
41%	Component brief	17%	Dependencies
39%	Component API	3%	We don't document.

"If you skip everything else and just have your do's and don'ts, that's good. And then accessibility."

Alissa Engstrom
Senior Product Designer, Design System / Apollo.io
Components: Building the Building Blocks

Addressing accessibility considerations

Documenting accessibility considerations for components is a mixed bag. 53.9% provide information related to ARIA attributes and keyboard navigation, while 15.7% include performance optimization tips.

Unfortunately, a third of respondents (33.9%) don't address accessibility considerations in documentation at all.

53.9%
Providing information on ARIA attributes and keyboard navigation

33.9%
Not addressing accessibility considerations

15.7%
Including performance optimization tips

6.0%
Other

"There used to be quite an old school thinking that websites had to be either beautiful and aesthetic or accessible, and never both... Accessibility is everyone's responsibility, and a design system is a great place to assemble allies. "

Geri Reid
Design System Consultant
How to create accessible design system components guide

Providing code examples

Q: What type of code examples do you include for components?

Half of the respondents (49.6%) include basic usage examples in HTML/CSS/JavaScript, offering clear and fundamental illustrations. Additionally, 46.1% opt for live demos, providing users with an interactive and practical understanding of component functionality. A smaller percentage (8.7%) delves into complex implementation scenarios using different frameworks. Notably, 24.3% indicate that they include no code examples, while a minimal 3% opt for other types, including code examples only in Storybook.

Basic usage examples in HTML/CSS/JavaScript — 49.6%

Live demos — 46.1%

No code examples — 24.3%

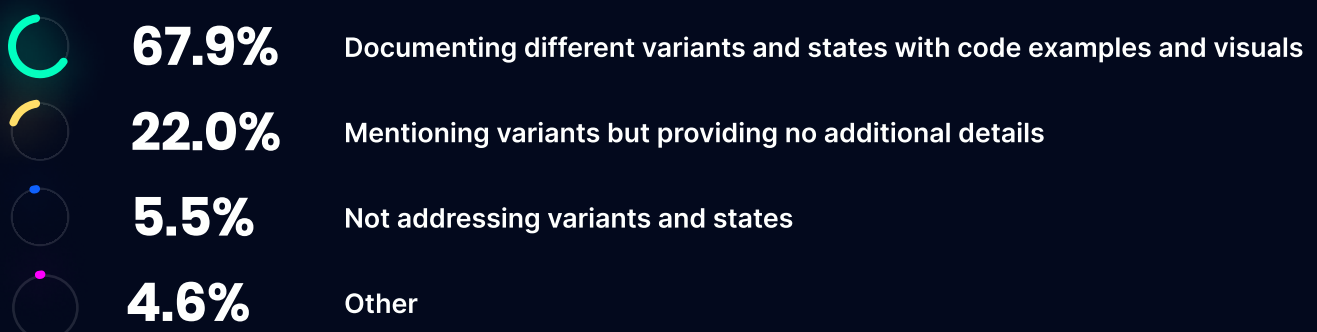**Complex implementation scenarios using different frameworks** — 8.7%
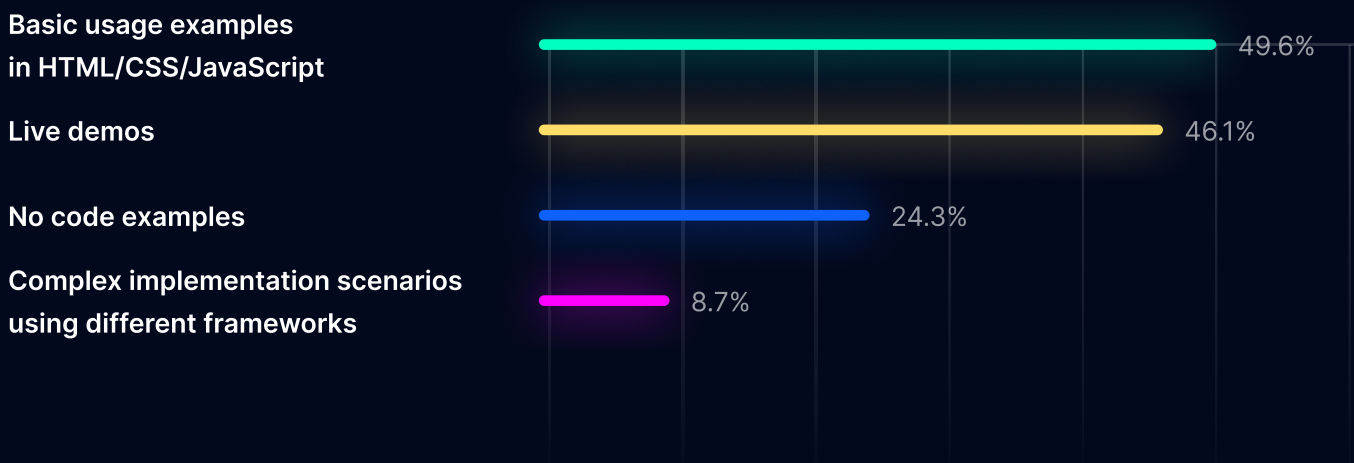
Discussing variants and states

