

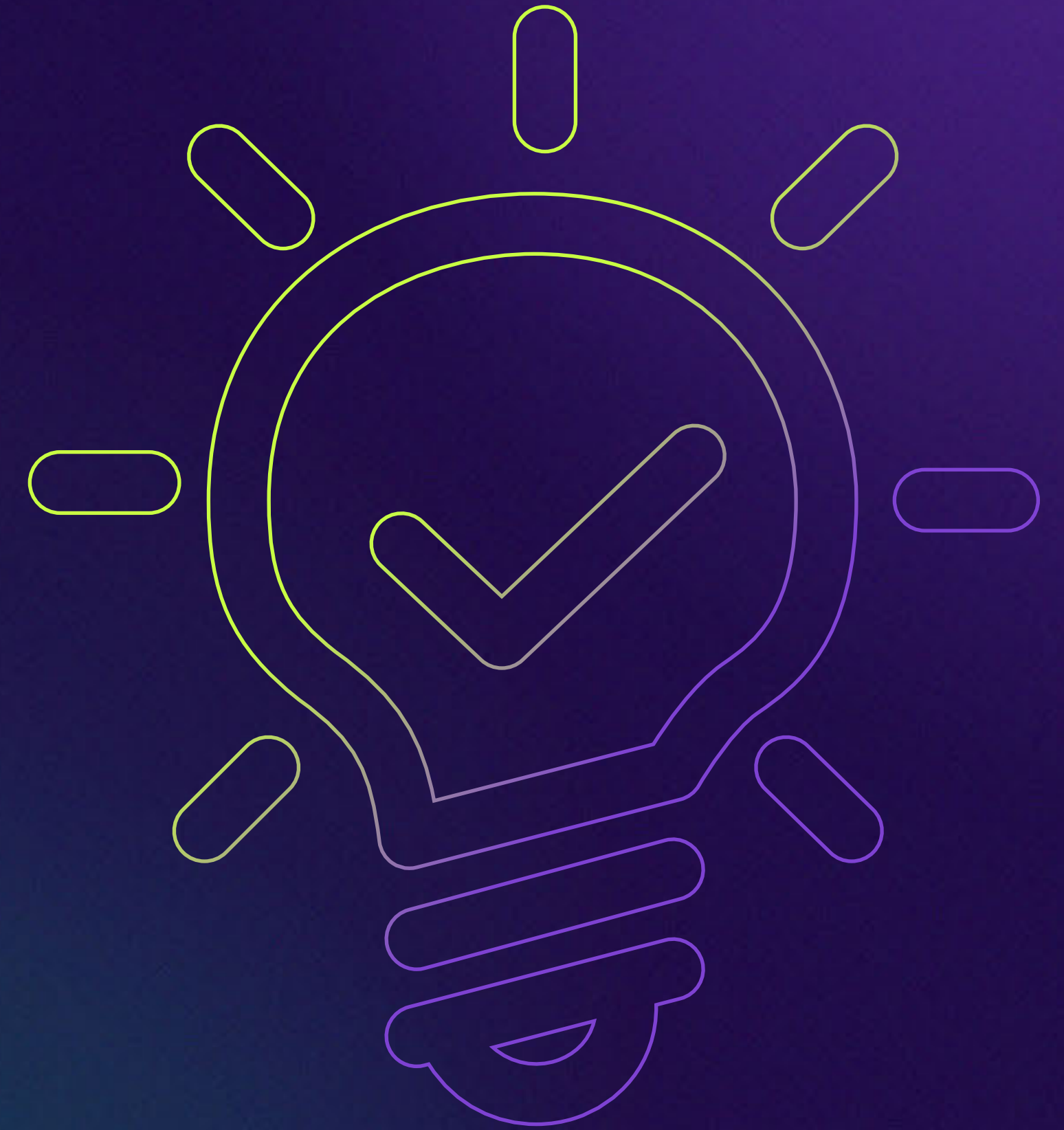
The State of
**Collaborative
Editing**

2024

Table of Contents

| | |
|----|--|
| 03 | Introduction |
| 08 | Chapter 1: The Collaborative Editing Landscape |
| 13 | Chapter 2: Collaboration Features and Experience |
| 22 | Chapter 3: The How and Why of Implementing RTEs |
| 30 | Chapter 4: The Future of Collaborative Editing |
| 36 | Conclusion |

Introduction



Introduction

The face of work took a seismic shift just four short years ago. Suddenly, teams that sat next to one another were miles apart. Coworkers who turned to see if a colleague was available at their desk were now pinging them in chat about their availability. Video meetings became the norm. And in-person collaboration became a memory.

Flash forward to today, and some have returned to the office, while many others have embraced distance work. Regardless of where you sit, however, the tools that allowed for both asynchronous and synchronous collaboration have come to be an expected part of our day-to-day workflows.

Collaboration seems like it should be an easy thing in the digital age. Cloud computing has made it simple to share files. Collaboration apps, like Slack and Microsoft Teams, have streamlined communication amongst geographically disparate groups (and even those in the same building). Video meetings mean that decisions can be made as a group with team members from across the globe. Experiences with these applications set the bar high for users who demand that all of their collaborative experiences work just as seamlessly.



It seems logical, then, that collaborative editing should also be easy, right? Yet, somehow, this key element of work hasn't completely caught up with the other mechanisms of collaboration.

There are great options out there for collaboration applications, but for some reason, some organizations are still choosing to email documents or “chair turn” from one editor to another, sharing files and working across multiple systems.

It's a tough nut to crack - anyone who has tried creating their own rich text editor (and our data says there are more than a few who have done so) has discovered that simply underlining words can be complicated. User-expected functionality, like automatically resolving editing conflicts and keeping track of who changed what is complex to develop and challenging to maintain.

The demand for collaborative functionality has influenced how developers create applications with core editing capabilities.

Giving users the best experience and the most focused functionality is more important than ever before and more complicated. Large organizations, like Microsoft and Google, inherently set the standards for this functionality by leveraging bigger teams (and bigger budgets) to bring collaboration features to life. Enabling document collaboration within an application is now table stakes, and application owners, developers, and technology leaders must consider how to enable working and editing together while looking ahead at what users will expect their apps to do next.

To better understand the demand for collaborative tools, CKEditor has once again commissioned its annual global survey of technologists, software engineers, product owners, and others to explore how companies view rich text editors and the collaboration functionality needed for successful applications and streamlined work.

We learned that on the surface, it appears that there is less of a focus on collaboration functionality. But that's not because

it's any less important than it was just a year ago. Rather, the ability to collaborate in an editor has simply become expected, not a capability that's nice to have. In fact, these features are so important that those not planning to include them in their applications could soon be left behind.

That's because the benefits of collaboration have become clear. **Teams with seamless collaborative editing capabilities are more efficient, have increased meaningful interactions and communication, remove blocks and bottlenecks, and can work in real-time to share ideas.** Collaborative content creation and editing reduces confusion and accelerates teamwork in ways that flipping back and forth between different editors and storage mediums simply can't match. As one respondent said:

"For the projects I work on, the ability to be able to write text in a collaborative manner is an important aspect. It is [better] to have it within our projects than to rely on third-party platforms such as Google Docs, so for this reason, it is very important that collaboration tools are integrated within the editor we use."

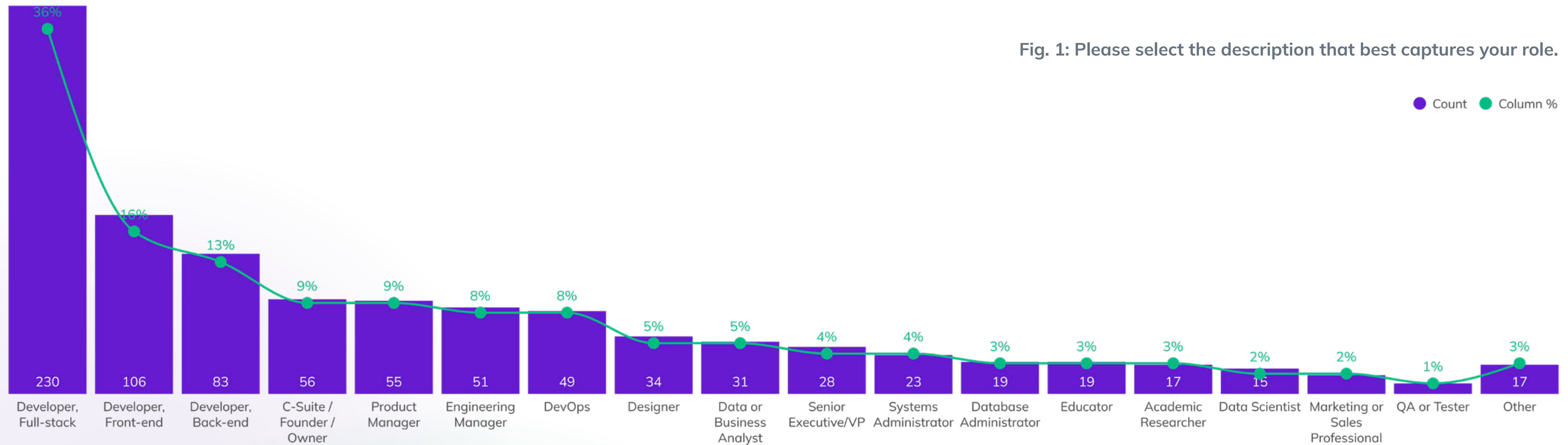
In the following pages, you'll learn, as we did, what today's application creators find important when bringing collaboration to their users, and how users feel about these critical tools, as well. You'll get to see how companies, from large enterprises to the smallest, scrappiest startups, are using collaborative features in their own applications. Plus, you'll get a glimpse at what the developers and product owners at the tip of the spear think the future holds for the world of collaborative editing.



Who we surveyed

CKSource, a Tiugo Technologies company, conducted this survey in the first quarter of 2024 in partnership with DO FEEL THINK. The sample size included 646 technical professionals who develop or manage rich text editors in their applications today, including 270 power users who seek a wide range of advanced features.

The survey encompassed a broad range of technologists using RTEs within their own solutions across a wide range of products and custom-built editor platforms and included free, open source, and commercial software usage.



Chapter 1:

The Collaborative Editing Landscape



The importance of rich text editors

Modern applications are about getting things done and getting them done fast. For every organization, communication is critical to both speed and efficiency, and content plays an important role as well.

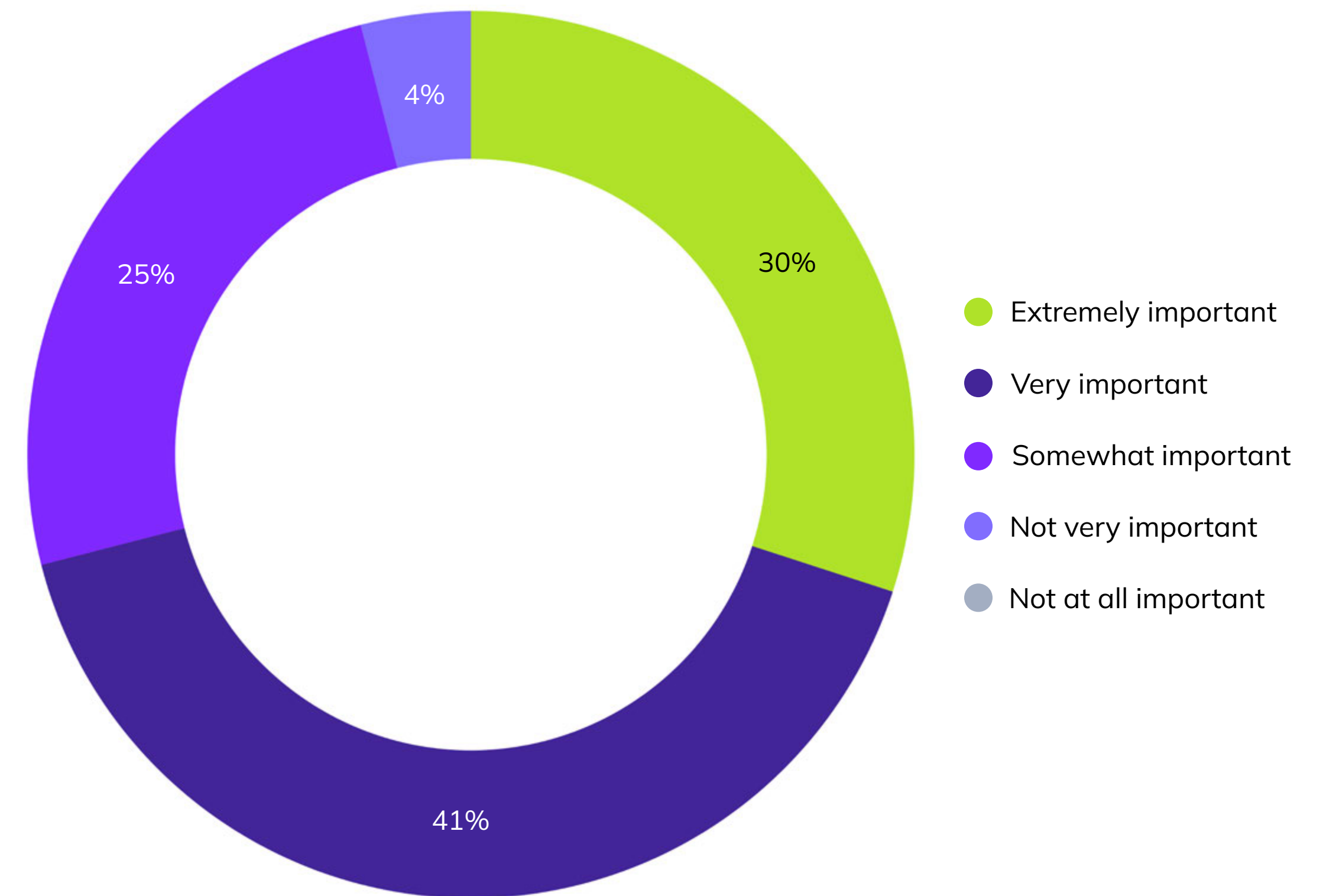
It's no surprise, then, that 71% of those surveyed reported that a rich text editor (RTE) is either extremely or very important to their application, product, or platform.

The mere presence of an RTE isn't enough, though. The potential uses are broad, driving the need for flexibility and features.

More importantly, those features are expected to be straightforward and easily understood by the app's end users while still feeling seamless within the context of the product.

Adoption of an application is key. Lacking the right tools, offering incomplete toolsets, or including features with a high learning curve can create friction and hinder user acceptance and satisfaction. Collaboration is one of those critical features that users expect across many contexts.

Fig. 2: How important is rich text editing to your platform, product, or application?



The role of collaboration in RTEs

Collaboration within an application's rich text editor ranges from a nice feature to a key component. While not every product has a need for things like co-editing, track changes, or at-mentions, only 13% of respondents felt collaboration features weren't important at all. In contrast, nearly half rated collaborative functionality as extremely or very important for their application.

Those creating or managing these applications recognize their user's demand for collaboration features, as well. More than 40% of respondents identified significant or considerable interest from their user base for collaboration tools within their product.

Fig. 3: For your application, how important are collaboration tools and features?

