CKEditor

State of Collaborative Editing

2025

Table of Contents

Introduction

A renaissance in collaborative editing

In the 2020s, the trend of decentralized teams accelerated. The pandemic drove the shift, but decentralization eventually became the way we just work. Teams started collaborating across different cities, countries, and continents. The infrastructure that enabled this also freed employees to work whenever they wanted, whether that was in an office during business hours or on their smartphones while at lunch.

Collaboration wove into content editing as well. Rich Text Editors (RTEs) became a central gateway for fostering team collaboration. Real-time co-authoring, in-context commenting, and AI-assisted drafting turned the RTE into a shared workspace for teams sitting in different offices around the globe. The RTE has become a frontline enabler of collaboration, communication, and productivity.

In 2023 and 2024, the dust appeared to have settled on collaboration's rising demand. The importance of collaborative features for RTEs dipped, with a full 13% of users even claiming collaboration features weren't important at all for their RTEs.

In the third year of our State of Collaborative Editing report, we've seen the appetite roar back to life. This year's data shows renewed momentum for collaborative editing.

Of course, this heightened focus places burdens on teams. C-suite expectations push employees to produce more with fewer resources. Development teams face added strains integrating collaborative features into their applications and editors. Still, collaboration remains a competitive advantage, making effective implementation essential.

So what's driving this renewed interest? What complexities do organizations contend with in this new era? This year's report answers these questions and highlights the key trends and insights in the space. With it, you'll be able to make informed decisions that lead to cleaner, more effective content collaboration, regardless of organizational complexity or application demands.
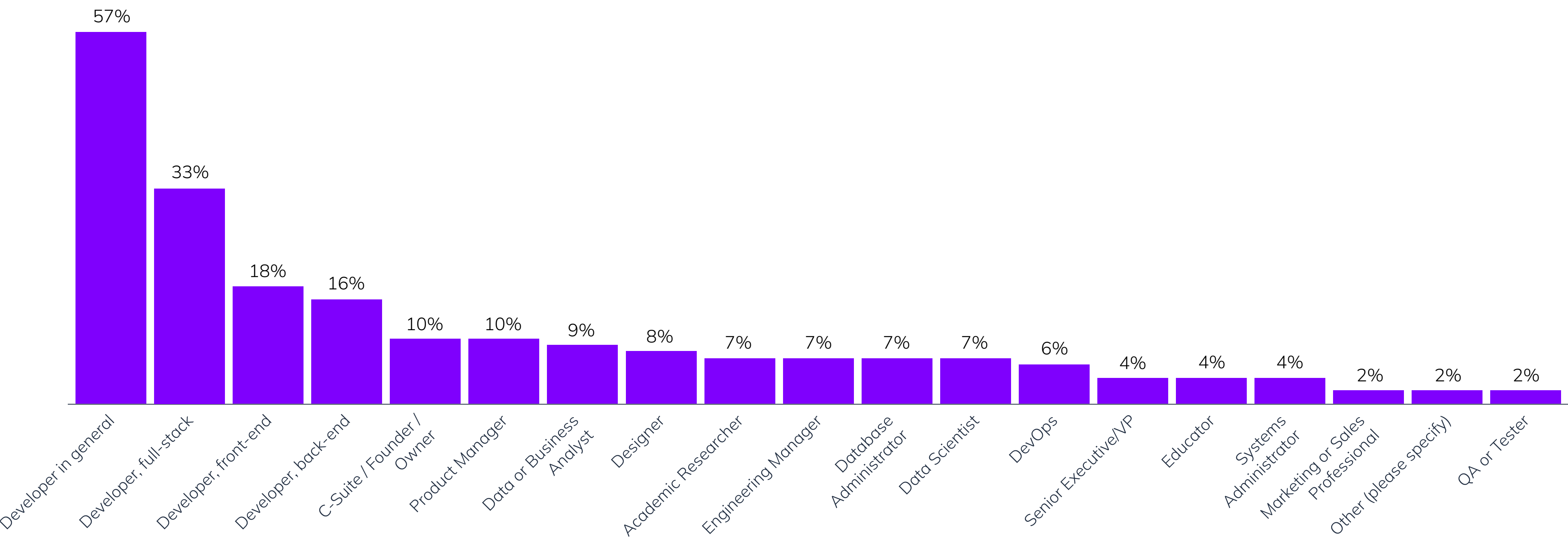
Introduction

Who we surveyed

CKEditor conducted this survey in the second quarter of 2025 in partnership with DO FEEL THINK. The sample size included 534 technical professionals who develop or manage rich text editors in their applications today.

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. It also included participants working on individual projects and in organizations, from a single developer to large enterprises.

Fig. 1: Please select the description that best captures your role.

Role	%
Developer in general	57%
Developer, full-stack	33%
Developer, front-end	18%
Developer, back-end	16%
C-Suite / Founder / Owner	10%
Product Manager	10%
Data or Business Analyst	9%
Designer	8%
Academic Researcher	7%
Engineering Manager	7%
Database Administrator	7%
Data Scientist	7%
DevOps	6%
Senior Executive/VP	4%
Educator	4%
Systems Administrator	4%
Marketing or Sales Professional	2%
Other (please specify)	2%
QA or Tester	2%

Introduction

CKEditor

Chapter 1:

The Collaborative Editing Landscape

The Collaborative
Editing Landscape
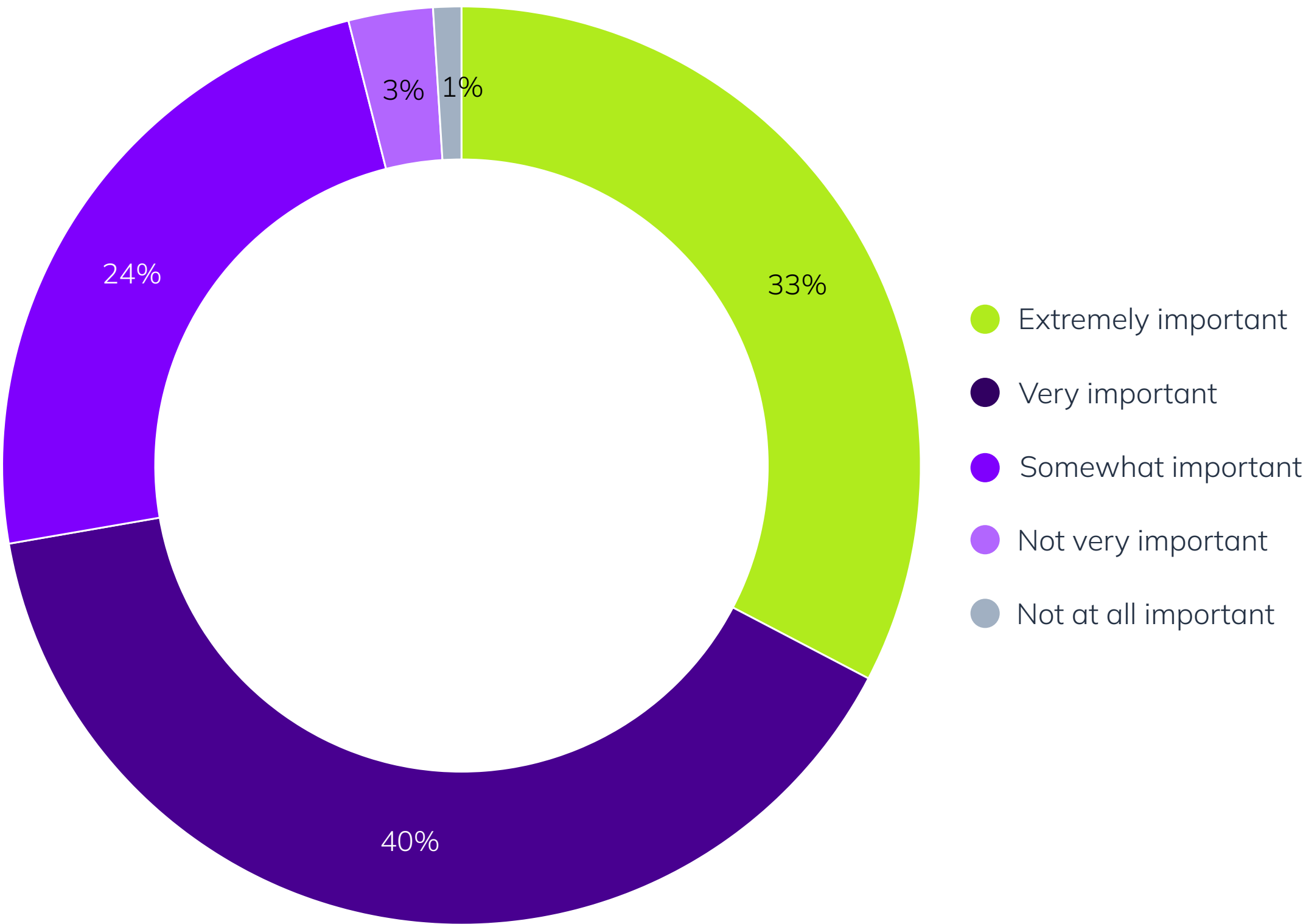
The importance of rich text editors

Communication is the beating heart of many applications. Whether in project management, content management, document management, or learning management systems, many applications can't function without strong text and content editing capabilities.