Q: How do you handle variants and states of a component in documentation?

Handling variants and states in documentation ensures that the design system's users have the necessary information to effectively leverage the versatility and adaptability of components in their projects. It's great to see that the majority of respondents (67.9%) document different variants and states with both code examples and visuals.

It aligns with the broader goal of providing comprehensive and user-friendly documentation to empower users to make the most of their components.

67.9% Documenting different variants and states with code examples and visuals

22.0% Mentioning variants but providing no additional details

5.5% Not addressing variants and states

4.6% Other

Adoption

Integrating components seamlessly into design and development workflows is crucial to their success. This section of the report highlights how components are used and measured. It also shares how teams communicate changes and updates to components.

How do you use components in your workflow?

Designers and developers incorporate components into their workflow to establish design consistency (82%). Developers also rely on them to streamline development.

82% To ensure design consistency

59% Collaborate with designers to refine and implement components

57% Rely on components for efficient development

50% Document components usage for their team

35% Customize components to fit specific project needs

16% Conduct usability testing with these components

2.0% Other

Designer in DS team

Developer in DS team

Designer

Other

Developer

"When we need to change or deprecate a component or pattern, we communicate early and often to let everyone know. Capture and share context, decisions made, why we're doing these things, and steps towards the change."

Jeremy Dizon
Lead Product Designer, Design Systems / Disney Streaming
Beyond Launch: Nurturing & Sustaining Design Systems

Communicating updates

Considering the widespread usage of components, it's important to ensure everyone is aware of upcoming changes, especially if an update may break production code.

The vast majority of respondents (73.9%) rely on a designated communication channel or platform to update their team and stakeholders, while 64.3% provide release notes and documentation.
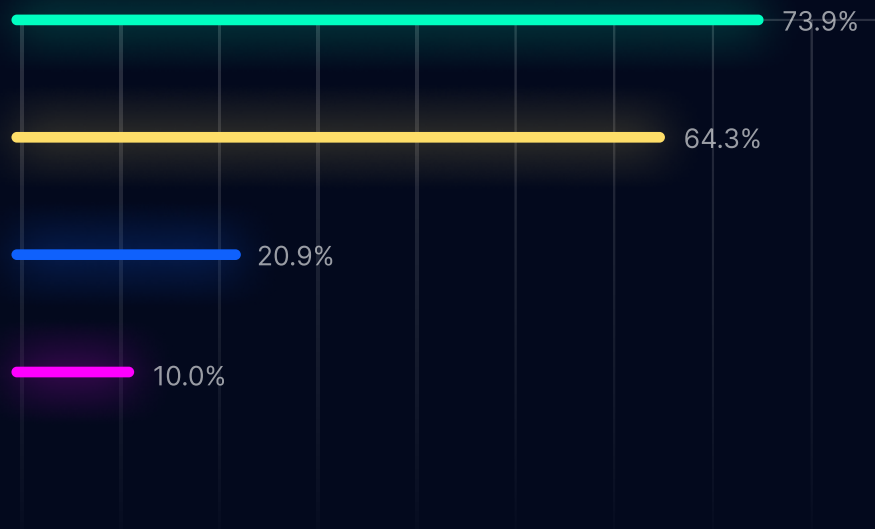
20.9% offer training sessions or workshops for team members. We also saw that people use Jira or comments in Figma as other ways of communication.