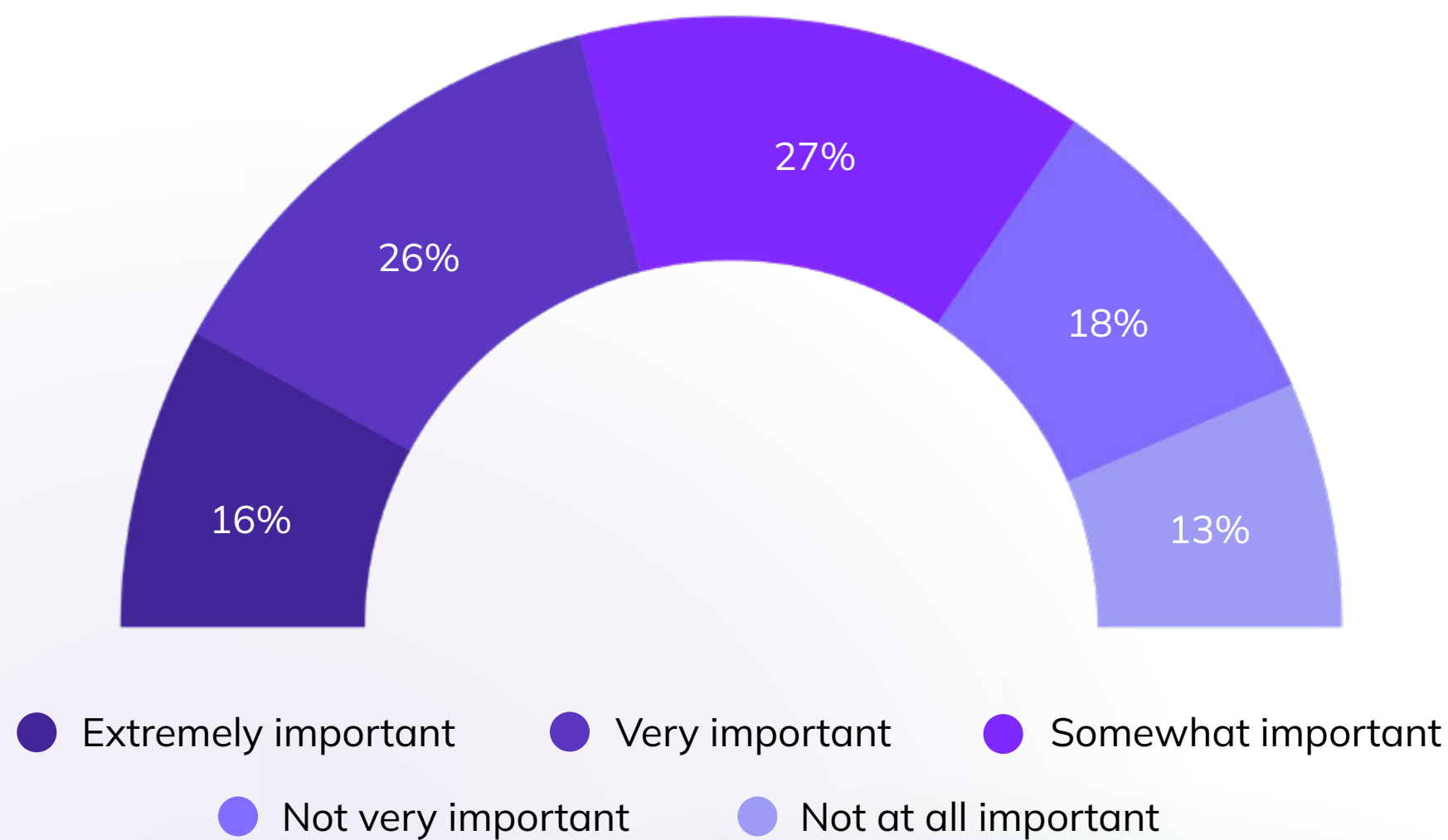
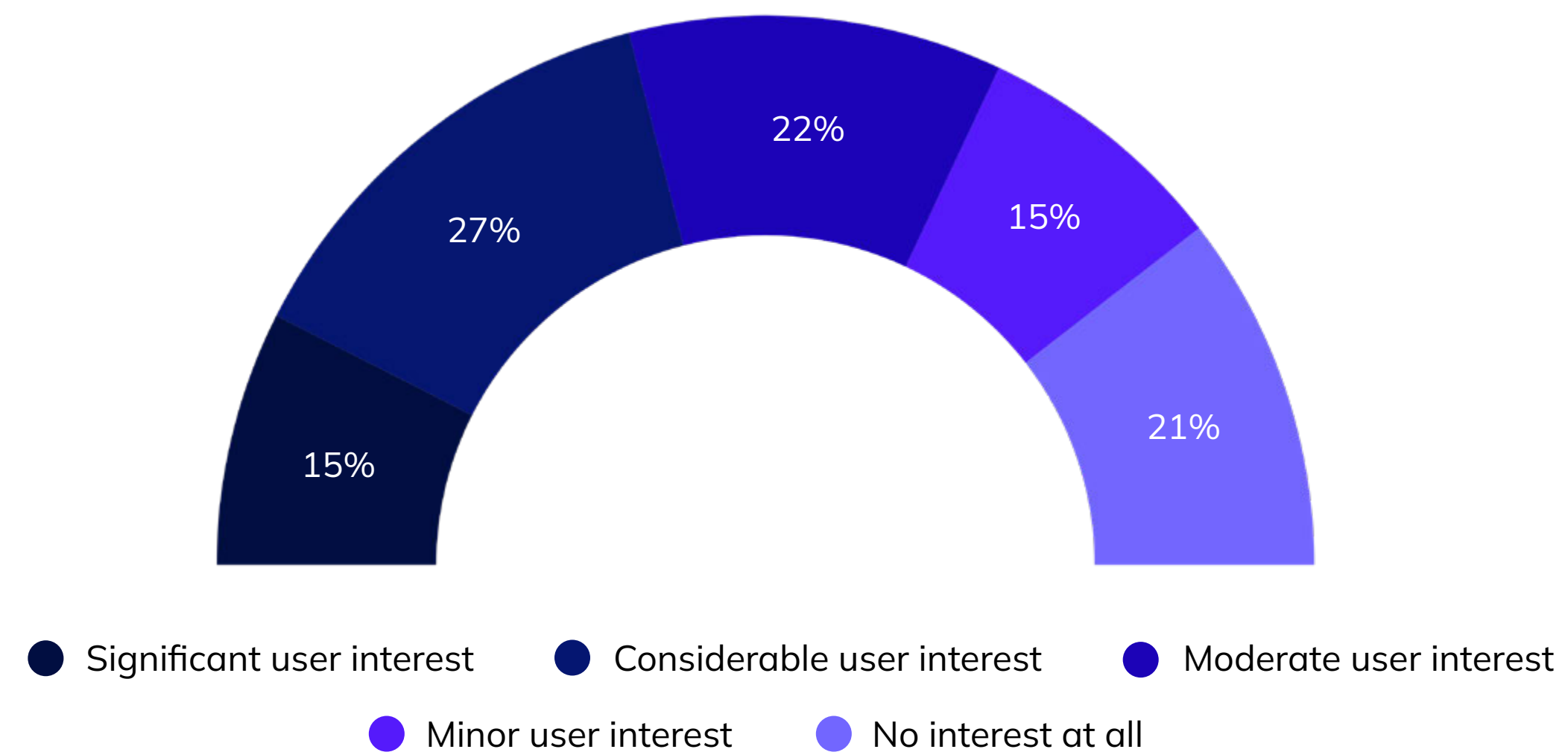


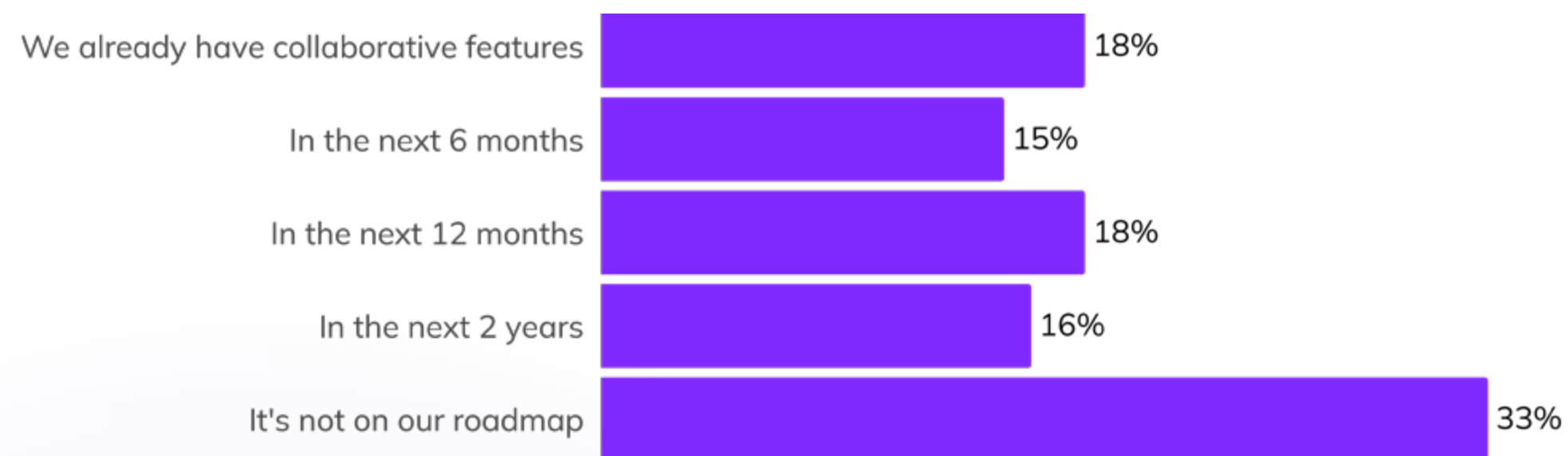
Fig. 4: How much interest are you seeing from users when it comes to collaborative editing features?



The implementation imperative of collaborative tools

User demand and the acknowledgment of the importance of collaboration within applications has driven many developers, product owners, and technology leaders to offer asynchronous or even real-time collaboration.

Fig. 5: At what point in the future do you see the need to integrate collaborative features?



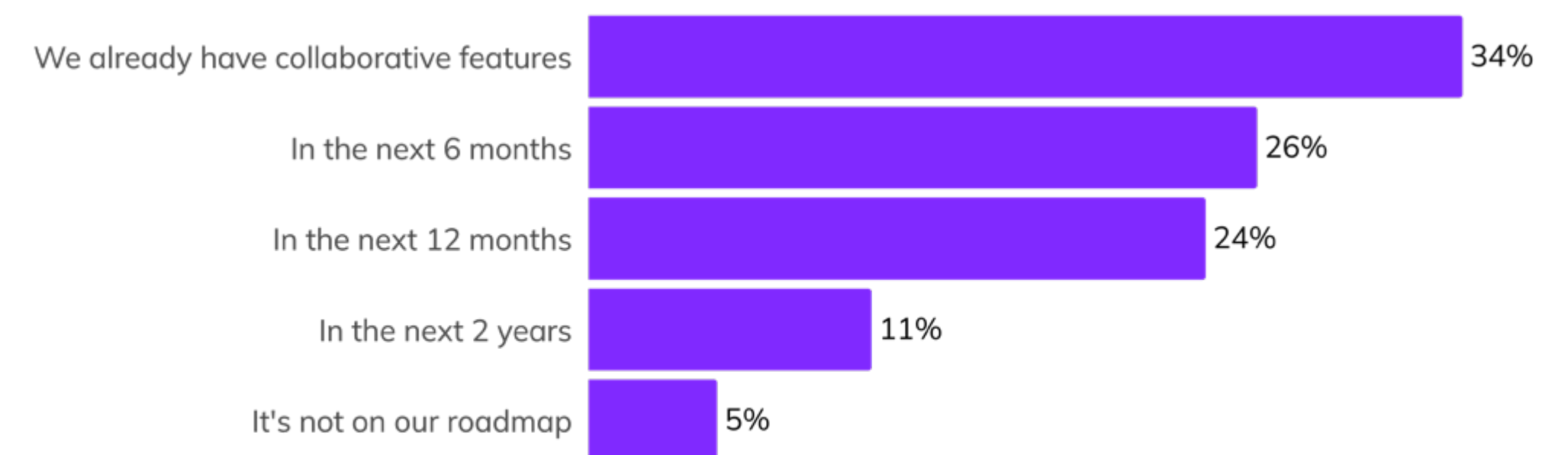
Of course, not every application has a use for collaborative editing. As we saw in Fig. 3, 13% didn't feel that collaboration tools and features were necessary, and 27% found them only somewhat important.

Among the remaining 42% who placed high importance on collaboration features, the data takes a stark shift toward urgency. More than a third of those that consider these

tools very or extremely important have already implemented them. The remaining 60% have these tools on their roadmap within the next two years, with nearly half ready to implement collaboration features in the next 12 months.

The takeaway? If your users want the ability to collaborate, the time to take action is now. Otherwise, your app could be lagging behind your competitors, impacting users' productivity and ability to comply with process and compliance guidelines, and causing frustration for them and for management.

Fig. 6: Respondents who rated collaboration as important - At what point in the future do you see the need to integrate collaborative features?



Collaboration expectations

An RTE can fill the need for a variety of applications with a multitude of requirements. As such, developers and product owners can differ in their expectations of the features and functionality within a rich text editor.

Similarly, not every feature is expected to be included as part of an RTE. For instance, respondents were asked to identify features they would find surprising for an editor to include, with things like an AI Assistant (63%), document import (41%), and markdown support (33%) being among those that were unexpected.