The data bears this out. When asked about the importance of RTEs to their application or product, 73% of respondents chose either "extremely important" or "very important."

This finding underscores a broad reality about software: almost every product has become a content platform. RTEs aren't nice-to-haves— they're core infrastructure. They're where people get work done, from collaboration to data capture to workflow automation.

Because of this, developers need reliable RTEs that are extensible, easy to integrate with their tech stacks, and future-proof. End users need RTEs that offer a strong user experience, both in terms of editing and collaboration.

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

- 33% — Extremely important
- 40% — Very important
- 24% — Somewhat important
- 3% — Not very important
- 1% — Not at all important

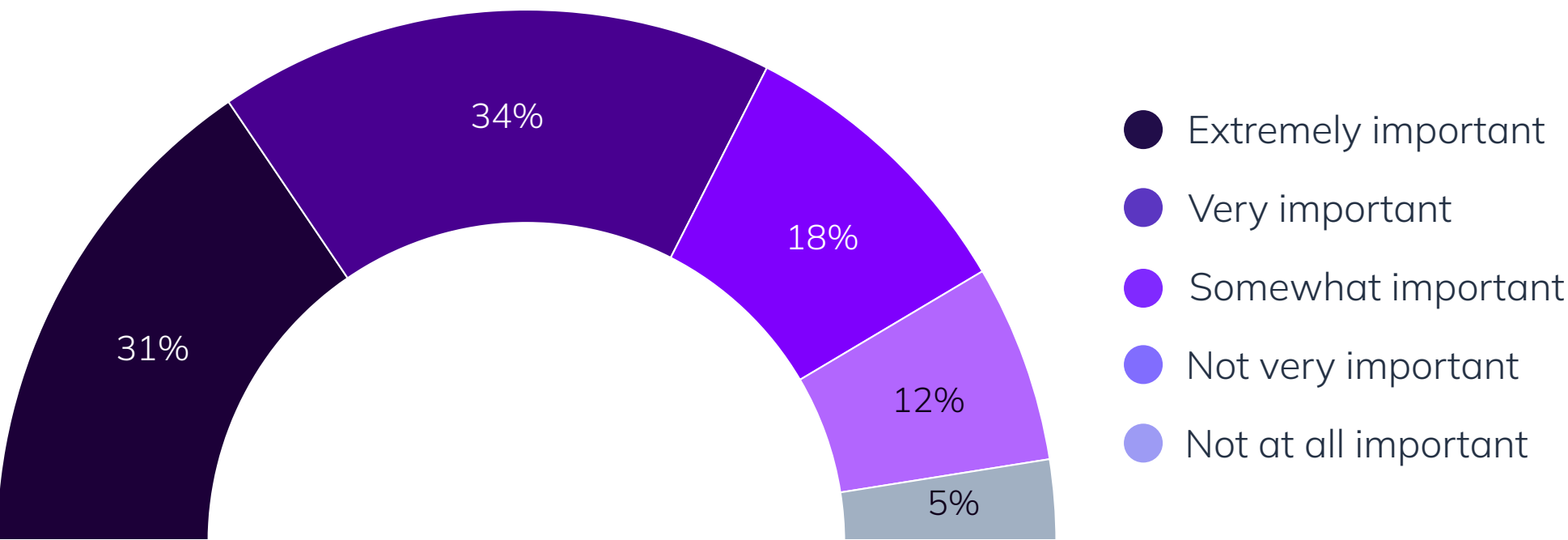**The Collaborative Editing Landscape**

The role of collaboration within RTEs

As RTEs grow in importance, collaboration becomes a must-have. This year, a full 65% of respondents identified collaboration features as either extremely or very important. Only 5% claimed that collaboration features weren't important at all, which was a massive shift from last year's survey, where 13% noted collaboration wasn't important for their applications.

This rise in importance to technology professionals likely emerges from user expectations. Application users have a strong appetite for collaboration tools: 62% of respondents cited either significant or considerable user interest in these features. This grew a whopping 22% from last year.
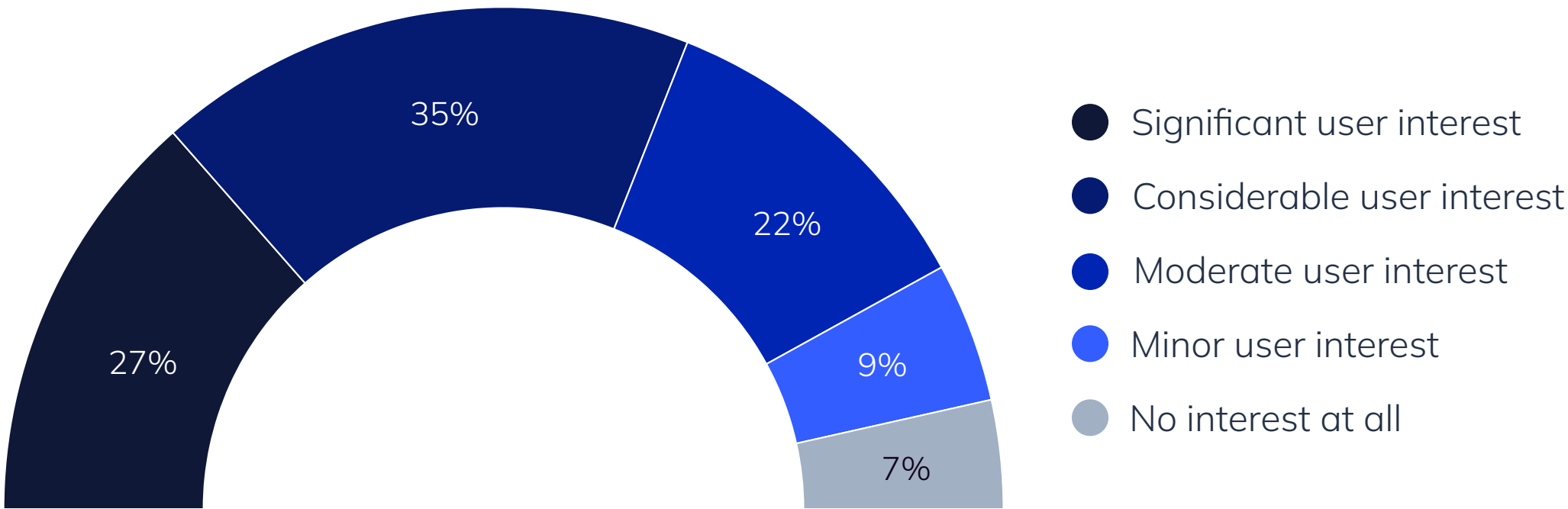
What's behind the growth? First, distributed workforces, with teams working across time zones, make both real-time and asynchronous collaboration critical for completing work. Second, AI tools now act as collaboration partners (more on this later), accelerating review and creation cycles. Third, businesses must do more with fewer resources. Boards and C-suites want to get more from their existing teams, and efficient collaboration cuts down on review cycles and project length.

It's a critical strategic enabler. And as AI usage expands, we should expect the demand for collaboration to grow even more.

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

- Extremely important
- Very important
- Somewhat important
- Not very important
- Not at all important

34%
18%
31%
12%
5%

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

- Significant user interest
- Considerable user interest
- Moderate user interest
- Minor user interest
- No interest at all