Use a designated communication channel or platform.   73.9%
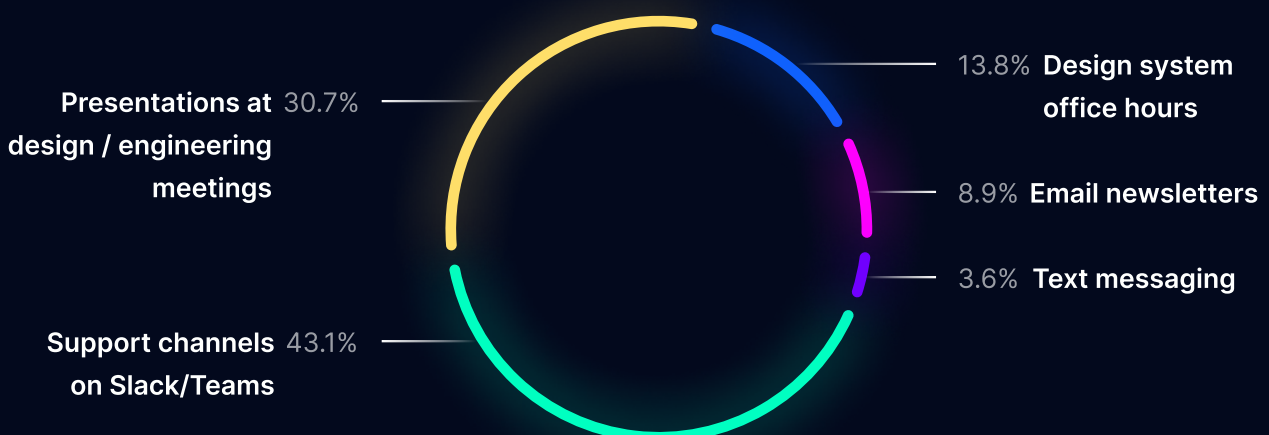
Provide release notes and documentation.   64.3%