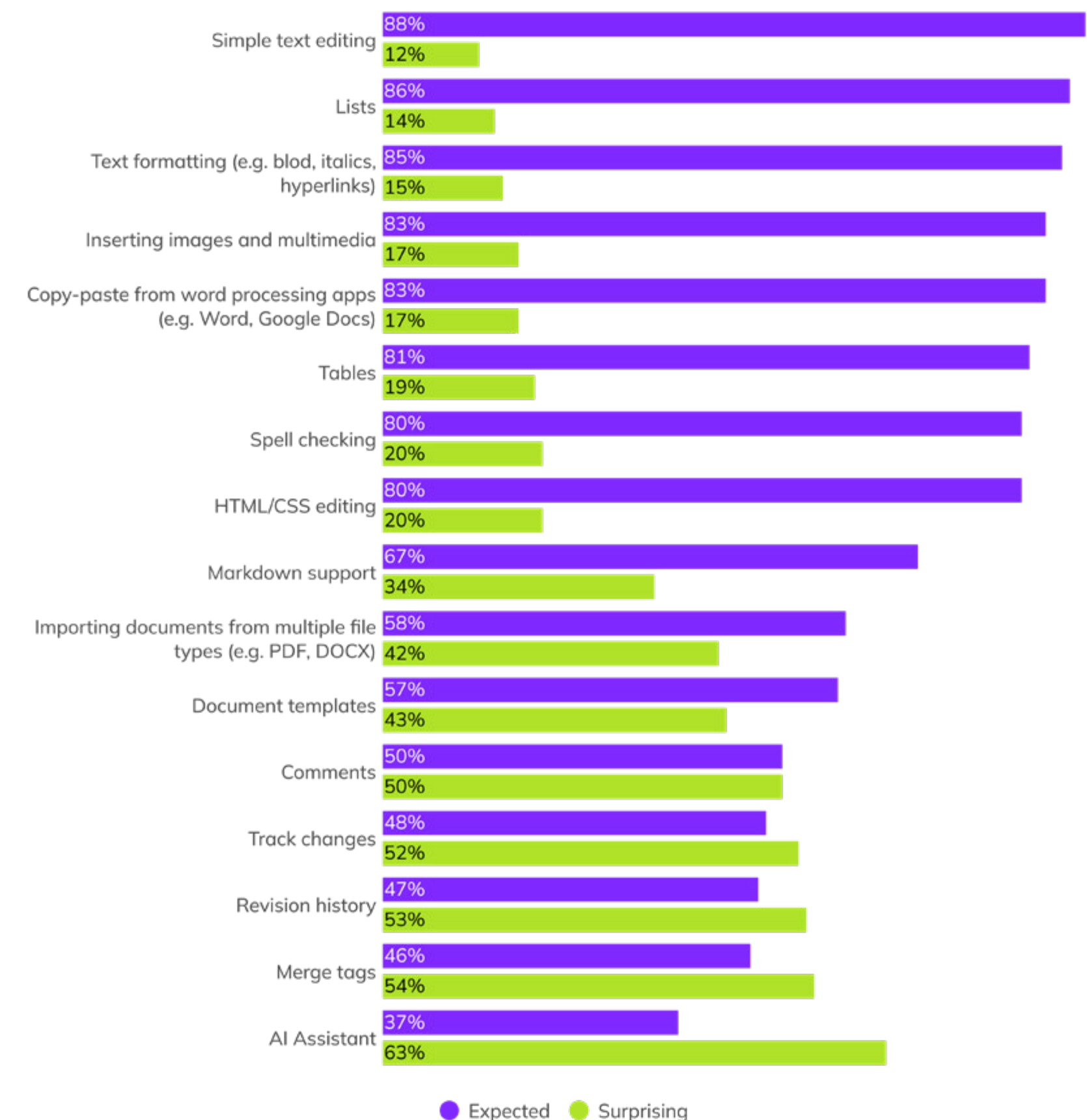
Compared to the results from our 2023 State of Collaborative Editing report, collaboration tools appear to have dropped in importance. Only 42% of respondents reported these features as highly important compared with 60% last year.

And yet, when contrasted with the number of app creators identifying collaborative features as being crucial, this might feel like cognitive dissonance. It's not. More likely, it's a hat tip to the ubiquitousness of collaborative tools in today's products.

Elements like comments, track changes, and revision history were split nearly 50-50 between being expected and surprising. It appears that collaboration tools have become table stakes for developers and users alike.

This raises the question: What features are defined as collaborative enablers?

Fig. 7. Which features would you expect to find in a rich text editor, and which would you find surprising?



Chapter 2:

Collaboration Features and Experience



Solving the challenges of collaboration

Clearly, the data points to the importance of collaboration for a majority of application developers with rich text editors in their apps.

You can't just wave a magic wand and have collaboration. Product owners and technology leaders recognize that enabling collaborative content creation and editing poses any number of challenges.

If not well executed, users can experience confusion with features or even how to collaborate with others, which hinders productivity instead of supporting it.

This can be further exacerbated by the number of tools that need to be used to achieve collaboration. If there is a great deal of "chair turning" - such as writing in one application, uploading the document to another share, then consolidating feedback and edits - the likelihood of frustration and potential mistakes soars.

This type of workflow can also challenge the integrity of information and the data within the content. Emailing files for review without some kind of governance can lead to details being altered or merges combining

incompatible changes. Context can be lost, and workflows can be stagnated by bottlenecks.

Certain challenges can surface based on industry, too. Regulated industries may require changes be tagged with user identification. These companies may also have compliance or governance rules around information retention.

A rich text editor that allows for collaboration in a single application solves for the complications created by multiple, disconnected tools. The key is understanding what's most important and what the benefits are to the organization and to the users.

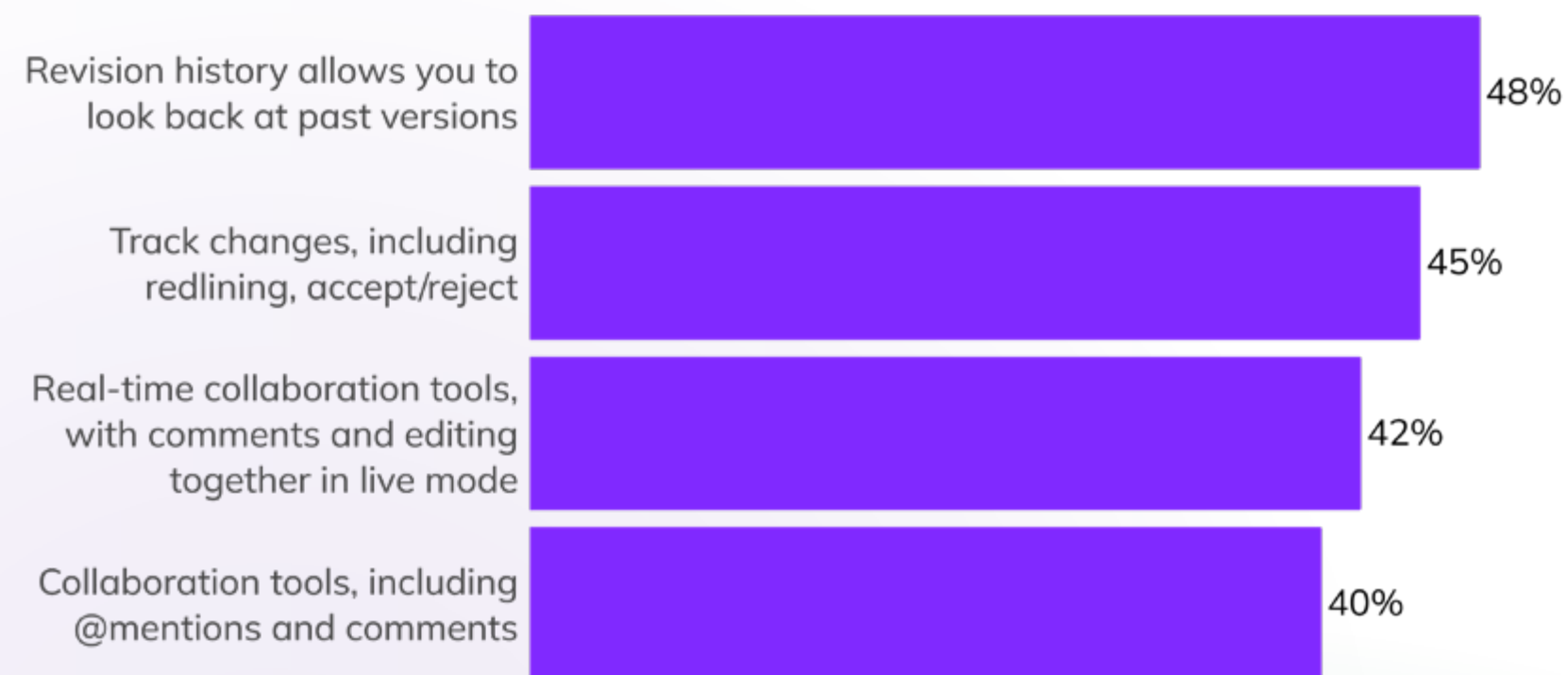
In this year's survey we asked about the collaboration features that were considered key for applications. We also dug in to understand which industries needed which collaboration features the most, and heard directly from respondents on the reasons that overcoming the challenges of collaboration was worth the time and effort.

Rating collaborative tools

Not all collaborative tools are created equal. Some are specialized. Others have broad applications and appeal.

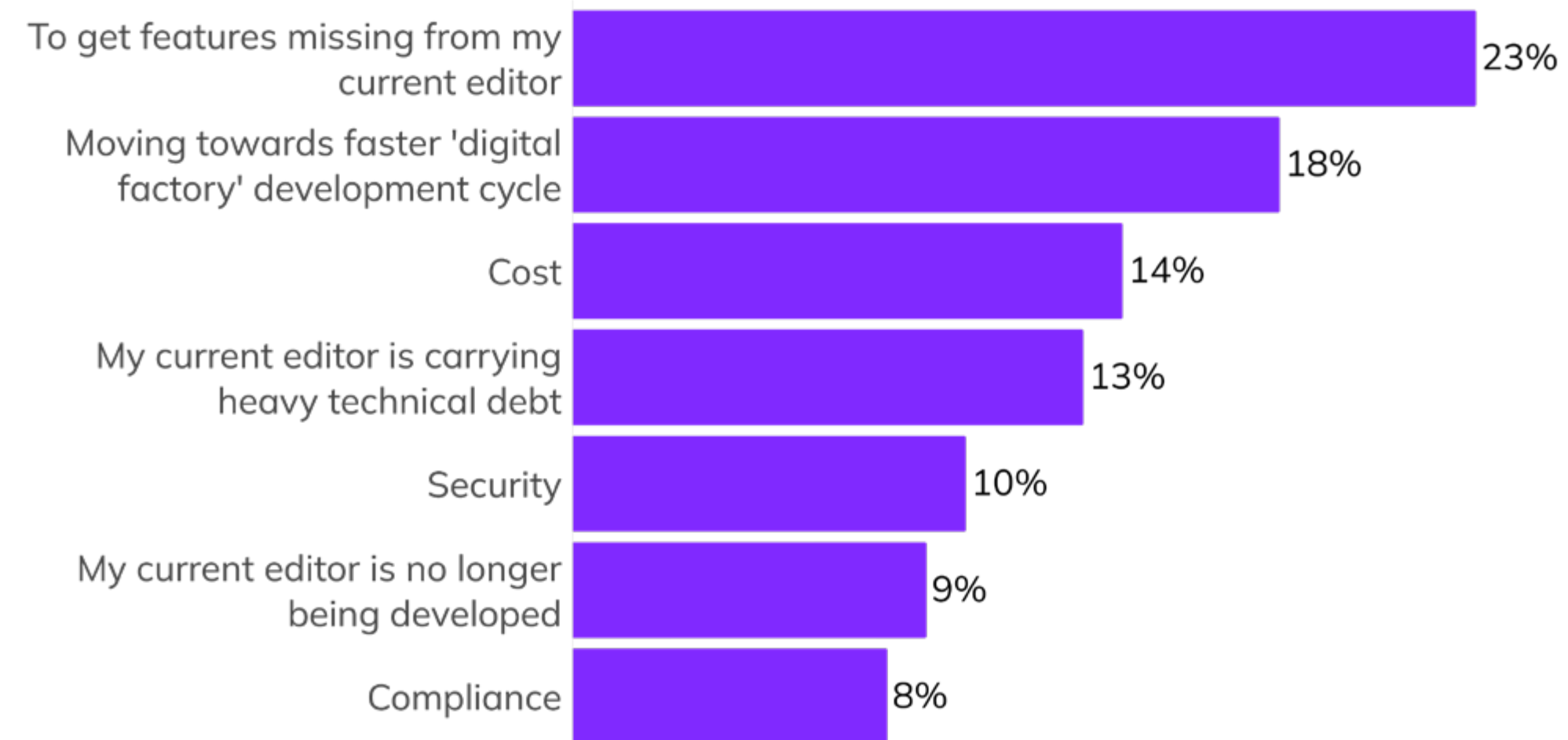
According to those who took part in this year's survey, the most important collaborative tool is revision history, with 48% ranking it in the top two highest categories. The least important? Tools like mentions and comments, with real-time collaboration following closely behind. But even those were considered important by 42% and 40% of respondents, respectively.

Fig. 8: How important are each of the following features for your organization?



When these tools do matter, though, they **really** matter. Forty-three percent of respondents said they would likely consider switching RTE providers in the next two years. The biggest reason to switch? Nearly a quarter indicated that they would change rich text editors to get features that are missing from their current editor - nearly double the rate that would switch because of cost.

Fig. 9: What is the primary reason you are likely to switch rich text editors in the next two years?



Feature expectations by company size

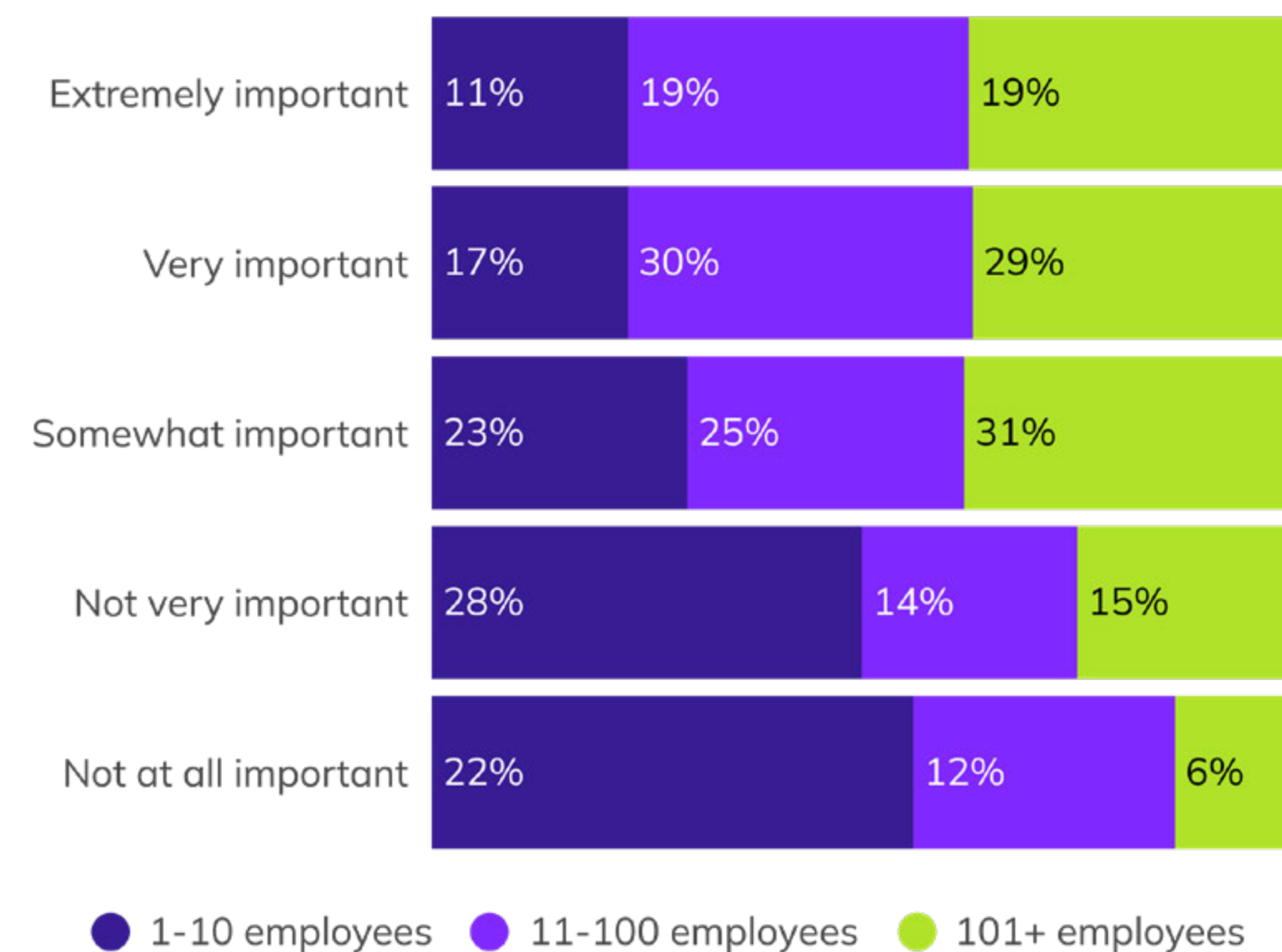
While there may be broad agreement on the importance of rich text editors across industries and organization sizes, the reasons and requirements of implementation vary. Companies of different sizes have different priorities. The focus on features and deployment methods shows some variance across organization size and type of development work.