35%
22%
27%
9%
7%

The Collaborative
Editing Landscape

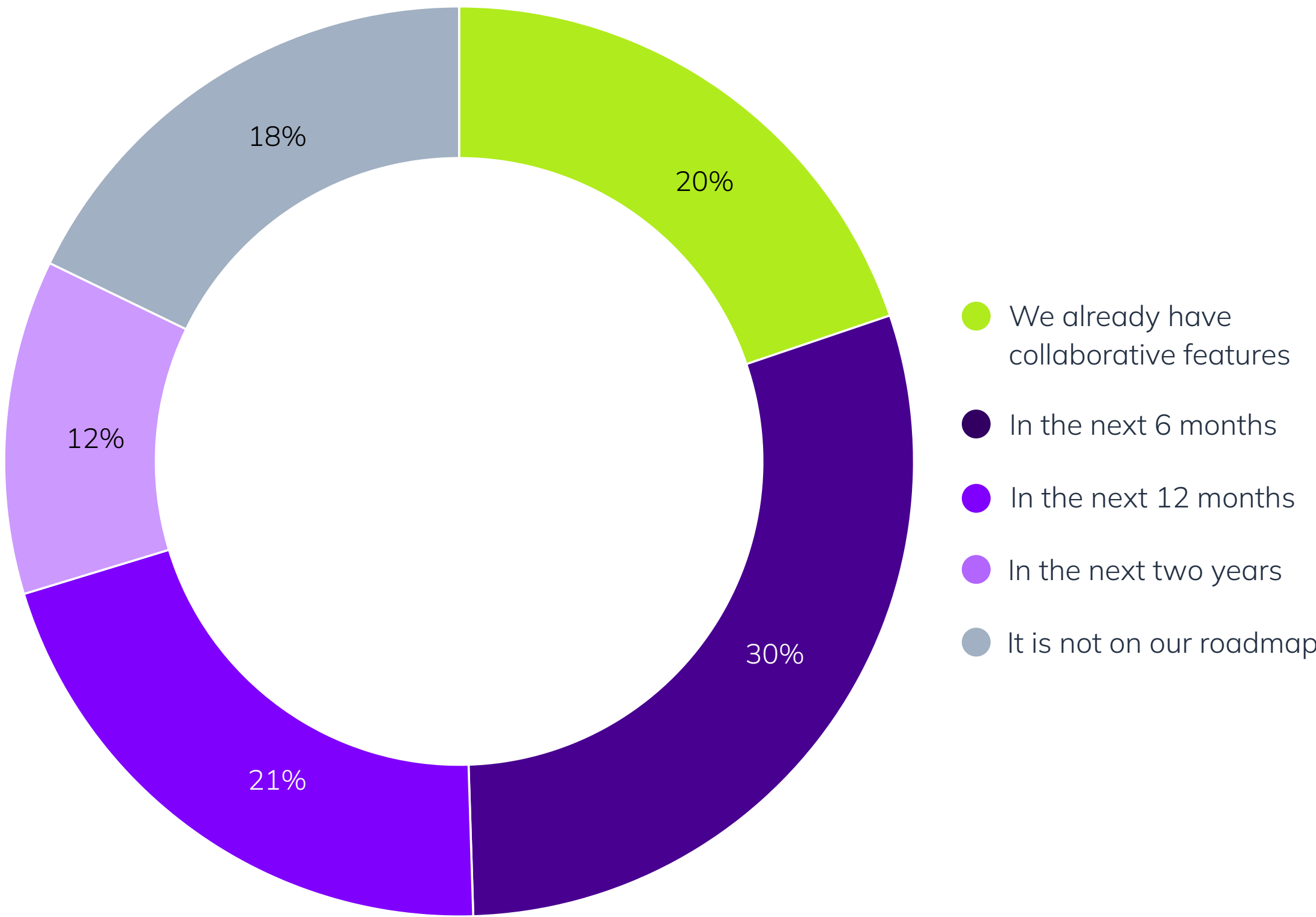# Implementing collaboration tools

CKEditor

So, demand has skyrocketed for collaboration tools. But how well are businesses meeting this demand?

It's mixed. Seventy-one percent either have collaboration tools already implemented or plan to add them in the next year. But a full 18% don't even have collaboration tools on their product roadmap. Users who lack these features can lose hours per week waiting on approvals, addressing comments asynchronously, or facing time lags between content and review.

Given that, why wouldn't product teams prioritize these features? For the most part, it's about capacity. Organizations can be constrained by legacy architectures or have different feature development priorities. In some cases, collaboration features might raise privacy concerns, although these occur more often in strictly regulated industries where risk tolerance is nil.
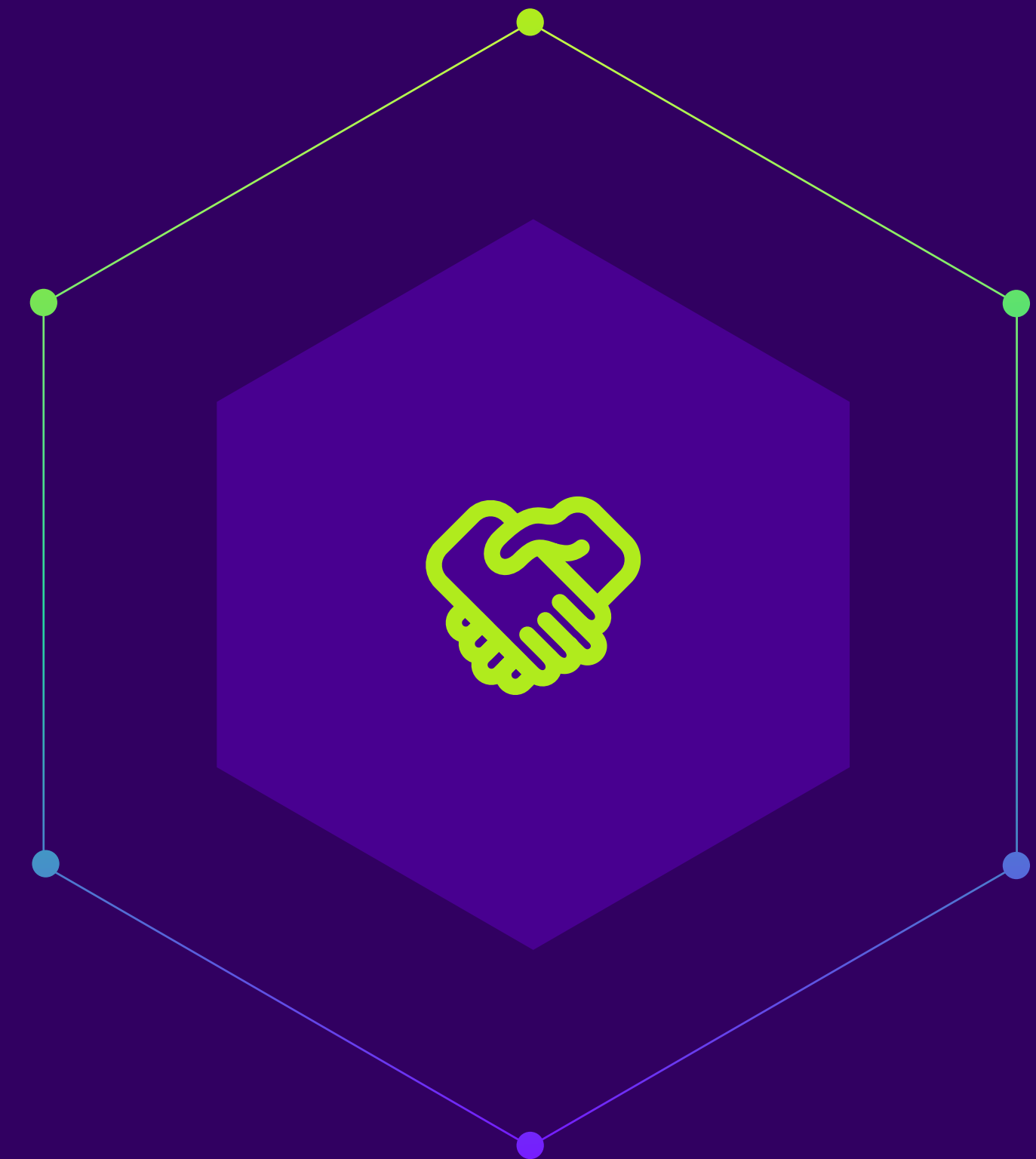
Whatever the reason, organizations should adopt collaboration tools as a matter of strategic imperative. Fast movers will capture compound gains, especially during this AI mainstreaming era.

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

- We already have collaborative features
- In the next 6 months
- In the next 12 months
- In the next two years
- It is not on our roadmap

The Collaborative Editing Landscape

9

CKEditor

Chapter 2:
# Collaboration Features and Experience

An exploration of collaboration features

Because RTEs have become communication hubs within applications, it makes sense that collaboration regained its footing since last year's survey. Collaboration boosts productivity, simplifies the lives of users, and leads to higher-quality content.

But collaboration is an outcome. How you get there is driven by the features you select. This path varies by company size. Organizations that lack requirements for collaboration features today will need them in the future. The smart play comes from picking RTE components that offer collaboration as an option now, so you don't have to switch editors down the line.

Also, this year, many collaboration features have become table stakes. However, artificial intelligence puts a new digital collaborator on teams, bolstering the workforce. As a result, we're seeing some new trends surface. The winners will be those who view AI as a collaborator and integrate AI tools directly into their RTE components.

In this section, we'll explore collaboration features with the greatest demand. We'll also break down demand by company size, showing which capabilities align with scale and where the data reveals unexpected patterns.