Offer training sessions or workshops for team members.   20.9%

Other   10.0%

The most commonly used communication channel was support channels on Slack or Teams (43.1%) followed by presentations at design / engineering team meetings (30.7%).
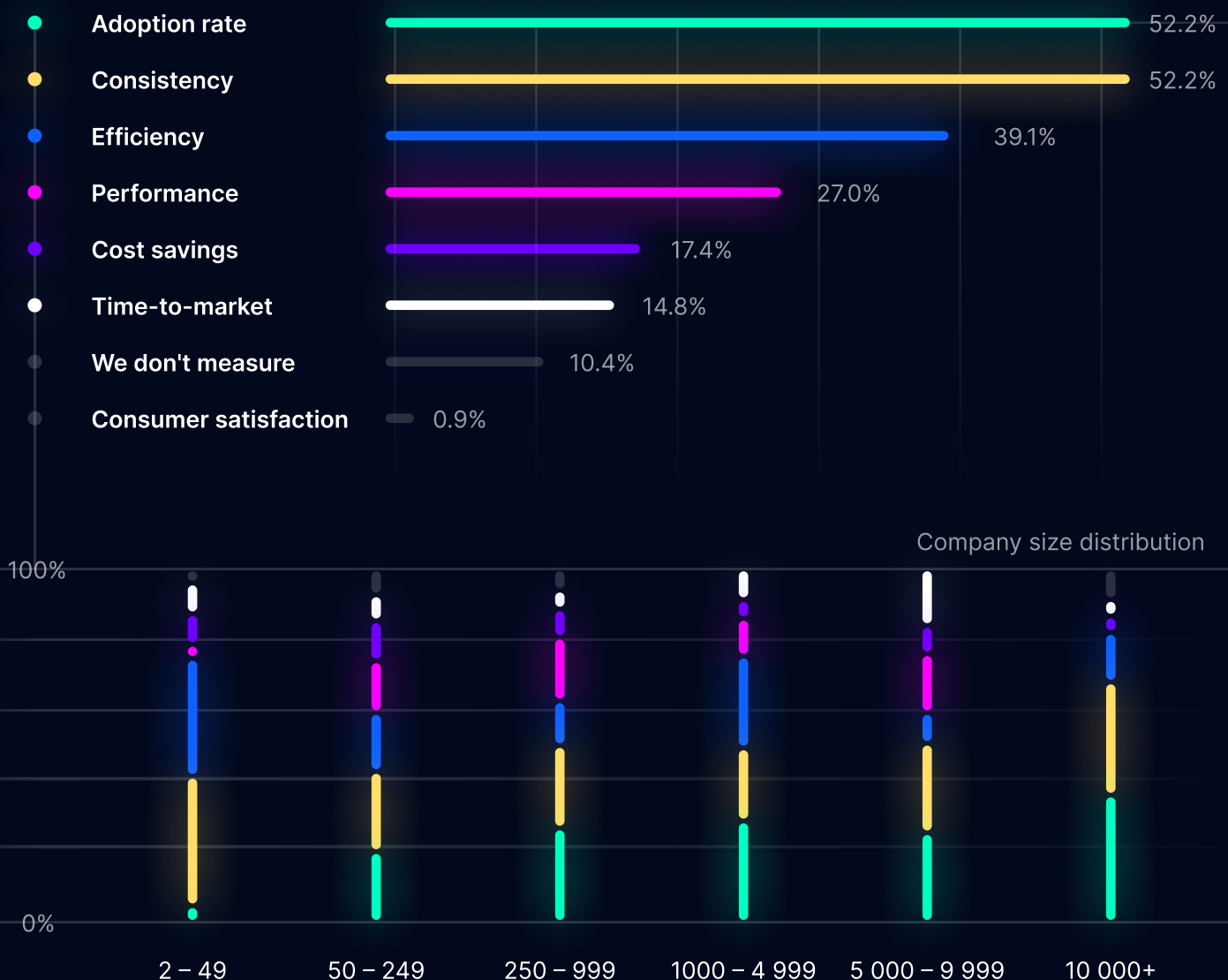
Presentations at 30.7%
design / engineering
meetings

13.8% **Design system office hours**

8.9% **Email newsletters**

3.6% **Text messaging**

Support channels 43.1%
on Slack/Teams

24

Measuring the effectiveness of components

For a majority (52.2%) of respondents, the primary metrics for measuring component effectiveness are adoption rate and consistency. This indicates that most teams are concerned with how widely a component is used and how consistent they are.

However, when we cut the data by company size, we see that companies with less than 50 employees prioritize consistency (37.5%) and efficiency (33.3%). With only 4.2% saying they care about adoption rates. This suggests that smaller teams are focused on the utility of components and are less concerned with measuring how many people adopt them.