Medium-sized and larger companies tend to regard collaboration features as more important than smaller organizations, with 49% of companies with 11 to 100 employees ranking collaboration tools as extremely or very important, and 48% of companies with more than 100 employees doing so. Only 28% of companies with 10 or fewer employees placed collaboration high on their list, with an equal percentage feeling they weren't important at all.

This could be because larger companies have more people trying to work together on a team or multiple teams across an organization attempting to collaborate on a single project or document. In organizations with fewer than 10 people there may only be a single resource working on a document, or processes within the organization may not be mature enough to support true collaboration. Additionally, some survey

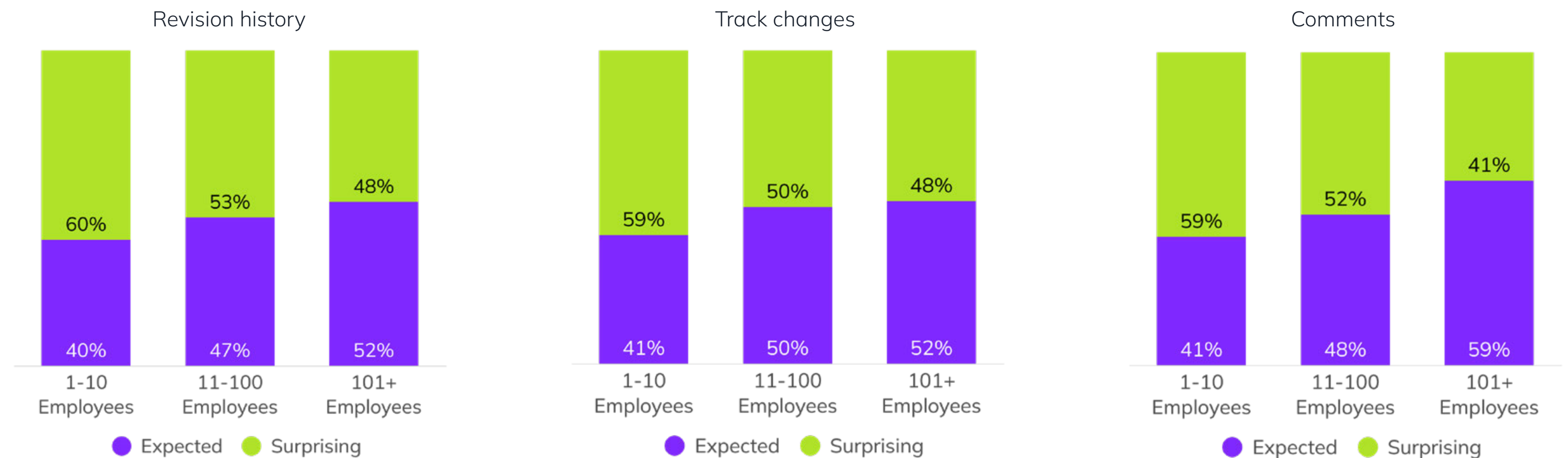
respondents may be working solo on projects that incorporate rich text editors and therefore do not value collaboration capabilities as much as those working with others.

Fig. 10: Data by company size - For your application, how important are collaboration tools a features?



Mid-sized to larger companies also have greater expectations when it comes to the collaboration features available in their rich text editor. Roughly 60% of small companies thought features like track changes, comments, and revision history would be surprising to find in their RTE. Conversely, 52% of respondents from large companies expected track changes and revision history to be in their editor, and nearly 60% expected comments to be available.

Fig. 11: Data by company size - Which features would you expect to find in a rich text editor, and which would you find surprising?

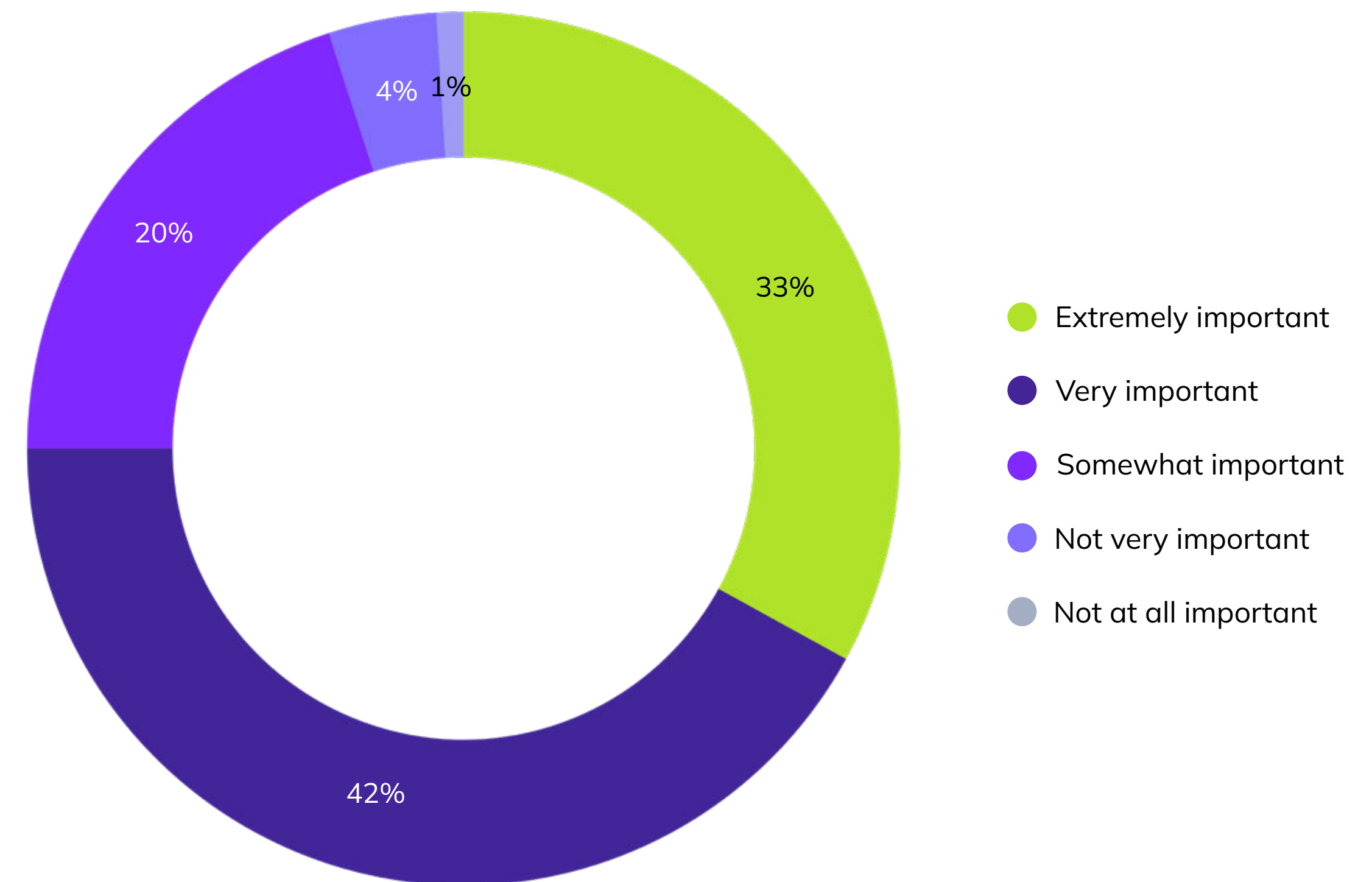


Special look: Collaboration tools in regulated industries

Users in regulated industries - such as finance, healthcare, and government - have special compliance requirements when it comes to recording changes, retaining data, and identifying the users who have made changes.

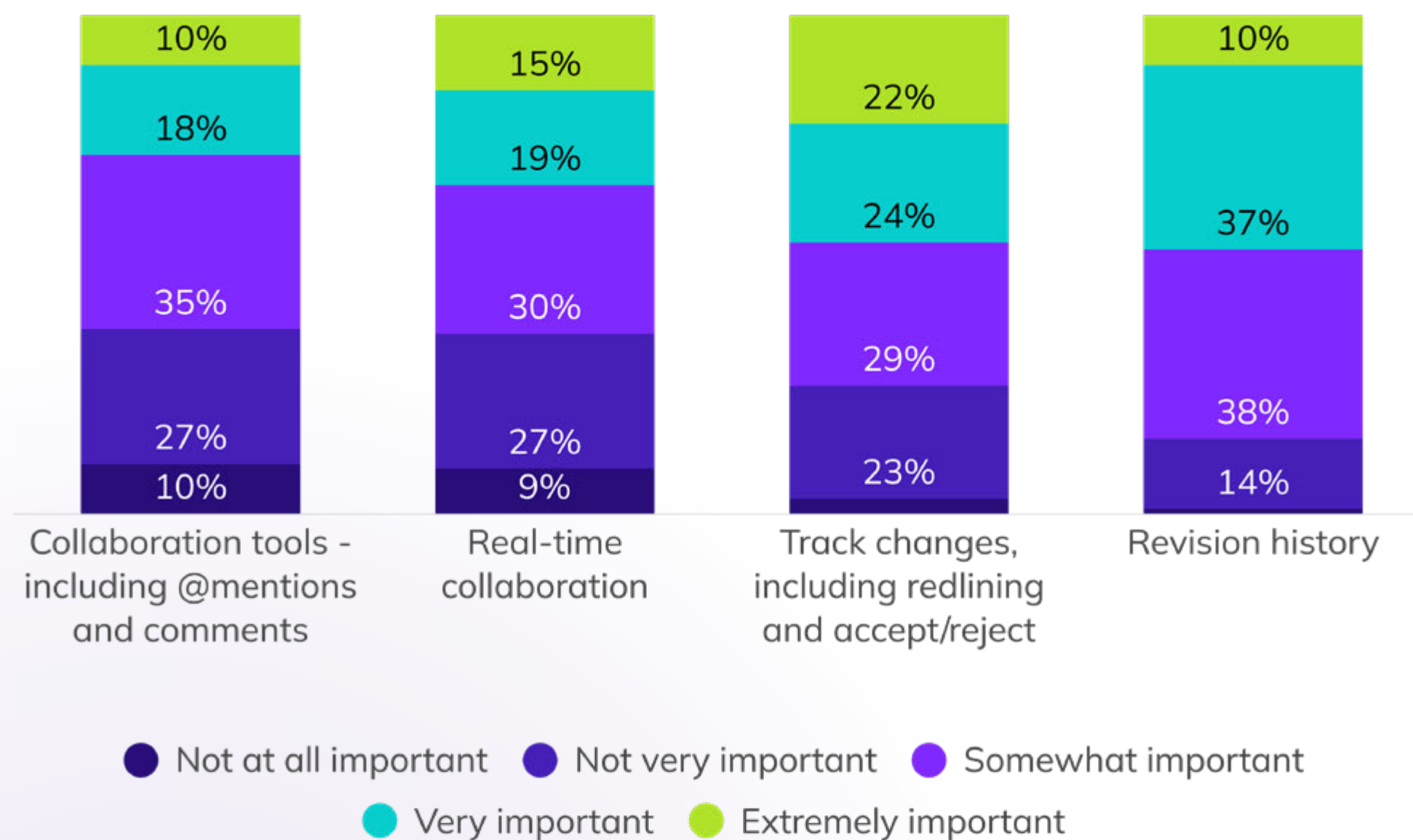
The most important features for survey respondents in these industries was, unsurprisingly, security and privacy protections, with 74% identifying security as being of top importance.

Fig. 12: Data by regulated industry - How important are security and privacy for your organization?



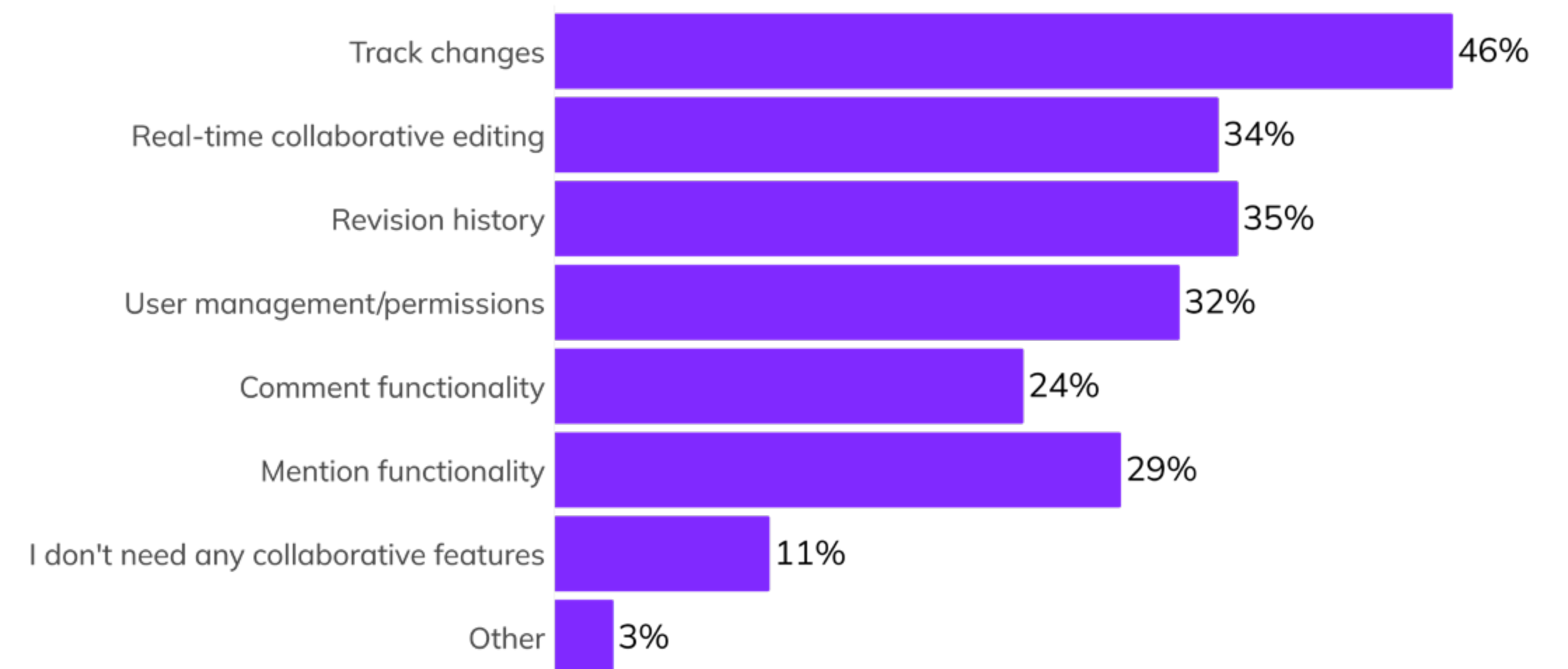
Collaboration features weren't recognized for being as critical, with only 27% identifying comments and mentions as highly important.