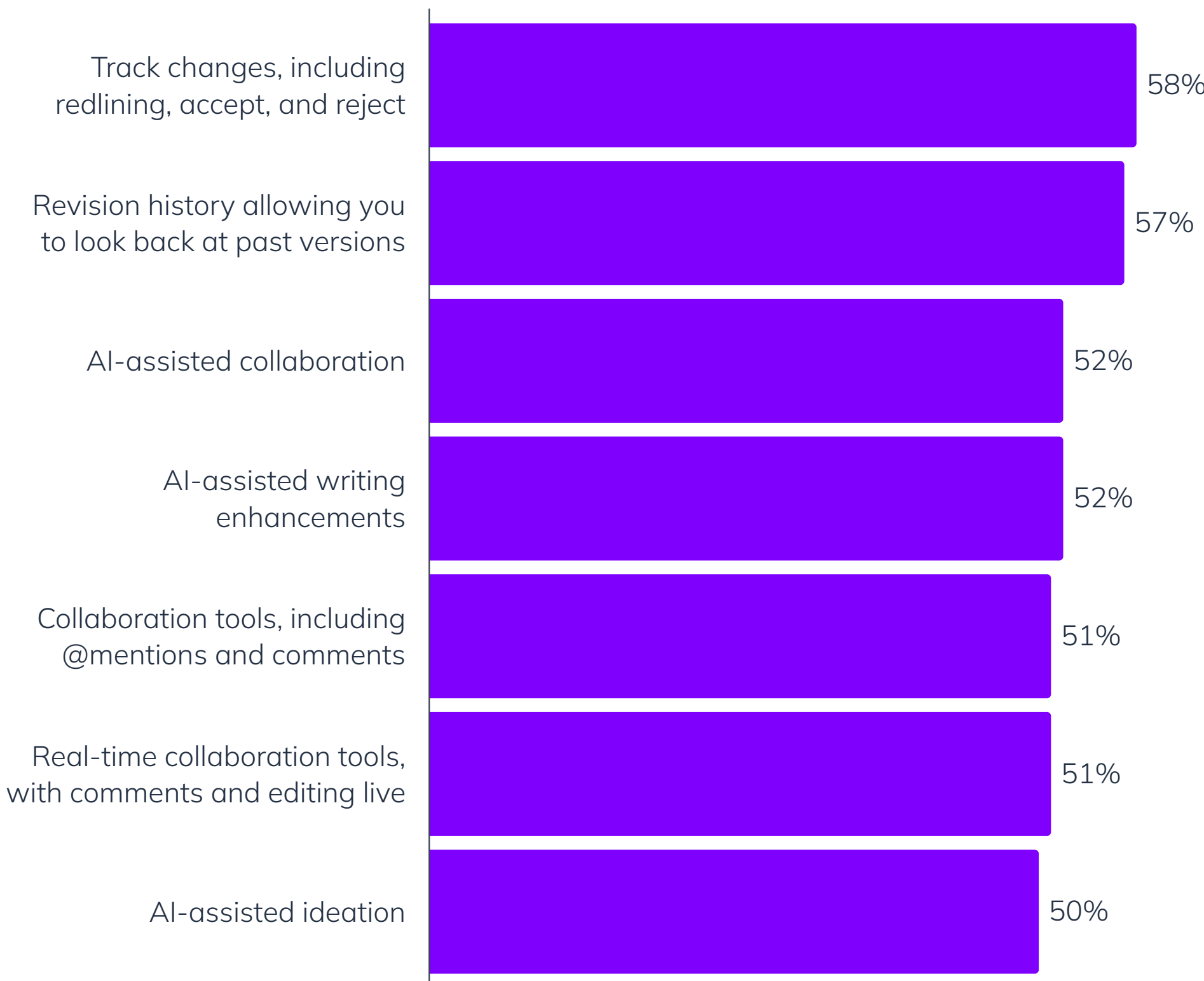Collaboration Features
and Experience
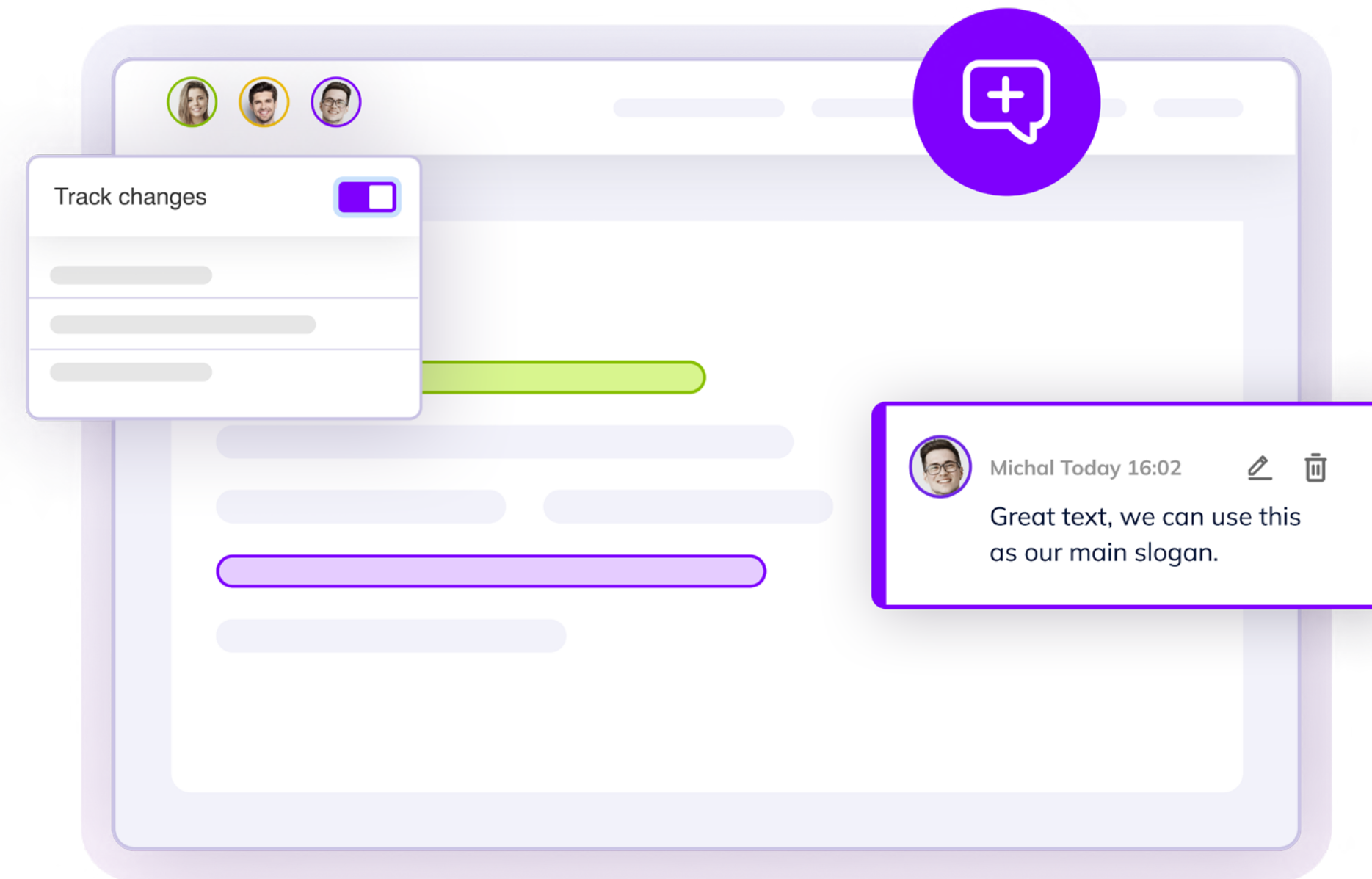
Rating collaborative tools

How did respondents rate each major feature? When comparing the wide range of rich text editor features overall, 51% placed collaboration tools, including mentions and comments, in the top two levels of importance. That number correlates with scale, reaching 61% for organizations with more than 1,000 employees.

When we dig deeper into the importance of specific collaboration features, track changes sits at 58%, revision history at 57%, and AI-assisted collaboration at 52%. This shows how some features have become expected, but it's notable that AI-assisted collaboration falls within 6 points of far more established features like track changes and revision history.