It was also surprising to see that 10.4% of respondents don't measure the effectiveness of components at all.
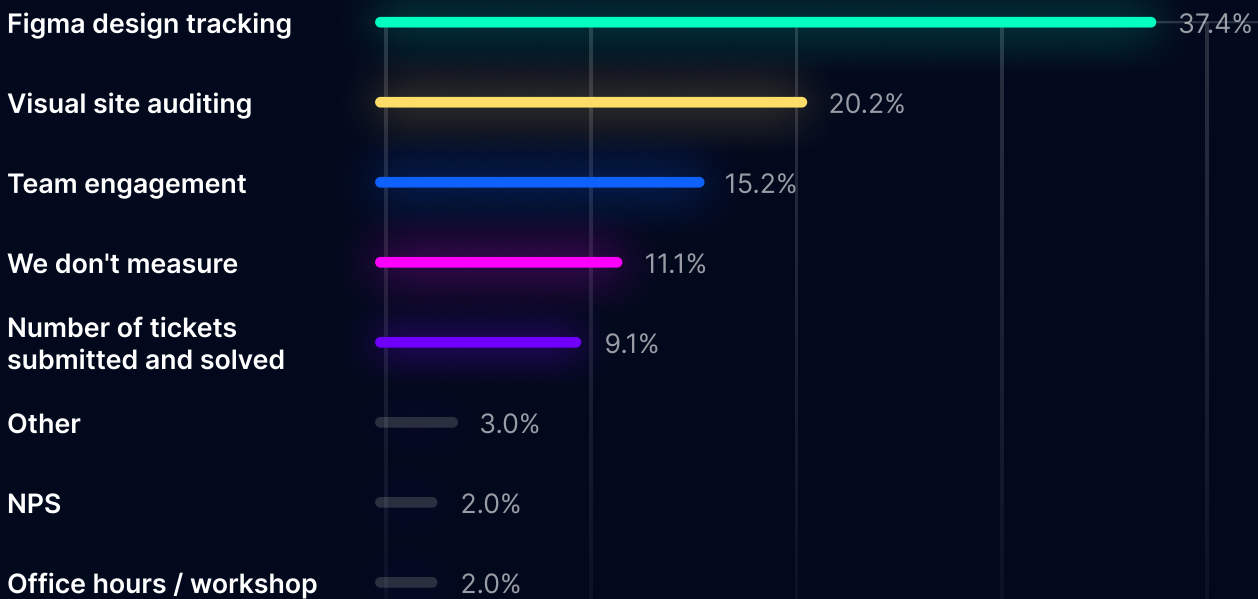
Metric	Value
● Adoption rate	52.2%
● Consistency	52.2%
● Efficiency	39.1%
● Performance	27.0%
● Cost savings	17.4%
● Time-to-market	14.8%
● We don't measure	10.4%
● Consumer satisfaction	0.9%

Company size distribution

100%

0%

2 – 49   50 – 249   250 – 999   1000 – 4 999   5 000 – 9 999   10 000+

Measuring adoption of components

Metrics for measuring adoption underscored the need for tangible data in evaluating the impact of components on the overall design process.

Although adoption rate is a key measure of the effectiveness of a component, the survey shows that how adoption is measured is inconsistent and largely dependent on the visual or design aspects, with little consideration for tracking component usage in code.

Metric	Percentage
Figma design tracking	37.4%
Visual site auditing	20.2%
Team engagement	15.2%
We don't measure	11.1%
Number of tickets submitted and solved	9.1%
Other	3.0%
NPS	2.0%
Office hours / workshop	2.0%

Drive Adoption With Supernova

Streamline your design processes and break down team silos with a truly connected design system using Supernova.