Fig. 13: Respondents in regulated industries - How important are each of the following features for your organization?



Looking to the future, these respondents place significance on things like track changes (46%) and real-time collaborative editing (34%) in regards to collaboration, which mirrors their current importance for collaborative features. These tools may not top the list, but the need for these features looks to remain stable in the years to come. Those looking for an RTE in these industries should be evaluating ones that offer security and privacy while also meeting users' collaboration needs.

Fig. 14: Respondents in regulated industries - In the next 5 years, which of the following collaborative editing features will be critical?



Collaboration Benefits: In their words

The benefits of collaboration within applications goes far beyond enabling team members who work from geographically disparate locations. Even for teams located in the same office, in-app collaborative editing offers significant value.

For example, features like track changes and revision history provide context across the life of a document. It's clear from the survey results that respondents recognize the value of collaborative tools and features. But what value, specifically, do they see?

Accelerates teamwork and improves efficiency

Collaboration tools, like comments and track changes, can streamline the work between team members and even across teams while adding clarity and eliminating confusion.

"(It) Makes communication between different team members smoother and reduces misunderstandings and conflicts."

Because users can work together, content creation, review, and publication can happen faster and with a lot less confusion.

"(Collaboration) allows for a smooth workflow that includes efficiently and actively producing content on a large scale."

Real-time editing

"I work in an industry that is document based. Real-time collaboration is key to success."

Having multiple stakeholders interacting with the same document at the same time can be a game changer. While it can be a significant engineering feat to accomplish well, rich text editors that support synchronous editing and creation satisfy user needs and create opportunities for in-the-moment inspiration and interaction.

"It's what people expect, and it solves the revision collision problem set."

Encourages accountability

Clarity goes beyond seeing what someone else has changed within a document. With features like mentions, revision history, and asynchronous editing, teams can share the workload, helping scale content development while supporting transparency.

“Collaboration tools can help teams better plan and assign tasks to ensure that work tasks are completed effectively.”

“Users can assign tasks within the document, making it clear who is responsible for specific sections or edits, fostering accountability.”

Having a full-featured RTE that can provide collaboration tools supports these benefits. It brings the facets of co-authoring and co-editing into a single application.



Chapter 3:

The How and Why of Implementing RTEs



As we'll see in the following pages, a lot goes into the why and how of including a rich text editor and collaborative features into an application. Choices like what to implement, where, and how are influenced by a variety of factors, from company size to application needs and user expectations.

Project size, resource availability, and requirements can all drive which editor is chosen for an application as well as where and how it's deployed. For those that decide to adopt a third party tool, the choice can be driven by very similar criteria, including scalability and cost, but reputation, support, and ease of implementation also play factors in the decision.

A majority of companies choose to buy an RTE to integrate into their apps, but around 23% of those surveyed chose to build their own. A significant portion of the companies who choose the build path still place high importance on collaboration features, despite the complexity of implementing things like real-time collaboration, mentions, and revision history.

Where a project's RTE is deployed can be as complex of a decision as what features it provides. Scalability, maintenance, and company policy all play a role in deployment decisions, while cost, feature availability, and the availability of internal resources can drive the decision to build or buy an editor.

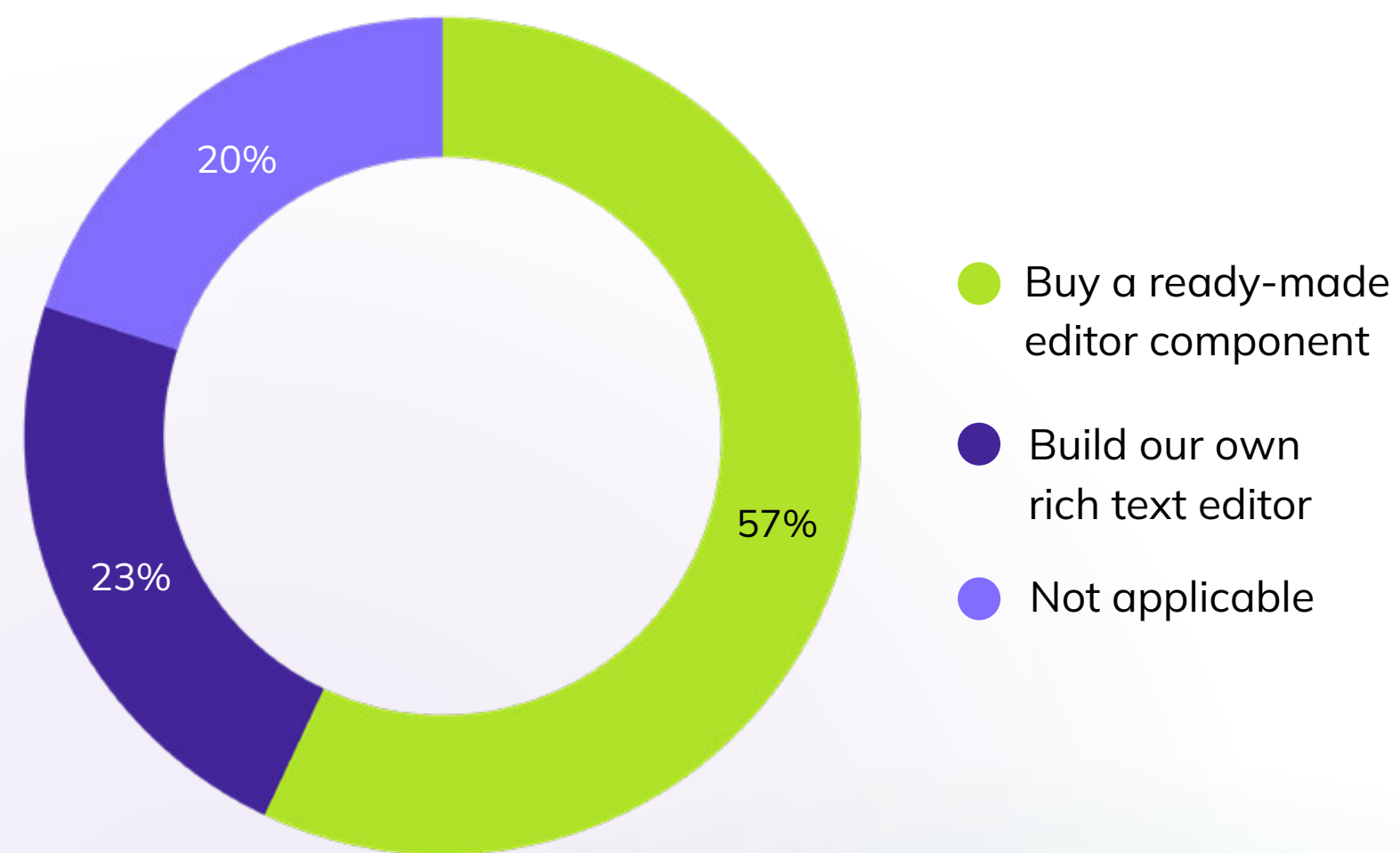


Buy? Or Build?

For any project, the buy vs. build question is a complex one that must take factors like time, available resources, cost, and technical debt into consideration.

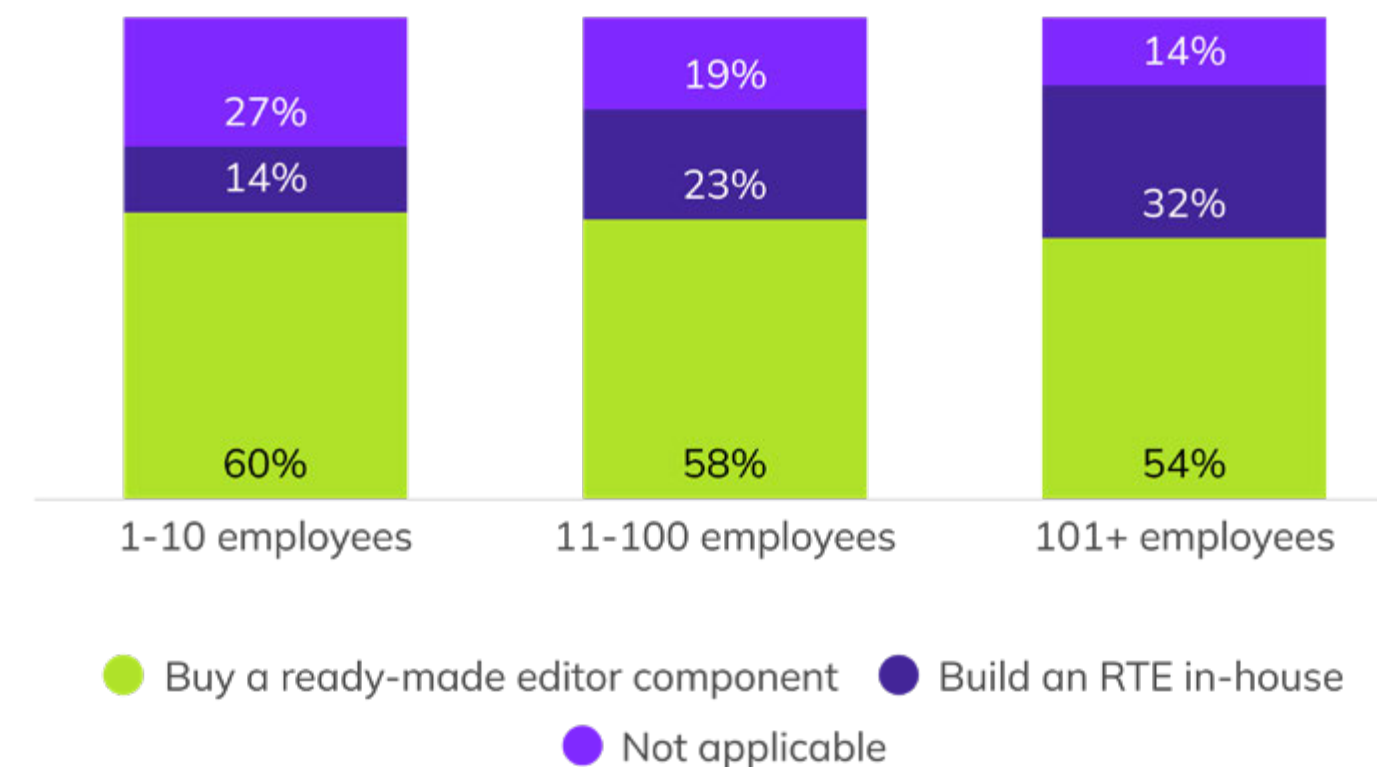
Of those who participated in our survey, 57% relied on a third-party RTE. Twenty-three percent of respondents built their own editor in-house instead of working with a ready-made component.

Fig. 15: For work projects that require a rich text editor, do you usually buy a ready-made editor component or build your own in-house?



Those who chose to implement a ready-made RTE into their application skewed slightly toward smaller companies with 1-10 employees, with 60% using a ready-made editor component. Choosing a third-party solution would allow smaller organizations to include a complex element like an RTE within their application without needing to create or maintain it themselves. The data for larger organizations, however, was similar, suggesting that the value of incorporating a ready-made component is recognized regardless of organization size.

Fig. 16: Data by company size - For work projects that require a rich text editor, do you usually buy a ready-made editor component or build your own in-house?



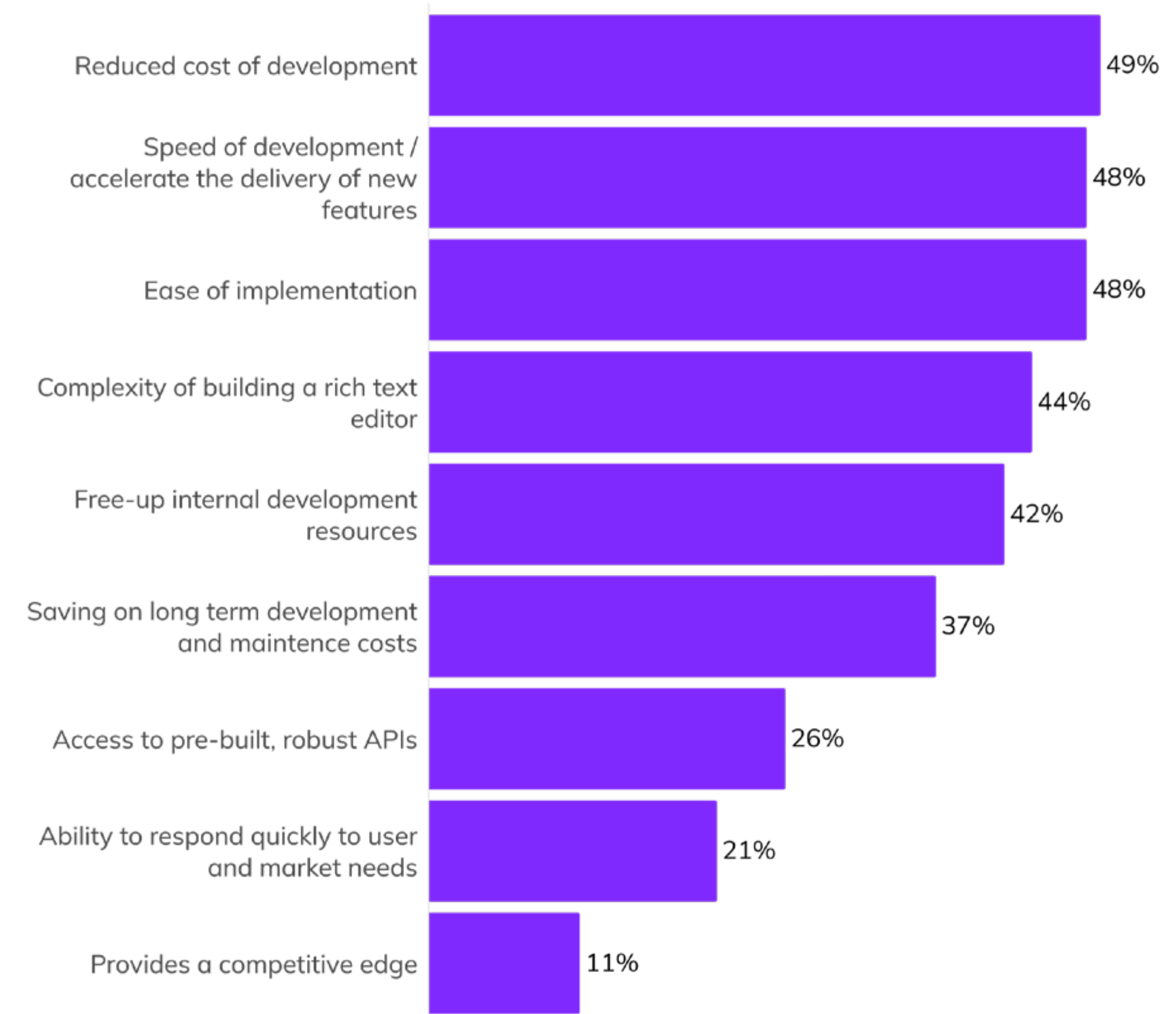
The choice to buy

The choice to use a third-party tool spans a wide range of reasons. The question “What drives your choice to use a ready-made editor instead of building your own?” allowed respondents to select multiple decision influencers in our survey.