Today, more companies expect embedded AI as a standard for any RTE. Before you choose an RTE, AI must be on your feature list, even if it's not important currently. Doing so keeps you ahead of the curve, future-proofing your RTE choice and positioning you on the cutting-edge of collaboration.
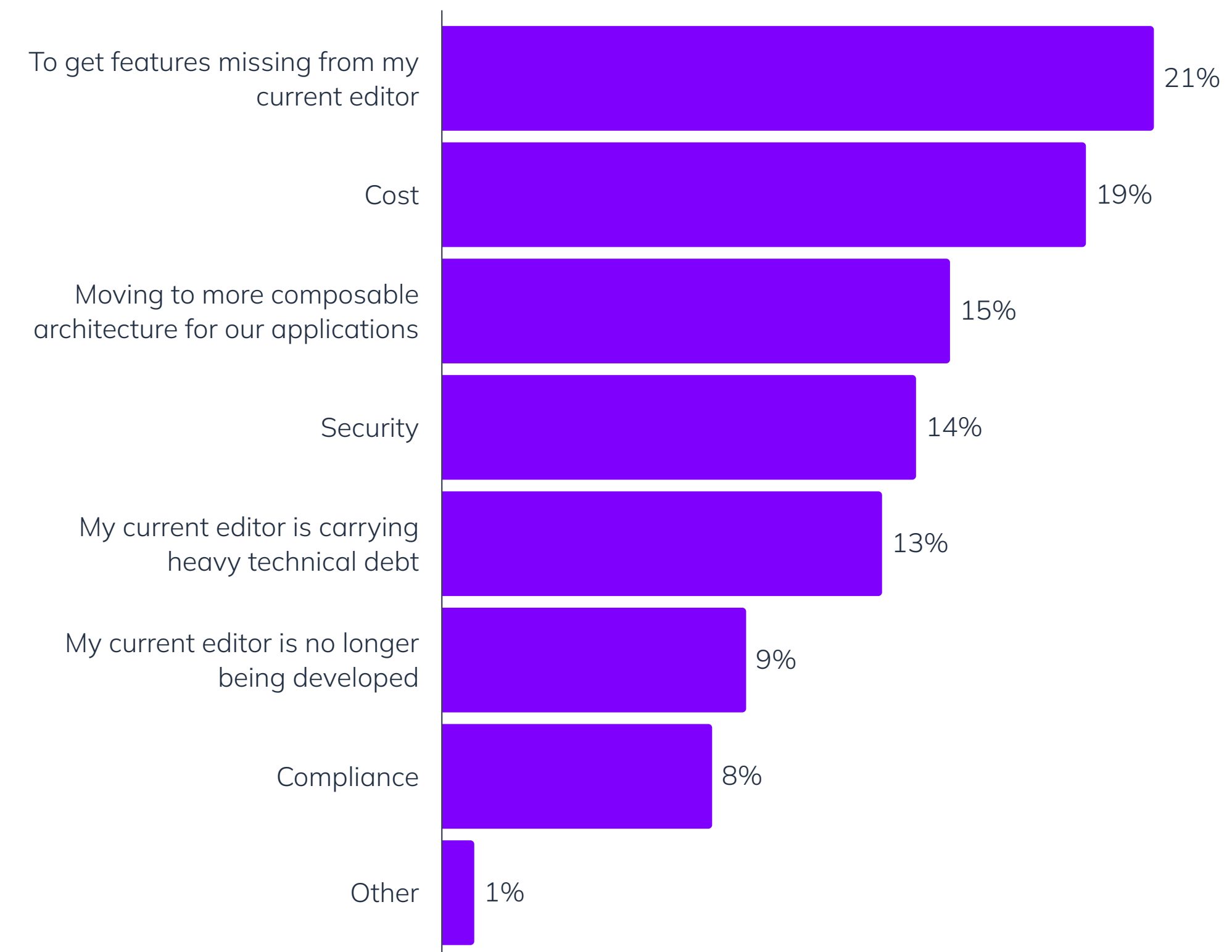
Fig. 6: How important are these features to your organization? (Extremely or very important)

Feature	%
Track changes, including redlining, accept, and reject	58%
Revision history allowing you to look back at past versions	57%
AI-assisted collaboration	52%
AI-assisted writing enhancements	52%
Collaboration tools, including @mentions and comments	51%
Real-time collaboration tools, with comments and editing live	51%
AI-assisted ideation	50%

Collaboration Features and Experience

In fact, to avoid the cost and complexity of switching RTEs at a later point, prioritizing future-forward features must be top priority. Those who plan to switch RTEs in the near future rated "missing features" as the primary driver (21% of respondents).

Switching can be costly. Too often, organizations consider only the upfront price tag or infrastructure requirements. Make sure to consider feature sets with an eye toward the future and, if possible, have user experience experts sit on your buying committee to ensure the editor meets your day-to-day needs.

Fig. 7: What is the primary reason you are likely to switch rich text editors in the next two years?
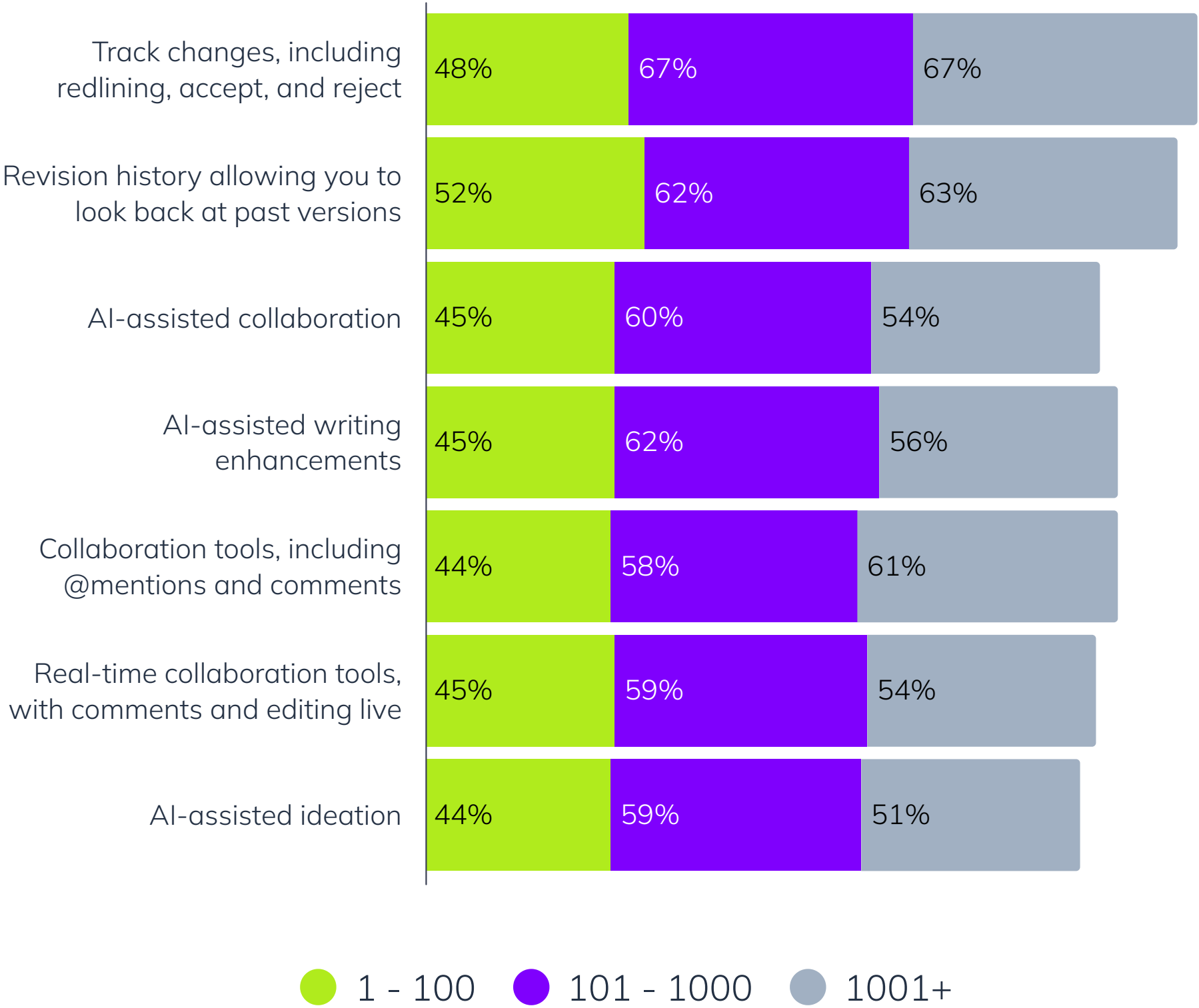

To get features missing from my current editor — 21%
Cost — 19%
Moving to more composable architecture for our applications — 15%
Security — 14%
My current editor is carrying heavy technical debt — 13%
My current editor is no longer being developed — 9%
Compliance — 8%
Other — 1%

Collaboration Features and Experience

Importance of collaboration features by company size

CKEditor

Most organizations agree that collaboration features are critical for any RTE or application. However, companies at different growth phases have varied needs around collaboration.

For collaboration features in general, only 44% of small organizations rated them as very or extremely important. Mid-sized organizations and large organizations showed much greater demand, at 58% and 61%, respectively.

This makes sense. Larger organizations must coordinate across more teams and departments, and their content faces longer, more formal review cycles. In some cases, content must clear several layers of management before final sign-off.