Learn more
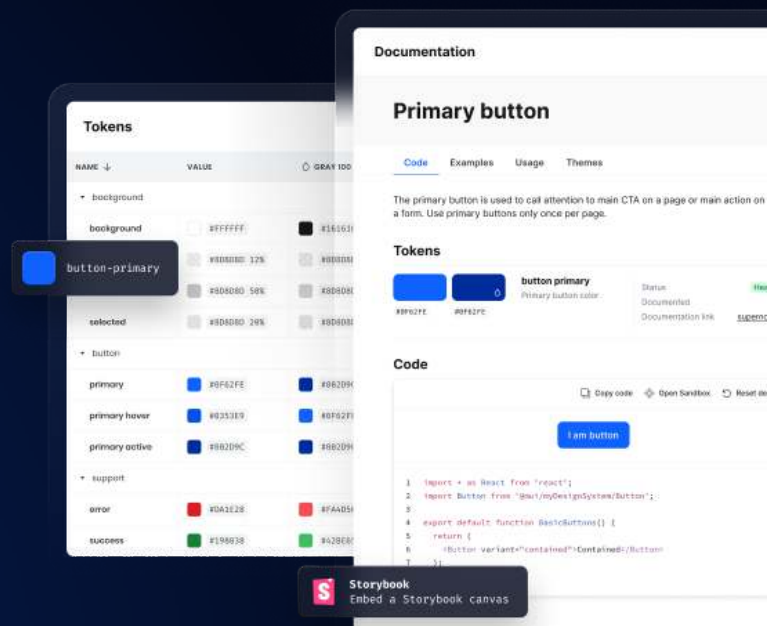
06

Governance & Maintenance

The final part of the report "Governance & Maintenance" looks at who owns components and how they're maintained. We also explore how and when teams version and deprecate components.

Who owns components in a design system?

Ownership of components vary. The majority of respondents (58.3%) say a dedicated design system team owns and maintains components, while 53.9% say designers primarily own the visual aspects of components and 47.8% of developers own the technical implementation of the components.

On the surface, ownership looks balanced across the board, but once we dig into the data, we see a stark difference in who owns components based on company size. 41.3% of companies with 250+ employees had a dedicated design system team that owned and maintained components, compared to 7.7% of companies with less than 250 employees. Smaller companies (43.1%) were more likely to have designers who primarily owned just the visual aspects of components.

In the "Other" responses, one respondent shared that designers own both the visual aspect and technical implementation of components. For another, the UX Product Manager owns components.

58.3% A dedicated design system team owns and maintains the components.

53.9% Designers primarily own the visual aspects of the components.

47.8% Developers own the technical implementation of the components.

13.9% Ownership is shared across cross-functional teams, designers, developers, and product managers.

3.0% Other

1.7% Ownership rotates periodically.

	7.7%	43.1%	32.3%	10.8%	4.6%
2-249 employees					

|	38.8%	27.5%	27.5%	5.6%
250+ employees				

Almost a quarter of respondents don't version components

Most respondents (48.7%) leverage a comprehensive approach and version components as a whole package. This implies that updates or changes to any part of the component set result in a collective version.

22.6% opt for a more modular approach, versioning components separately. This allows for more granular control over changes to individual components, streamlining updates.

Surprisingly, another 23.5% of participants don't version their components at all.

Want tips on how to version your components? Watch our panel "Components: Building the Building Blocks".

As a whole package	48.7%
As separate packages	22.6%
No versioning	23.5%
I don't know	8.7%
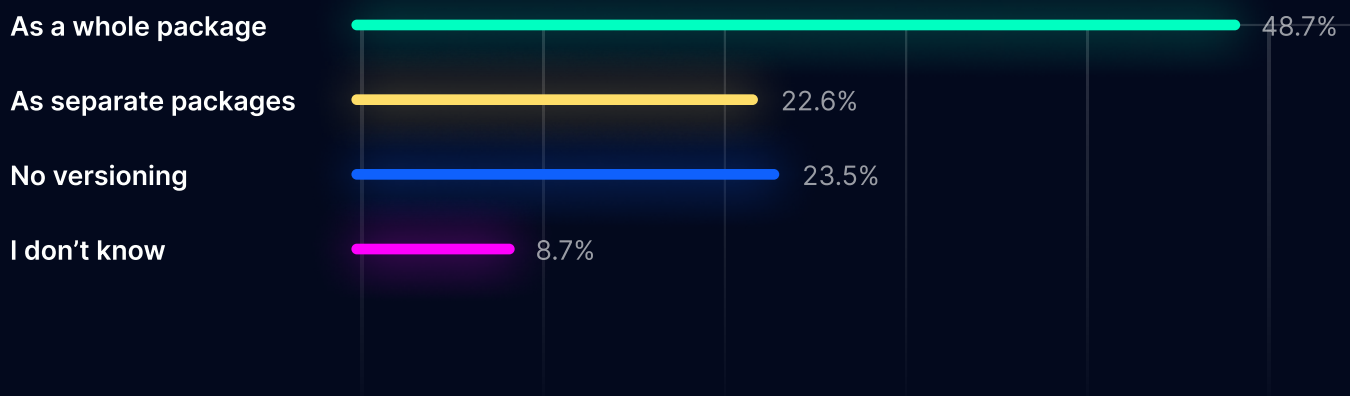
"We had our major updates which included breaking changes, which we did probably around twice a year. [We] tried to limit those because they're just very disruptive to teams and then we would do minor changes in between that wouldn't break the components."