Cost, speed, and simplicity appeared as the top three biggest factors in the buying decision. Reducing the cost of development was selected as being the biggest driver, being chosen by 49% of respondents. Forty-eight percent cited a third-party app helps them accelerate the delivery of new features and applications, while the same percentage noted that a ready-made application is easier to implement.

From our experience, a custom rich text editor with moderate features could easily take a 4 person team a year to develop, with an ongoing commitment of 10% of the team’s time for upgrades, enhancements, and support. The estimated cost? Roughly \$800,000, with ongoing costs of about \$80,000 a year.

Fig. 17: What drives your choice to use a ready-made editor instead of building your own?



Buying an RTE component means evaluating and choosing the right product. Overwhelmingly, respondents said that ease of implementation was the thing they most valued when selecting a third-party component (29%). Affordability was the next closest consideration (15%), with scalability following closely behind (12%).

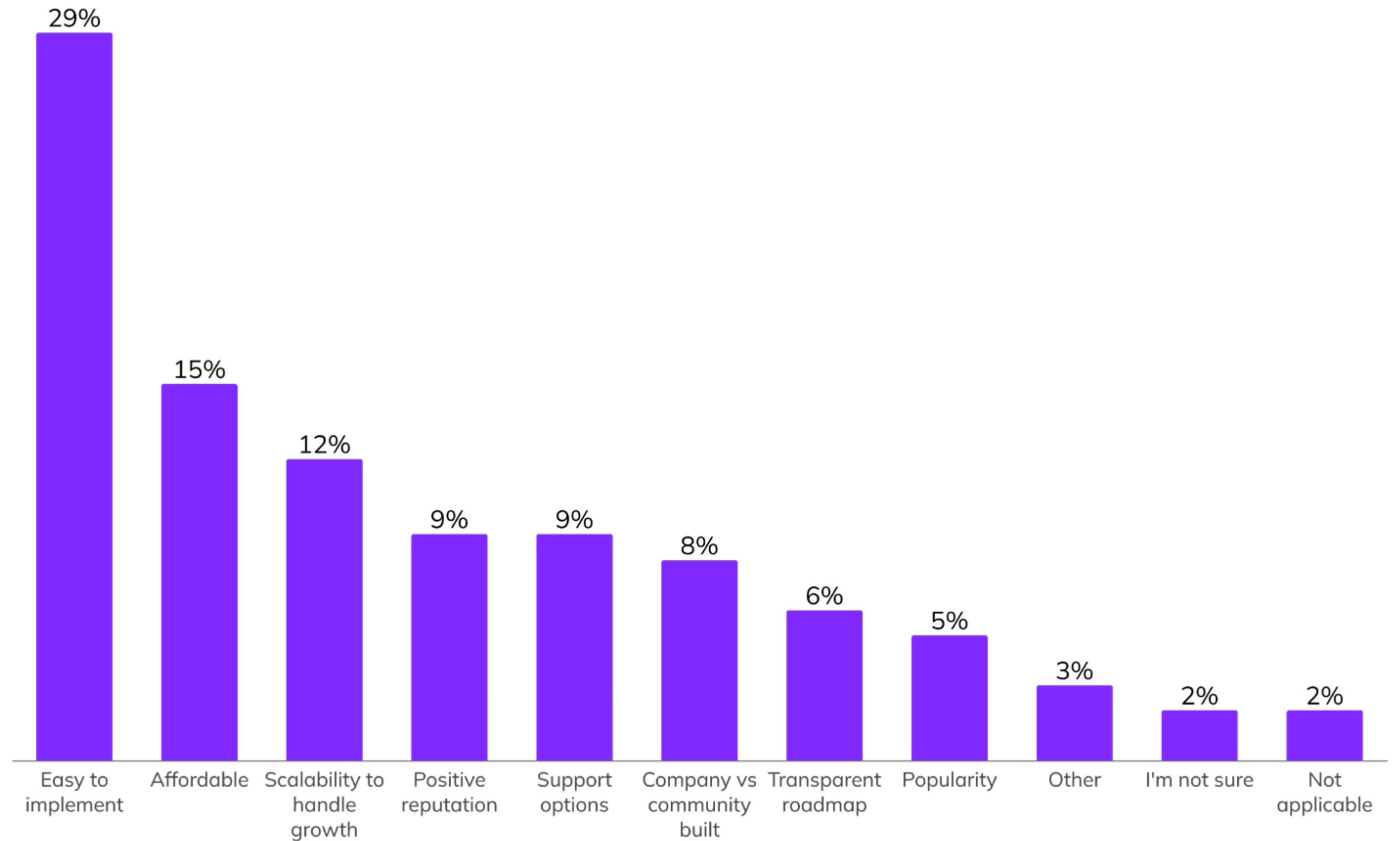
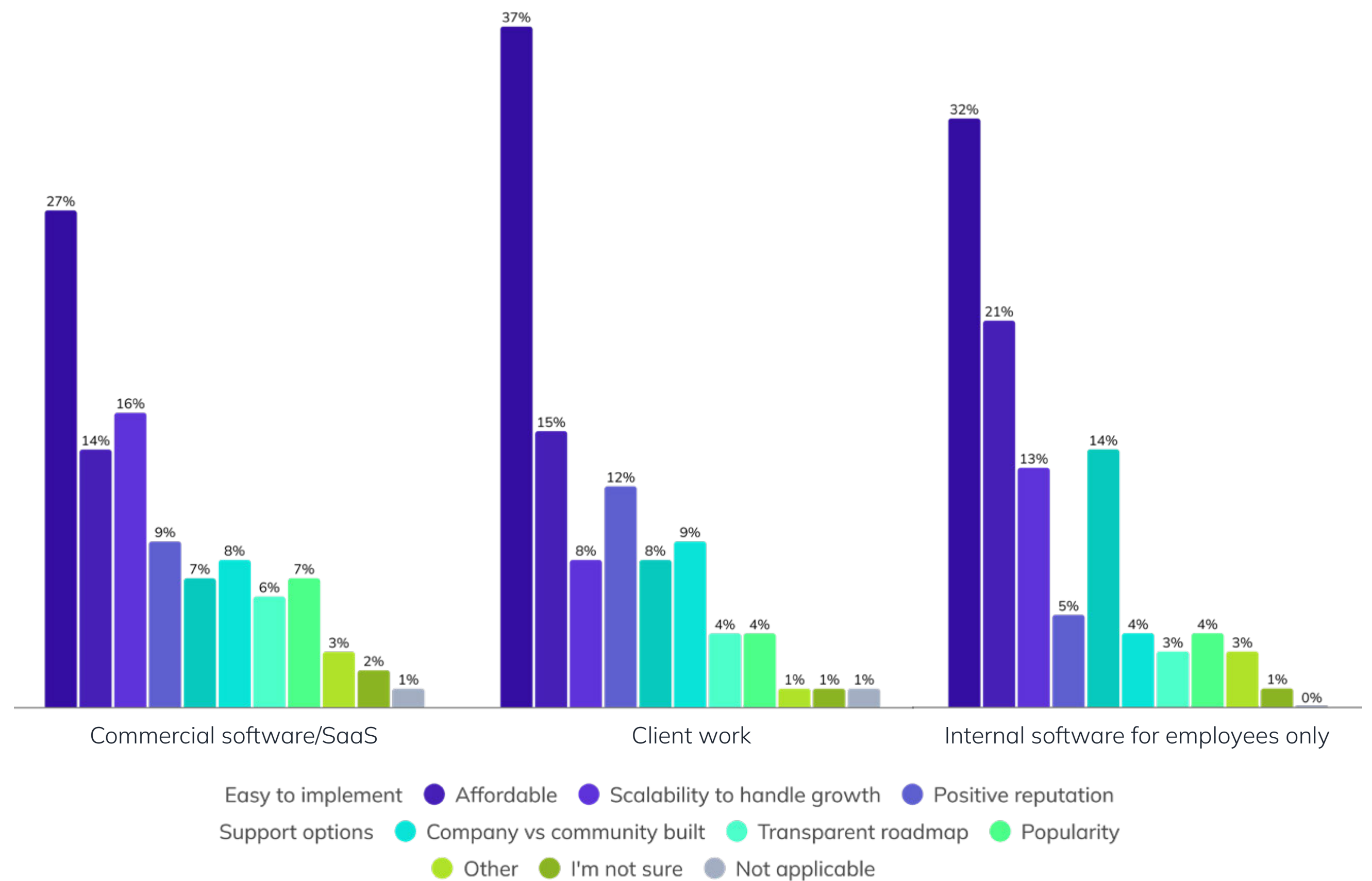


Fig. 18: What do you value most when evaluating a third-party software component?

Those building internal software were more influenced by cost than scalability, although implementation ease still remained the top priority. For those building commercial software, scalability held more sway than affordability.

When it came to client work, though, a tool's positive reputation rated nearly as high as affordability, with scalability dropping to the 4th most important consideration in these cases.

Fig. 19: Data by type of development work - What do you value most when evaluating a third-party software component?

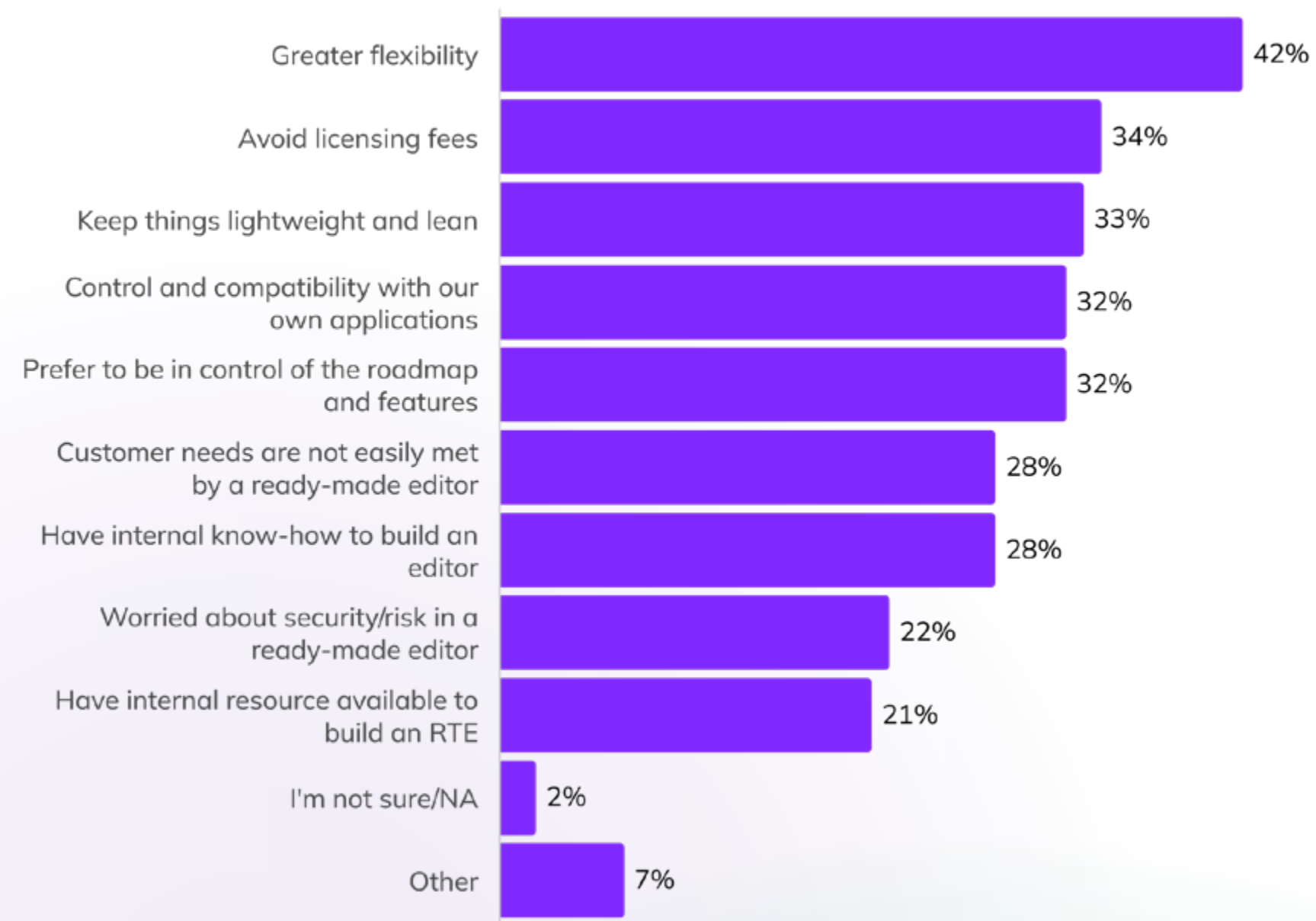


The choice to build

Choosing to build an RTE instead of buying one is a big decision.

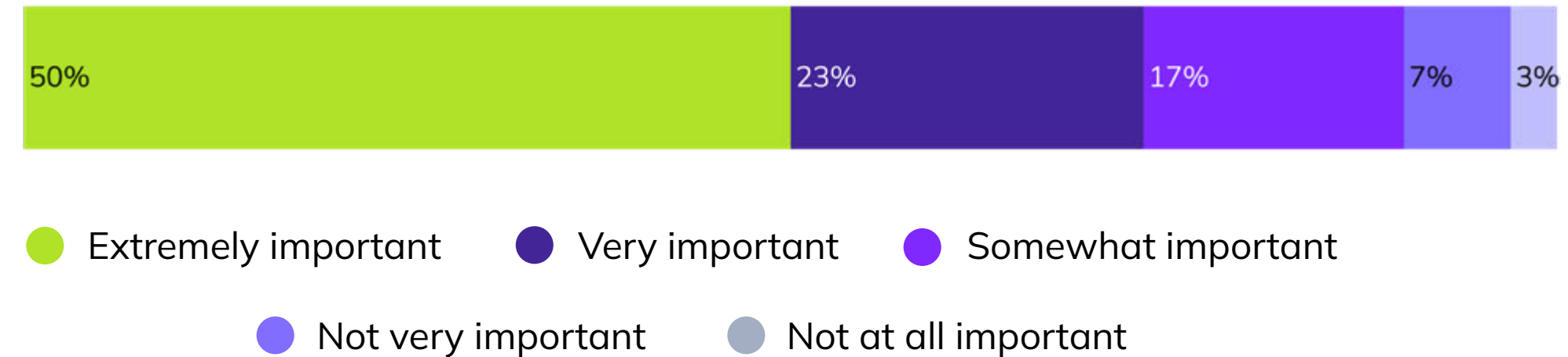
The choice to build was driven most by things like having greater flexibility (42%), licensing fees (34%) and having a lightweight component (33%).

Fig. 20: What drives your choice to build your own editor in-house instead of buying a ready-made editor?



Builders aren't just those creating a quick solution in house. Despite the hurdles, those who build their RTE say collaboration features are important (73%) - despite these features being difficult and time consuming to create. Syncing changes from co-editing, preserving changes across revisions, and tracking what was changed and by whom are considerable engineering challenges.

Fig. 21: Respondents who build an RTE in-house - For your application, how important are collaboration tools and features?