Fig. 8: Data by company size - How important are these features to your organization? (Extremely or very important)

Feature	1 - 100	101 - 1000	1001+
Track changes, including redlining, accept, and reject	48%	67%	67%
Revision history allowing you to look back at past versions	52%	62%	63%
AI-assisted collaboration	45%	60%	54%
AI-assisted writing enhancements	45%	62%	56%
Collaboration tools, including @mentions and comments	44%	58%	61%
Real-time collaboration tools, with comments and editing live	45%	59%	54%
AI-assisted ideation	44%	59%	51%

Legend: ● 1 - 100 ● 101 - 1000 ● 1001+

Collaboration Features and Experience

14

But for smaller organizations, individual creators hold more power. While 44% isn't insignificant, as businesses grow, they will inevitably need collaboration tools. It's helpful to select an RTE with mature collaboration capabilities early to prepare for growth.

Feature-set demand also varied by organizational size. Real-time collaboration was highest for mid-sized organizations at 59% compared to 51% overall. The track changes capability was very or extremely important for mid-sized organizations and enterprises at 67%, and revision history landed in these top two choices for both organization sizes as well, at roughly 62% for each.

Interestingly, AI-assisted collaboration demand peaked at the mid-sized organizational level at 60% compared to 45% for small organizations and 54% for enterprises. The reason is a bit of a head-scratcher, but it's likely due to AI adoption still being young.

Small organizations likely experiment with AI on an ad hoc basis or only use it if it comes standard with their existing applications. Employees likely access chatbots, but without uniform rules governing usage. They also likely lack the C-level pressure mid-sized companies face to adopt and show AI returns as a strategic initiative.

Larger organizations may have offloaded AI to centralized AIOps teams, making it less of an immediate imperative for AI collaboration to be embedded in the editor itself. Mid-sized organizations hit that sweet spot of experimentation, resource availability, and directives from leadership that drive AI tools into existing applications.

Together, these findings suggest that collaboration maturity—and particularly AI readiness—follows a predictable curve. As companies grow, collaboration shifts from convenience to necessity, and the rich text editor becomes the platform where that evolution plays out.

Chapter 3:

The How and Why of Implementing RTEs
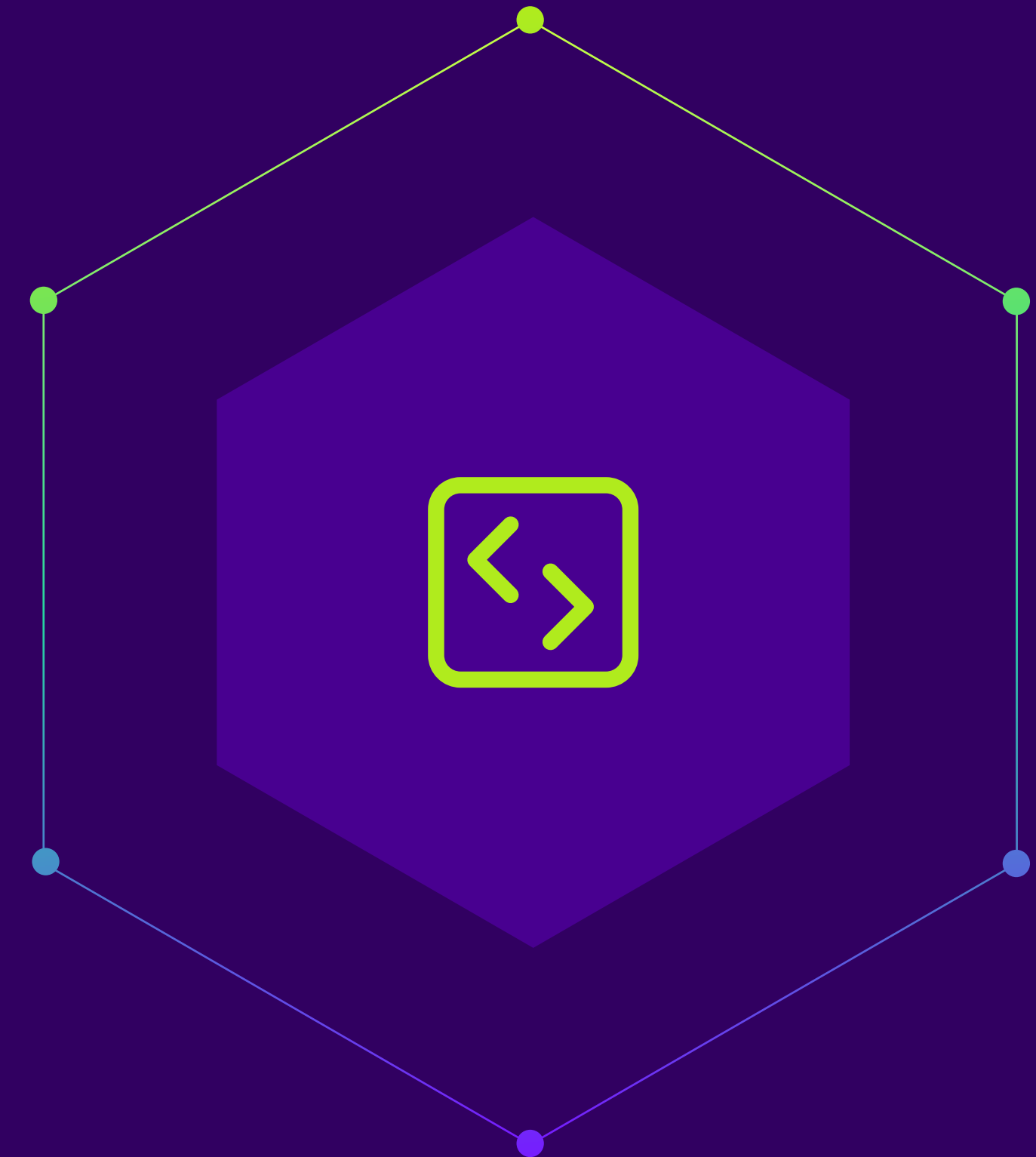
Implementation as a strategic choice

How you implement your RTE is a strategic choice as much as a technical one. Choosing to buy a best-of-breed RTE component from a vendor carries a lot of benefits, such as saving your team from having to build the basics.

Yet, many still prefer to develop in-house. This involves taking on additional development work and committing time and resources to the issue. Often, these teams have to develop features that are deceptively complex. Something as straightforward as list layouts quickly gets challenging to develop, let alone building collaboration features that pass data securely between servers and clients.

Additionally, teams must decide where and how to deploy. Some organizations prefer to keep their data in the cloud, deploying a pure-play SaaS RTE component and offloading a lot of the management stress to a third-party vendor. Others choose on-prem for elements like greater control, visibility, governance, or flexibility. And a growing number have switched to hybrid deployments to get the advantages of both.

These decisions offer a glimpse into how collaboration itself unfolds inside an organization. A well-implemented RTE is more than a text editor: It's the foundation for how people create, communicate, and share information. In short, implementation choices shape not only how your editor performs, but how effectively your people—and increasingly, your AI collaborators—work together.

Buy or build?

Rich text editors are complex beasts to create. What appears straightforward—like generating line formatting or building a table—requires sophisticated code and cross-browser and device testing. Yet, many still build their own rich text editors, with 30% of respondents choosing to build from scratch. However, 60% of respondents prefer to buy an RTE component to add into their applications, minimizing the development work.

Fig. 9: For work projects that require a rich text editor, do you usually buy a ready-made editor component or build your own in-house?