Alissa Engstrom
Senior Product Designer, Design System / Apollo.io
Components: Building the Building Blocks
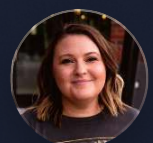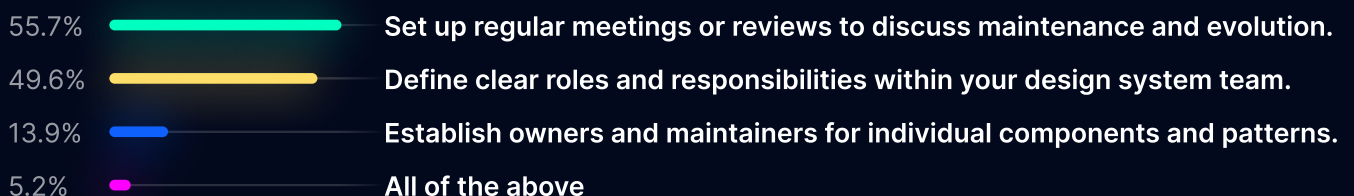
Common strategies to maintain components

Q: How do you maintain components in your library?

Most respondents (49.6%) maintain their component library by documenting changes and version updates for all components. Others (45.2%) regularly update and test components for functionality and accessibility. What surprised us is that some folks don't maintain their components. One participant said, "We never had to," and another shared that "Maintenance is low. Rarely revisit [components]."

49.6% ▬▬▬▬▬— Document changes and version updates for all components.

45.2% ▬▬▬▬— Regularly update and test components for functionality and accessibility.

22.6% ▬▬— Continuously monitor component performance and optimize for speed.
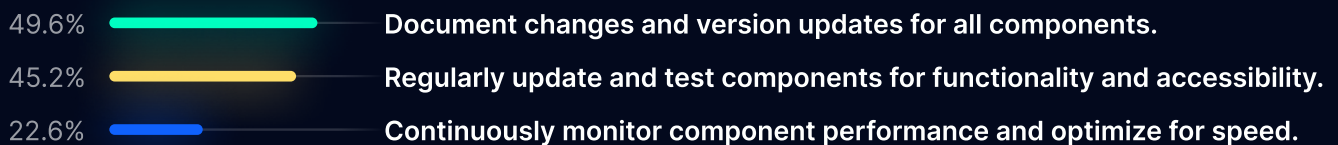
Q: How do you ensure collaboration for component maintenance?

For teams that do regularly maintain components, most teams (55.7%) set up regular meetings or reviews to discuss maintenance, and 49.6% also had clear roles and responsibilities defined for maintenance. Surprisingly, 13.9% had specific owners and maintainers for individual components and patterns instead of maintaining the component library as a whole.

55.7% ▬▬▬▬▬— Set up regular meetings or reviews to discuss maintenance and evolution.

49.6% ▬▬▬▬— Define clear roles and responsibilities within your design system team.

13.9% ▬— Establish owners and maintainers for individual components and patterns.

5.2% ▪— All of the above

"Single points of input are useful, we have just one repo where people can open tickets for all sorts of requests/issues. For communication it's probably a case of using literally whatever means are possible in your company, Slack or Teams channels, emails, mailing lists, good release notes, going to town halls — you can get really creative with your communications."