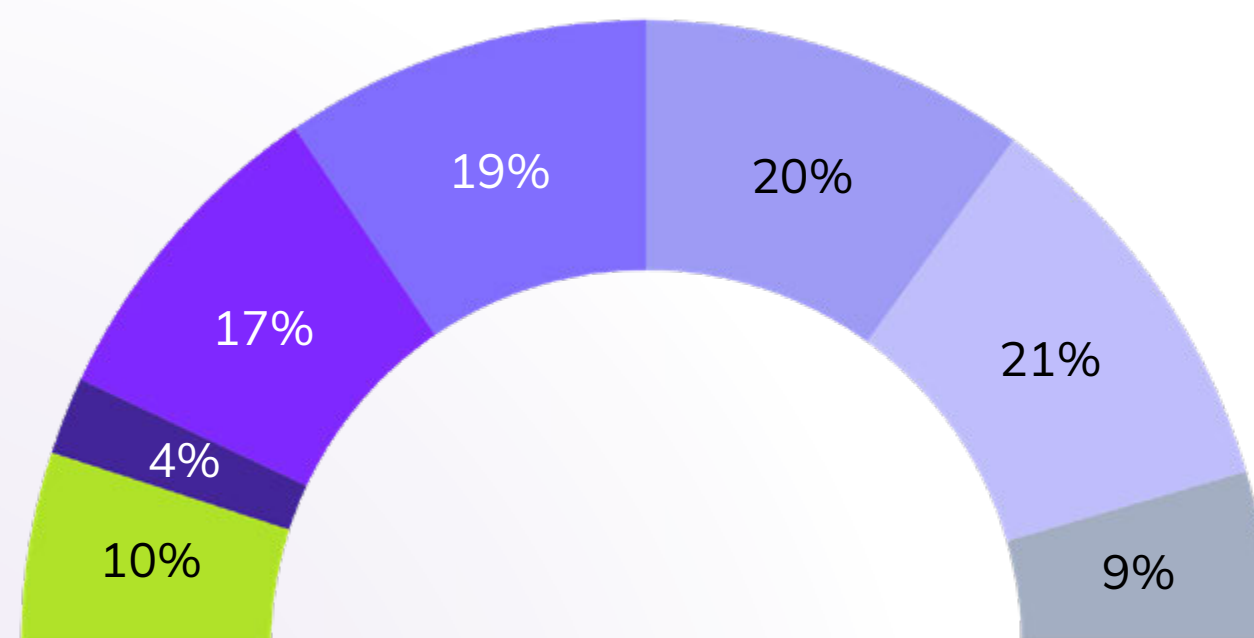


That's likely why there are significantly more large companies willing to tackle a project like a home-grown RTE. 32% of companies with over 100 employees indicated they have built their own, compared with only 23% of companies with between 10 and 100 employees.

Central to many buy vs. build conversations are topics like technical debt and maintenance. Of those choosing to buy a third-party solution, 37% pointed to long term development and maintenance costs as a reason for their decision, while 42% indicated that buying a tool would allow them to free up internal development resources. (Fig. 17)

When all respondents were asked how often they managed, updated, or worked on their rich text editor, 20% indicated they did so at least once a year, while only 17% touched their RTE at least once a month.

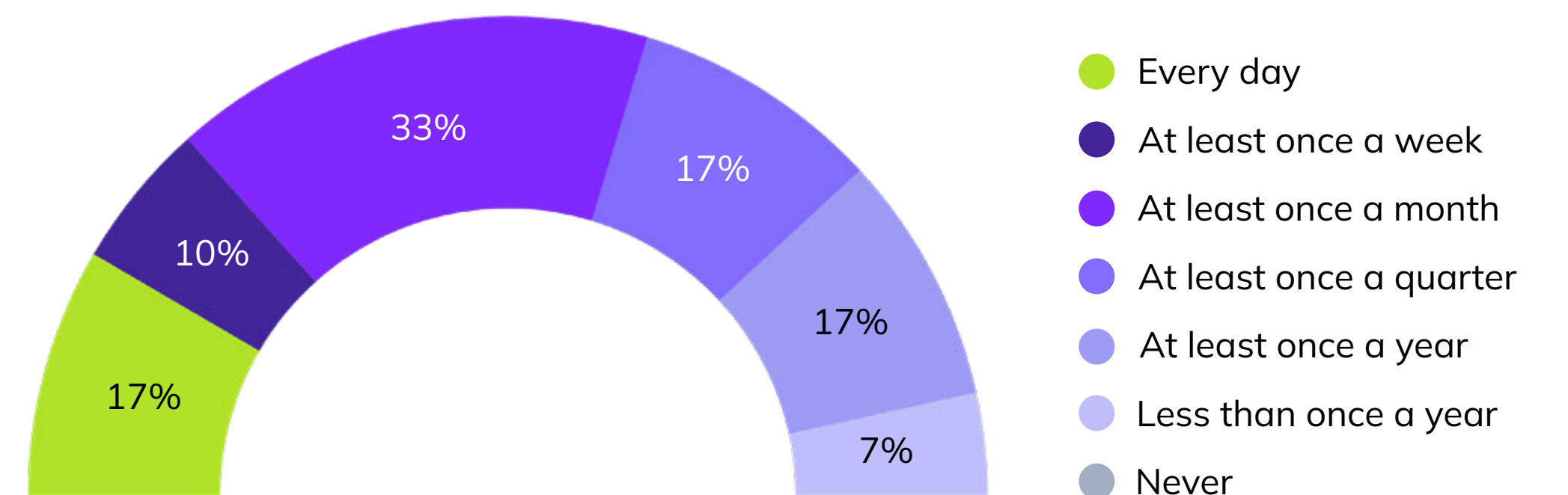
Fig. 22: How often does your team manage, update or develop a rich text editor?



When we isolate the maintenance responses for those who have built their own RTE, a telling trend emerges. The numbers shift to more frequent maintenance and development for those who built an RTE in-house.

Of those that had an internally built or self-built editor, the number that worked on their RTE at least once a year dropped 3 points to 17%. Importantly, those that developed, updated, or maintained their own editor reported a significant increase in monthly maintenance, jumping from 17% for all surveyed to 33% for just builders. Tellingly, 17% of those who built their own RTE indicated that they worked on their editor every day.

Fig. 23: Respondents who build an RTE in-house - How often does your team manage, update or develop a rich text editor?

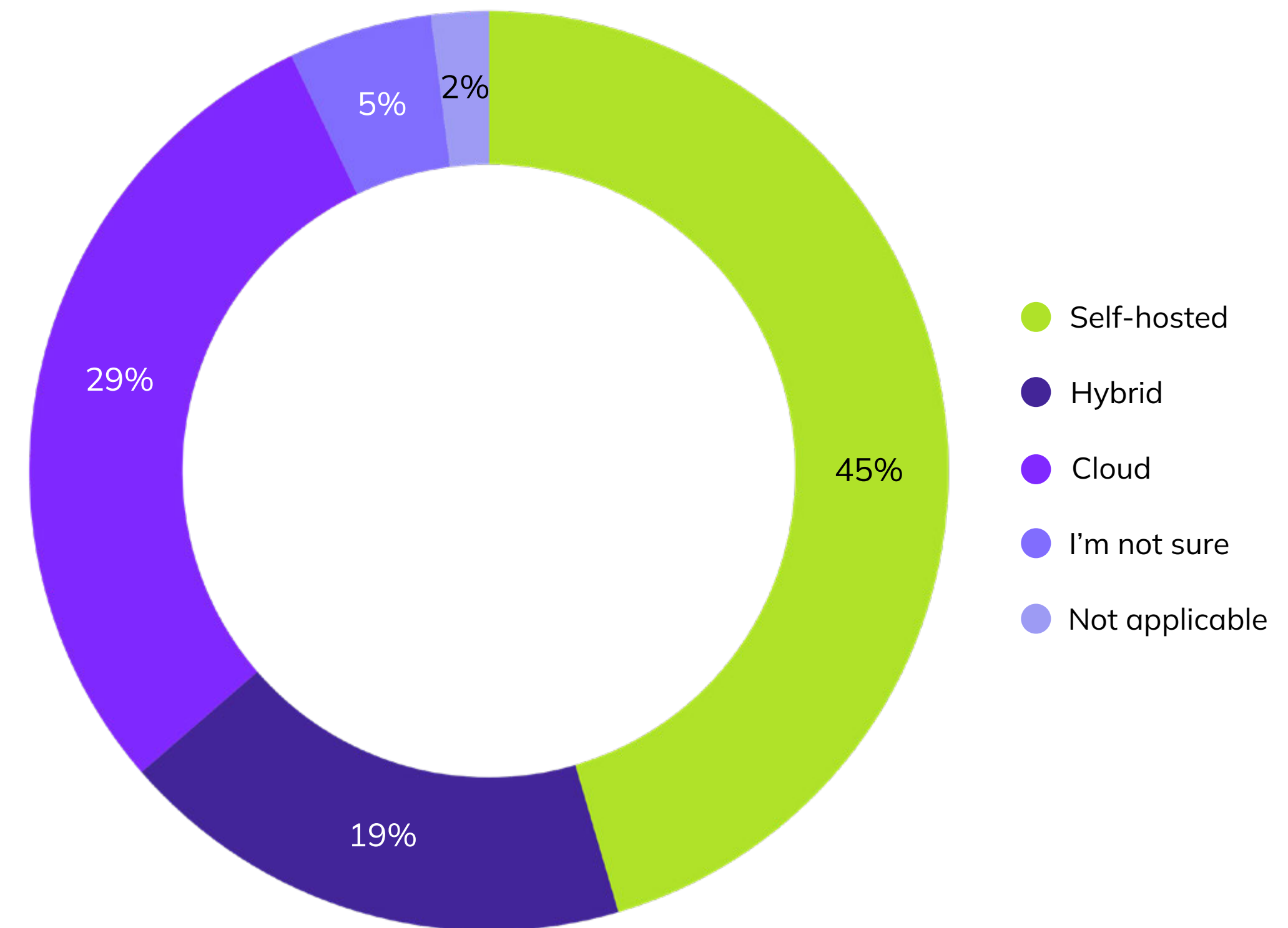


Deployment choices: Cloud, self-hosted, and hybrid

Where a rich text editor is deployed is as important a choice as what it can do. Some companies have mandates for where their applications are deployed. For instance, self-hosting may be driven by compliance, security protocols, data region requirements, and governance. Cloud deployments may be most important for those focused on scalability, resource dependencies, and accessibility.

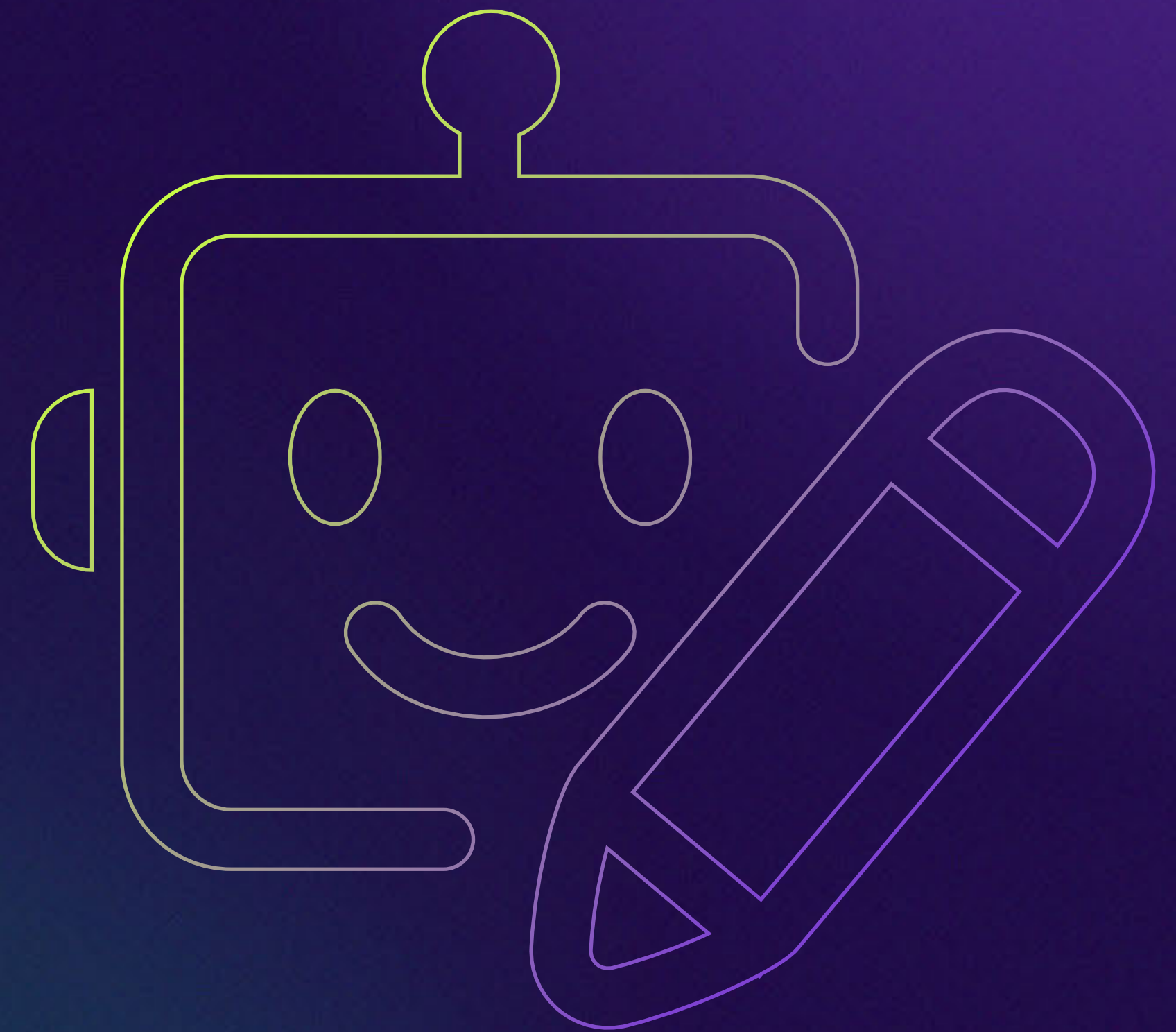
Nearly half of survey respondents indicated that they preferred self-hosting their RTE over cloud or hybrid options.

Fig. 24: What is your preferred deployment option for your rich text editor?



Chapter 4:

The Future of Collaborative Editing



Where is collaborative editing headed?