- Buy and work with a ready-made editor component
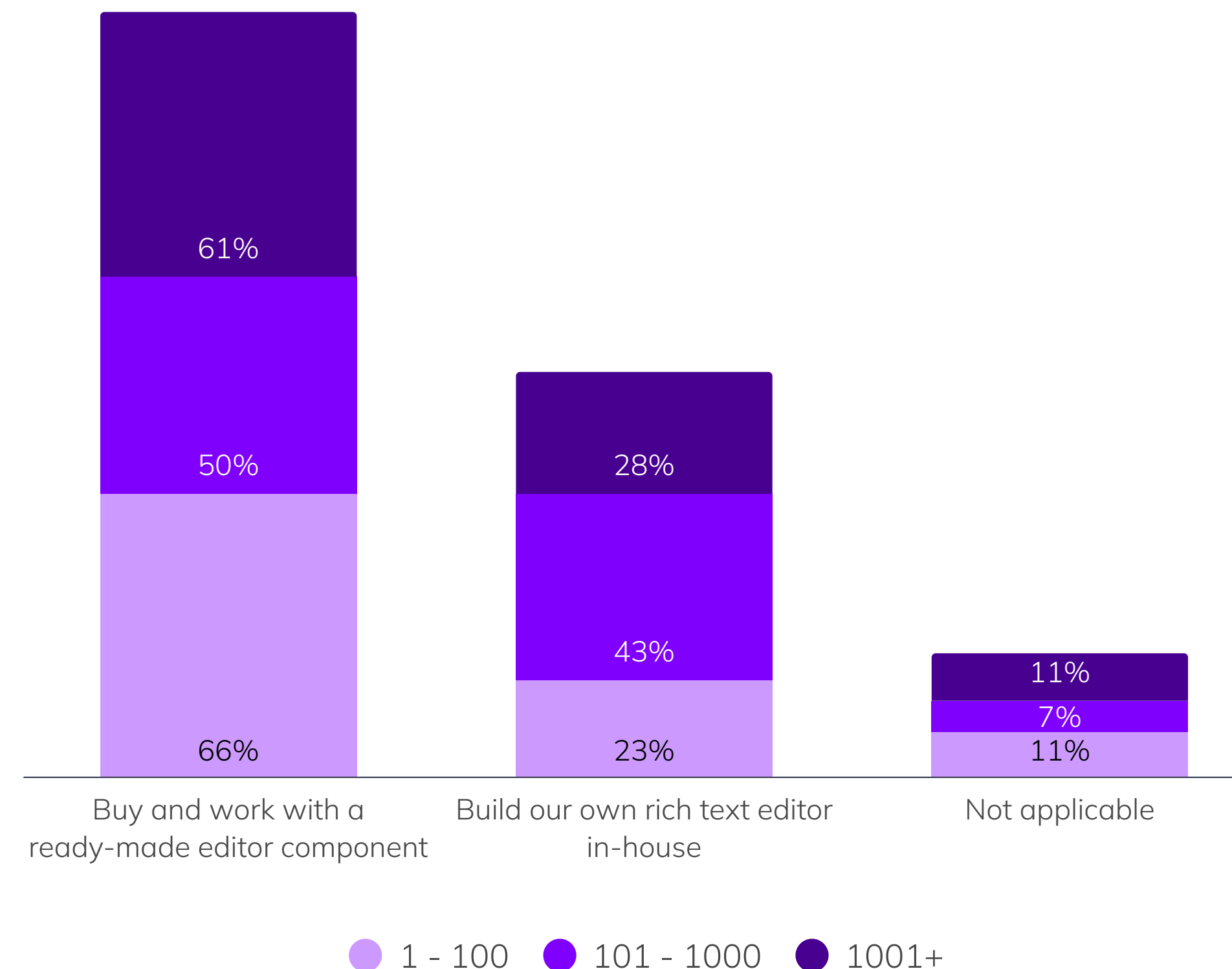- Build our own rich text editor in-house
- Not applicable
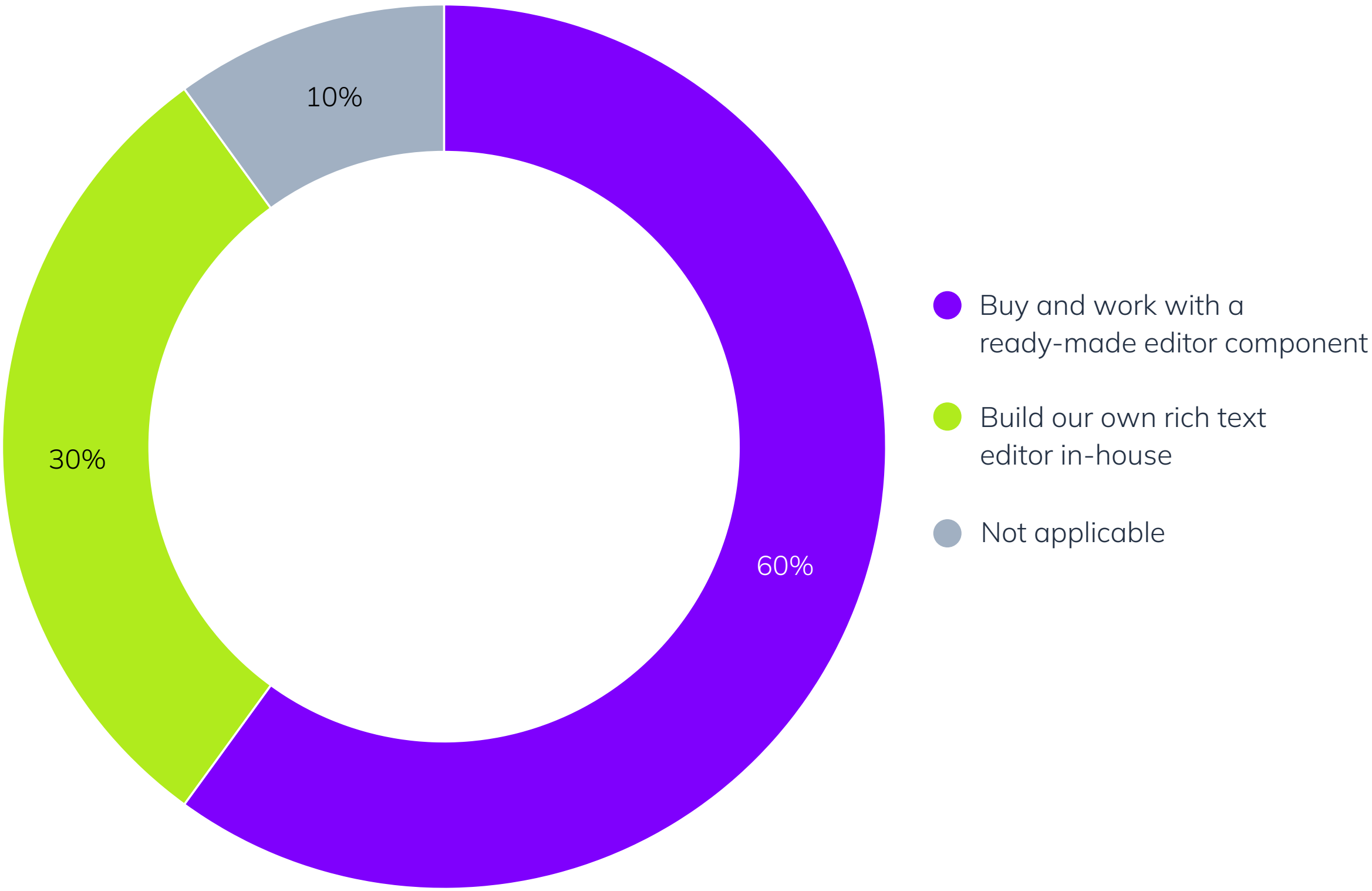
Breaking it down by company size, an interesting pattern emerges. Sixty-six percent of smaller companies (1 – 100 employees) and 61% of large companies (more than 1,000 employees) choose to buy a ready-made editor component.

For smaller organizations, this makes sense: ease of setup and creation are likely drivers, with a simple, extensible RTE reducing custom development work. This is critical for time-poor teams that lack resources for extensive development.

Larger companies may also face time constraints, with engineers deployed on other critical projects and lacking the availability to develop a new editor from scratch. Additionally, they may have more standardized processes and tech stacks, making it critical to just find something that already integrates with their technology.

The number dips for mid-sized companies, but still, about half choose to buy.

Fig. 10: 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?

Legend: 1 - 100, 101 - 1000, 1001+

Buy and work with a ready-made editor component: 66% (1-100), 50% (101-1000), 61% (1001+)
Build our own rich text editor in-house: 23% (1-100), 43% (101-1000), 28% (1001+)
Not applicable: 11% (1-100), 7% (101-1000), 11% (1001+)

The How and Why
of Implementing RTEs

Why build?