Henry Daggett
Design Systems Lead & Product Designer / Societe Generale
Beyond Launch: Nurturing & Sustaining Design Systems

When to deprecate components

Deciding to deprecate an existing component is a complicated process because teams need to look at how much it's used and how it affects the design system. They have to see if other patterns and features depend on it and how important it is to users.

It requires finding a balance between keeping up with new technology and making sure everything still works well together. During this process, everyone aims to make decisions that address why a new component is needed while making sure the change doesn't break any existing products. With this in mind, it's understandable that most respondents highlighted that similarity with other components (71.3%) is the most common reason for deprecating a component, followed by low adoption (64.3%).

73.1%
Similarity with other components

64.3%
Low adoption

53.0%
Maintenance challenges

33.9%
Ease of use

28.7%
Accessibility

Designer in DS team

Developer in DS team

Designer

Developer & Other

> "Sometimes it's just time for a refresh. Sometimes components aren't flexible. [You have to figure out] if what you have can scale and is going to continue to work or if it's easier to just go back to the drawing board and start from scratch."
>
> **Bethany Sonefeld**
> Design Manager / Duo Security
> Beyond Launch: Nurturing & Sustaining Design Systems

Components are the building blocks of design systems. They serve a key role in ensuring designers and developers can communicate and collaborate effectively.

The State of Components 2023 report discusses how designers and developers prioritize, document, measure, and maintain components. It also highlights opportunities for teams to take full advantage of their components.

Thank you for participating in the research study and reading the report! We hope the results help shape how you approach components moving forward.
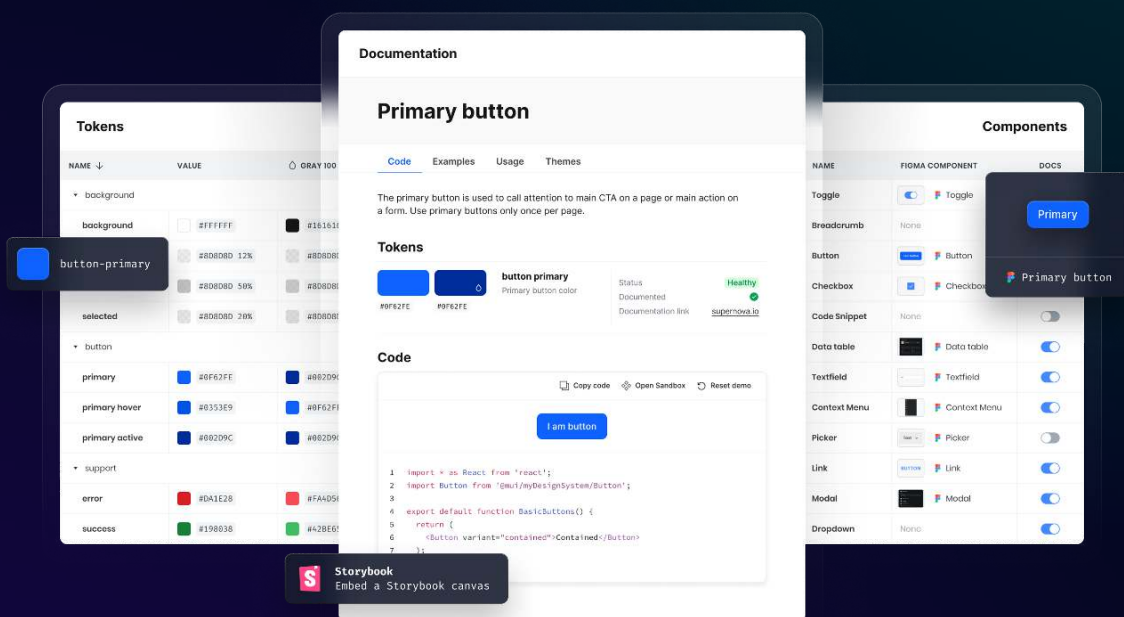
Unlock the potential of your components with Supernova

Track and document your components alongside your design tokens. Supernova connects design and code in one place so you can ensure design consistency and streamline development.

Sign up for free   Request a Demo