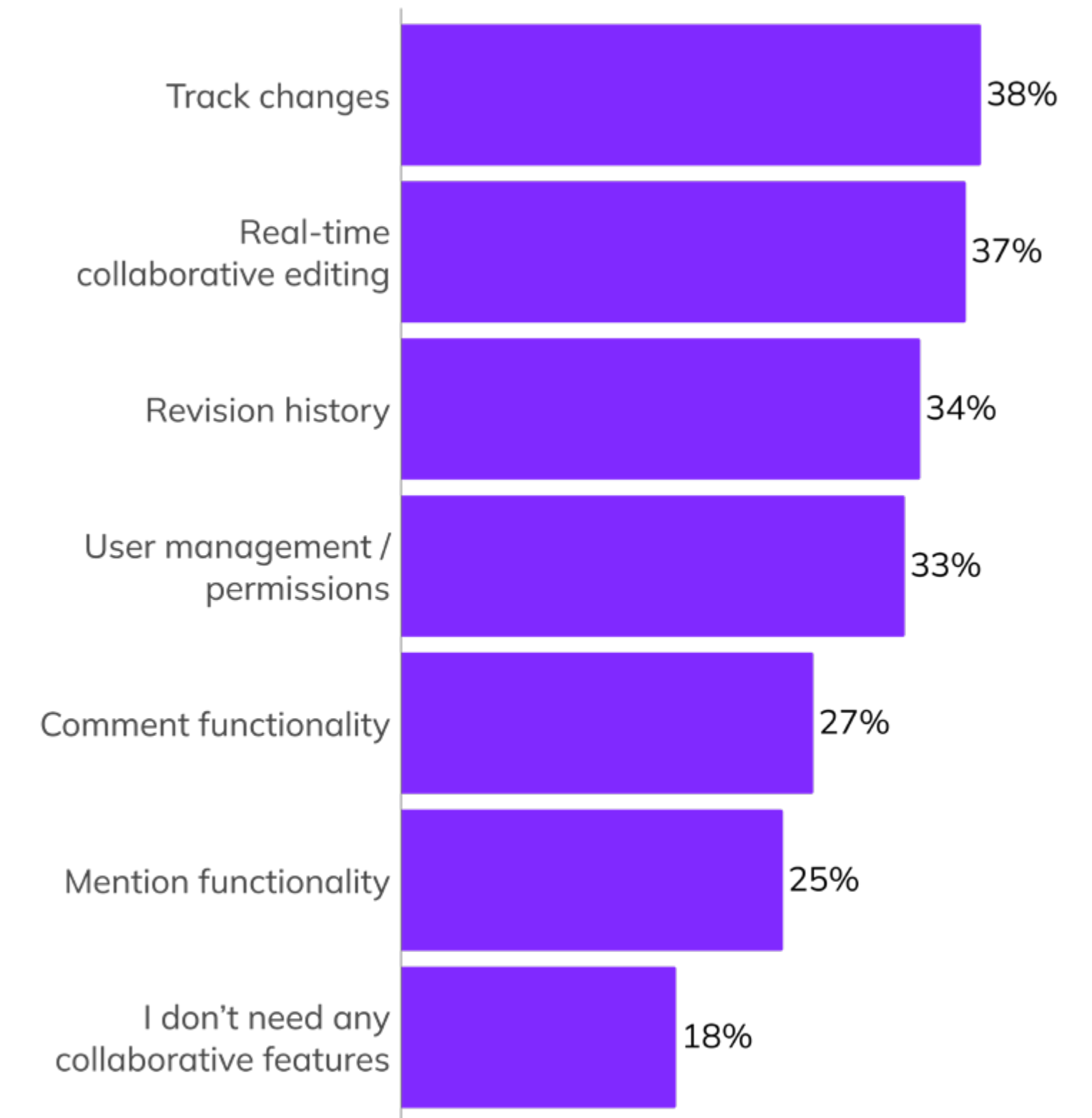
None of us have a crystal ball that tells us what the future holds. But, looking at the world of rich text editors and collaboration features, the trends and predictions made by those closest to these tools can give us a glimpse into what's on the horizon.

It's no surprise that many of the responses from survey respondents highlighted AI and LLMs. AI-centered technologies may lead the conversation when it comes to collaboration, but existing collaborative tools are expected to rise in importance, as well.

Respondents identified track changes and real-time collaborative editing as two of the most critical features in the next 5 years, each getting about 37% in a multi-choice question about the future of collaboration. Revision history and user and permissions management followed closely behind, at 34% and 33%, respectively.

As developers, product owners, and technical leaders look to their own roadmaps, it will be important to keep these features in mind. **Platforms that have mature and high-functioning collaboration already in place will be able to offer more stable and supportable solutions with the collaboration features users will demand.**

Fig. 25: In the next 5 years, which of the following collaborative editing features will be critical?



The most helpful collaborative editing technologies

While some collaborative features are considered critical capabilities in the next few years, respondents predict others to be more helpful than required. Among those tools, AI-powered options rank high. AI-powered grammar checking (46%) and AI-generated content (45%) are viewed as having the highest potential. Robotic Process Automation (RPA) was seen by 19% of respondents as a helpful future tool, while 17% of respondents didn't feel any future advances were needed.

When asked where LLMs and AI would have the greatest impact on RTEs over the next year, respondents leaned heavily into content creation (30%). Only 3% of respondents felt that AI wouldn't have an impact on RTEs at all.

Fig. 26: What future technology do you see as being helpful when it comes to collaborative editing?

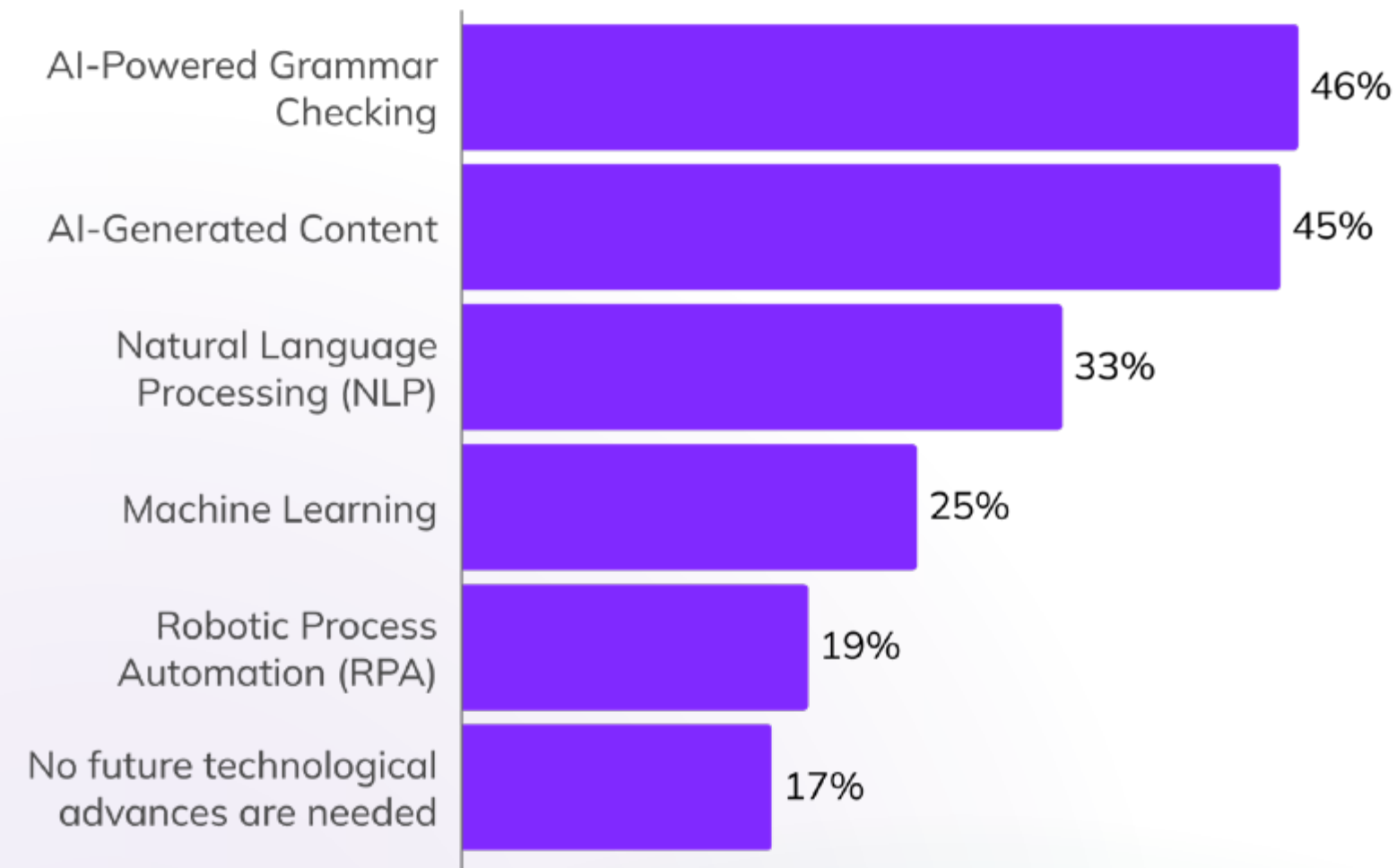
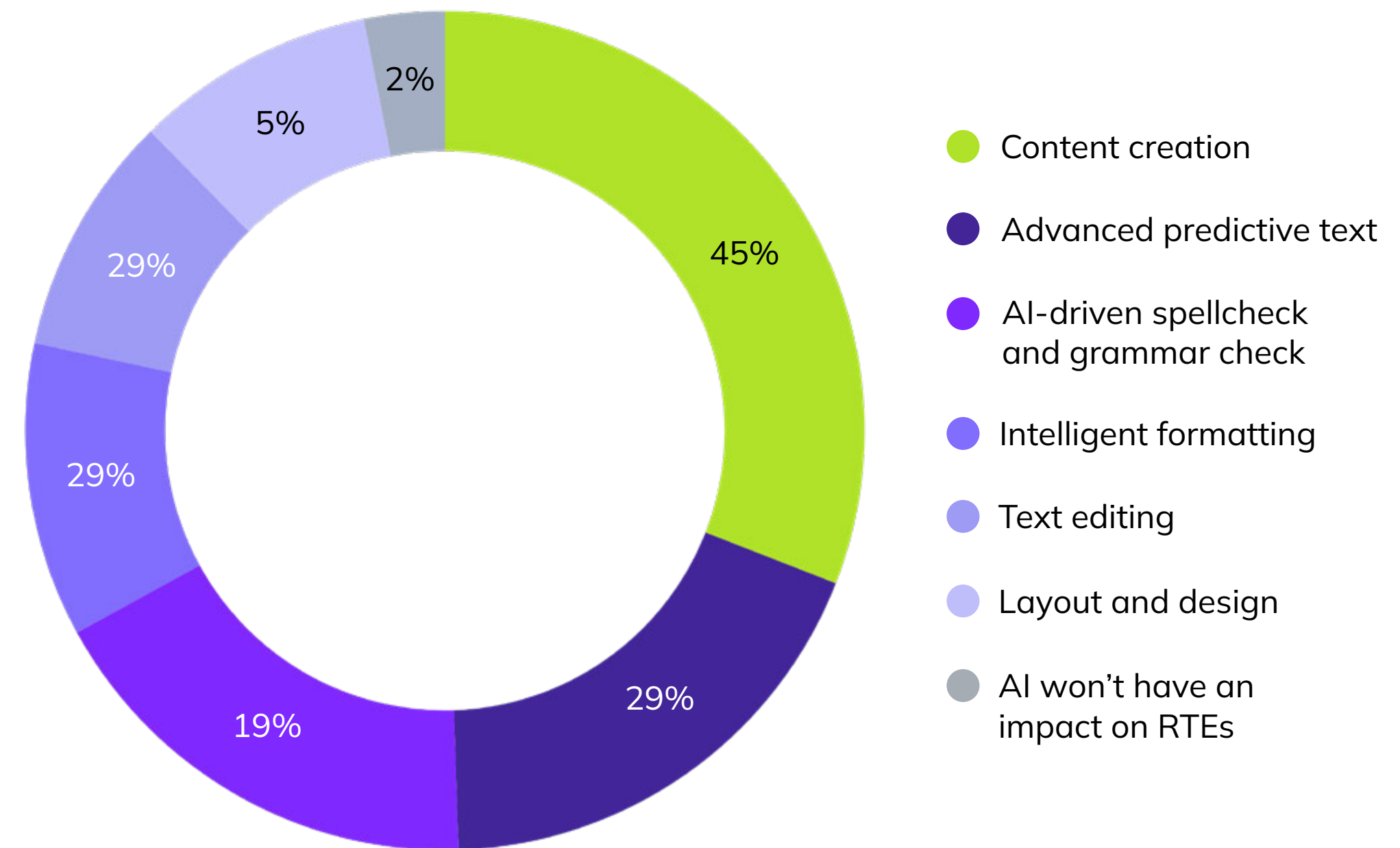


Fig. 27: Where do you think LLMs and AI will have the biggest impact on rich text editors in the next year?



Upcoming shifts in the space of collaborative editing

When we asked respondents, “What big underlying changes or game-changing features do you see in the future?” they had a lot to say about what they saw on the horizon for rich text editors and collaboration tools.

Generative AI

Having seen the quantitative results of the survey, it’s no surprise that many answers to an open-ended question looking ahead talked about the value AI would bring. No one expected that AI would replace their content creators. Instead, respondents saw an opportunity for AI to work alongside editors, with the final say and collaboration continuing to require human oversight.

Gen AI will certainly impact this space significantly, but in certain industries, it will not eliminate the need for finalized and collaborative documents.

AI wasn’t seen as only a rich text editor feature, however. Instead, some saw the potential for AI to act as another collaborator in the content creation process.

Updating the content will be less about one author typing and formatting content and more about multiple authors (including AI) merging their changes together into a cohesive document. We haven’t yet seen the final forms this will take - we’re still seeing users and AI basically copy/paste content together... Not a real melding of the inputs from various sources.

The growth of synchronous editing

As we saw in Fig. 25, 37% of those answering the survey thought that real-time collaboration would become a critical capability in the next 5 years.

I also think that there will be a shift towards more collaborative and real-time editing, where multiple users can work on the same document simultaneously.

The prediction is that these features will be expected over the next few years, rather than their current position as a nice-to-have addition. Real-time collaboration and co-editing capabilities will become mainstream, allowing team members to edit the same document at the same time.

Privacy and security

Regardless of industry, privacy and security are top priorities. This is true with RTEs. As collaboration and co-editing become the norm, security within RTEs will continue to move up the list of importance for many developers and product owners, as well as their users.

As privacy concerns continue to grow, I believe there will be a focus on creating more secure and encrypted options for sharing and storing documents within rich text editors.



Conclusion



Conclusion

Collaboration isn't a fad. It's another step in the maturation of web applications. Much like "Web 2.0", the rise of JavaScript frameworks like React, Angular, and Vue.js, and Cloud Architectures, in-application collaboration features are becoming expected elements of modern applications.

Being able to collaborate on documents has proven itself by accelerating teamwork, creating efficiencies, and encouraging accountability. Real-time collaboration brings together people to share ideas without constraints.

This style of collaboration in teams and across organizations is still relatively new. Users are still willing to tolerate cobbled-together processes that involve multiple applications and storage mediums.

But as these collaboration methods become burdensome and the risks become apparent, the disruption of collaboration across products will lose its appeal. Users will demand seamless collaboration within the applications they already use.

Full-featured rich text editors are ready to answer the call for in-app collaboration. Because it is now possible to integrate collaborative features in applications in a matter of weeks, collaborative capabilities are no longer exclusive to the big tech companies like Google (with Google Docs) and Microsoft (with Office Suite). Many products have already integrated collaboration into their applications, and many more plan to do so in the very near future. It's safe to say that applications that could benefit from collaborative features but don't have them will be quickly left behind.

It's time to recognize the value collaboration can offer your users. And it's time to embrace the collaborative tools that will bring those features seamlessly into your applications.