Across all company sizes, roughly 30% build their own RTEs in-house. Building from the ground up places heavy strains on development teams and resources, so why put in that time and effort?
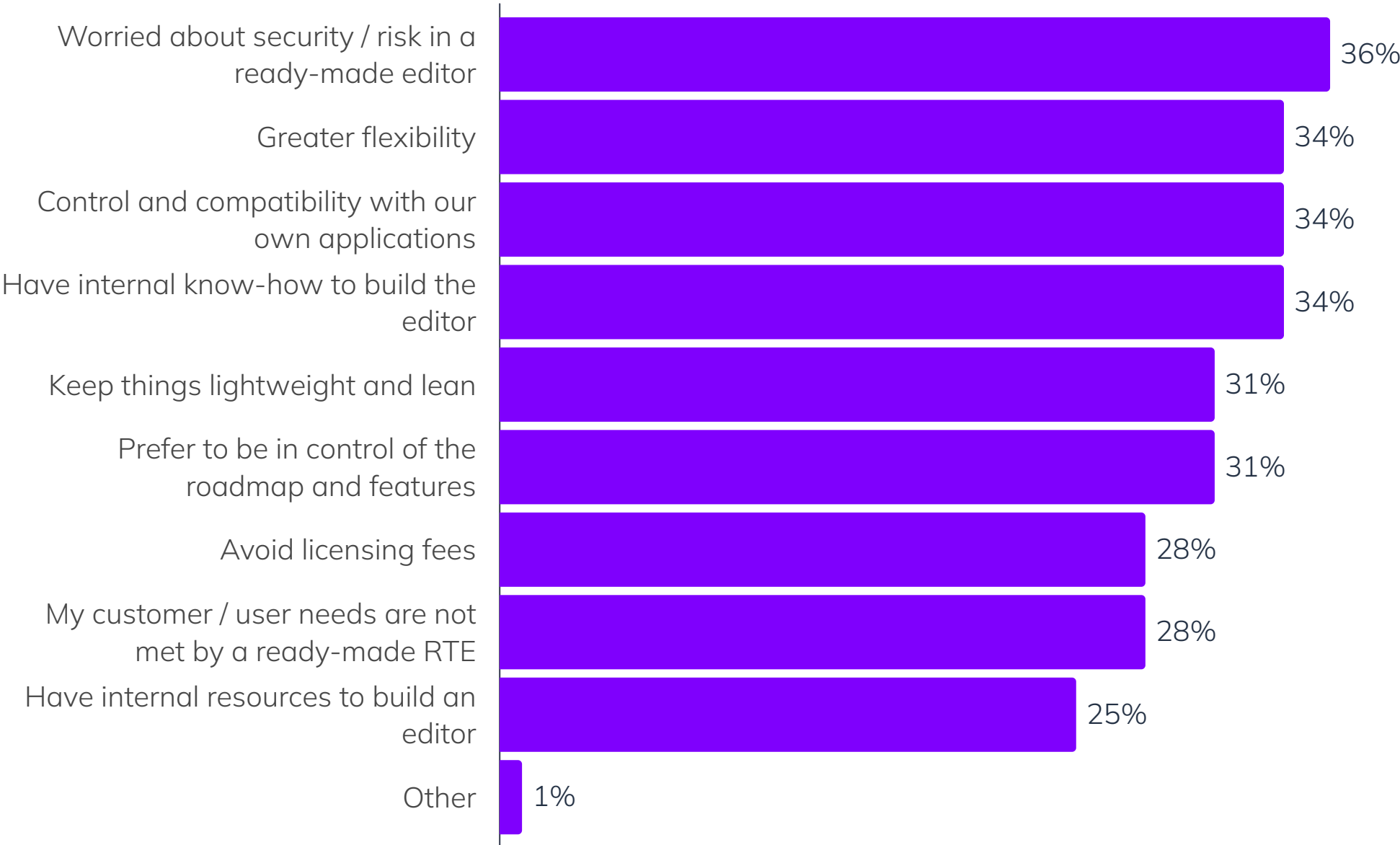
Leading the list was the worry about the security of a ready-made editor at 36%. Next, the desire for greater flexibility came in at 34%, matching the need for full control and compatibility with applications, also at 34%. Another 31% wanted to fully drive their own feature roadmap. All these point to the desire for ownership and control as a core theme, as even security concerns require complete infrastructure governance to ameliorate.

Also noteworthy, 34% said that their requirements were not met by a ready-made editor. This could come down to a mismatch with existing infrastructure, a lack of compliance certifications, or even feature deficits. However, these requirements might overlap with other answers as well, such as a need for security or keeping editors light and lean (which came in at 31%).

It's worth inspecting concerns about security more closely, especially through the lens of collaboration. This year's results speak to

a reprioritization on the topic of security, especially when compared to 2024's responses, which put security as a top driver to build at 22% (compared to 36% for this year). This likely stems from growing compliance burdens, increases in the types and numbers of cyberthreats, and AI scaling cyberattacks.

Fig. 11: What drives your choice to build your own editor in-house instead of buying a ready-made editor?

Category	Percentage
Worried about security / risk in a ready-made editor	36%
Greater flexibility	34%
Control and compatibility with our own applications	34%
Have internal know-how to build the editor	34%
Keep things lightweight and lean	31%
Prefer to be in control of the roadmap and features	31%
Avoid licensing fees	28%
My customer / user needs are not met by a ready-made RTE	28%
Have internal resources to build an editor	25%
Other	1%

Collaboration requires greater security. Each additional person working on a document adds complexity, which leads to more opportunities for malicious cyberactors to strike. Cyberattackers could compromise user accounts, exploit authorization gaps, or access sensitive customer data due to bugs in tenant isolation or link sharing.

This just scratches the surface. The bottom line is this: The more people working on a document, the more chances attackers have to gain access to sensitive data. Having strong security controls in place makes collaboration possible.
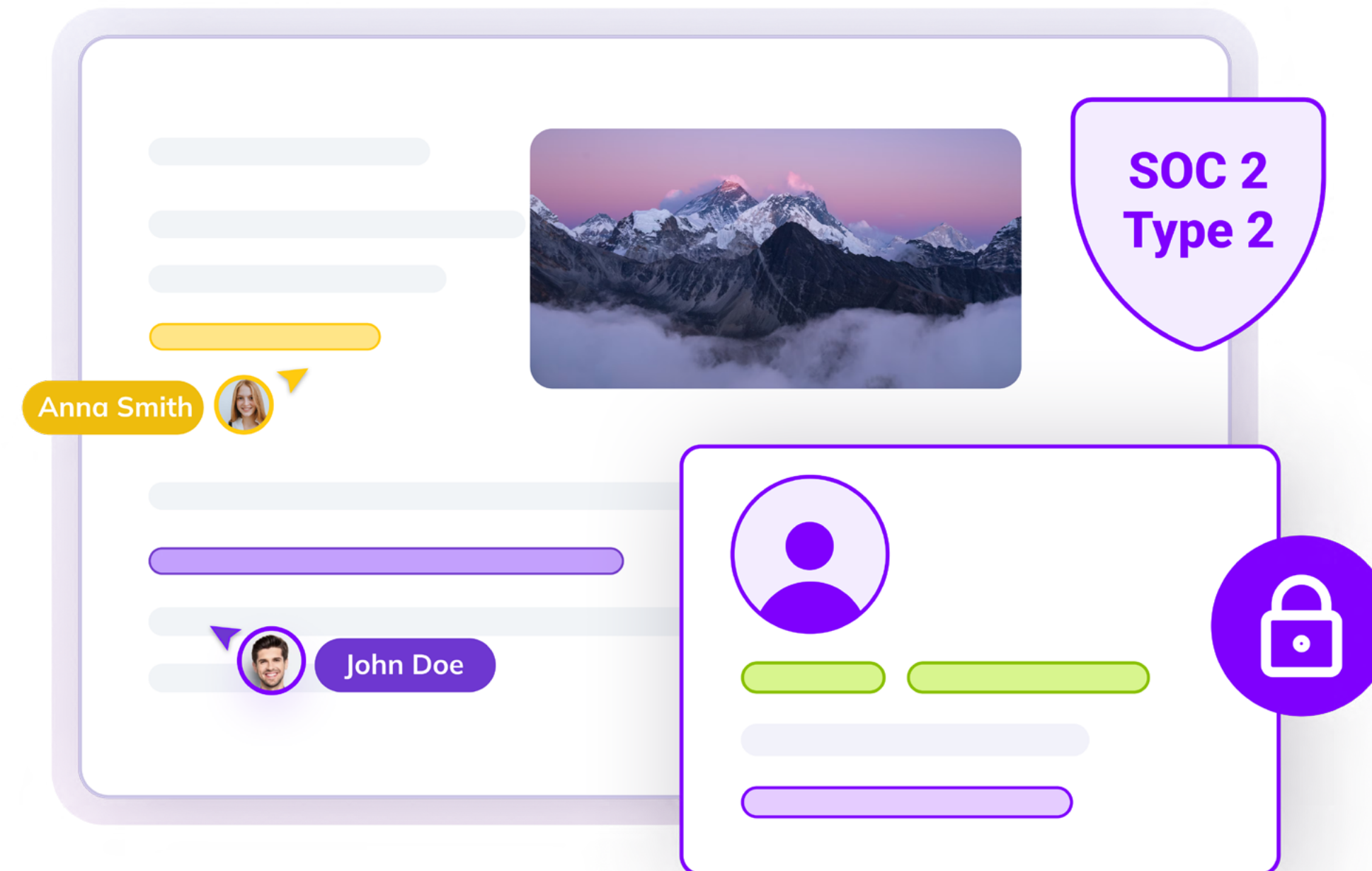
The burdens of building

Despite the reasons organizations choose to build, it's not all smooth sailing. Organizations that build their own editor face a slew of challenges that make the process more complex and challenging than expected. For the most part, organizations rated RTEs as harder to implement than expected. This result played out in other key challenges, such as going over project timelines, difficulties integrating with other components, never meeting the initial requirements, and the costs ballooning beyond initial expectations.

Collaboration makes building in-house even more challenging. Strong role-based access controls (RBAC) and permissions are required, and these must integrate cleanly with existing systems. Entire companies focus exclusively on user access management, which illustrates how demanding building and maintaining these systems can be.

Fig. 12: What was the biggest challenge you faced when building your own editor in-house?

Challenge	Percentage
Took longer than expected	23%
Cost more than expected	19%
Harder to implement than anticipated	17%
Pulled engineers from other projects	14%
Couldn't integrate with other components	13%
Never fully met original requirements	12%
Nothing, our build went off without issue	2%
Other	1%
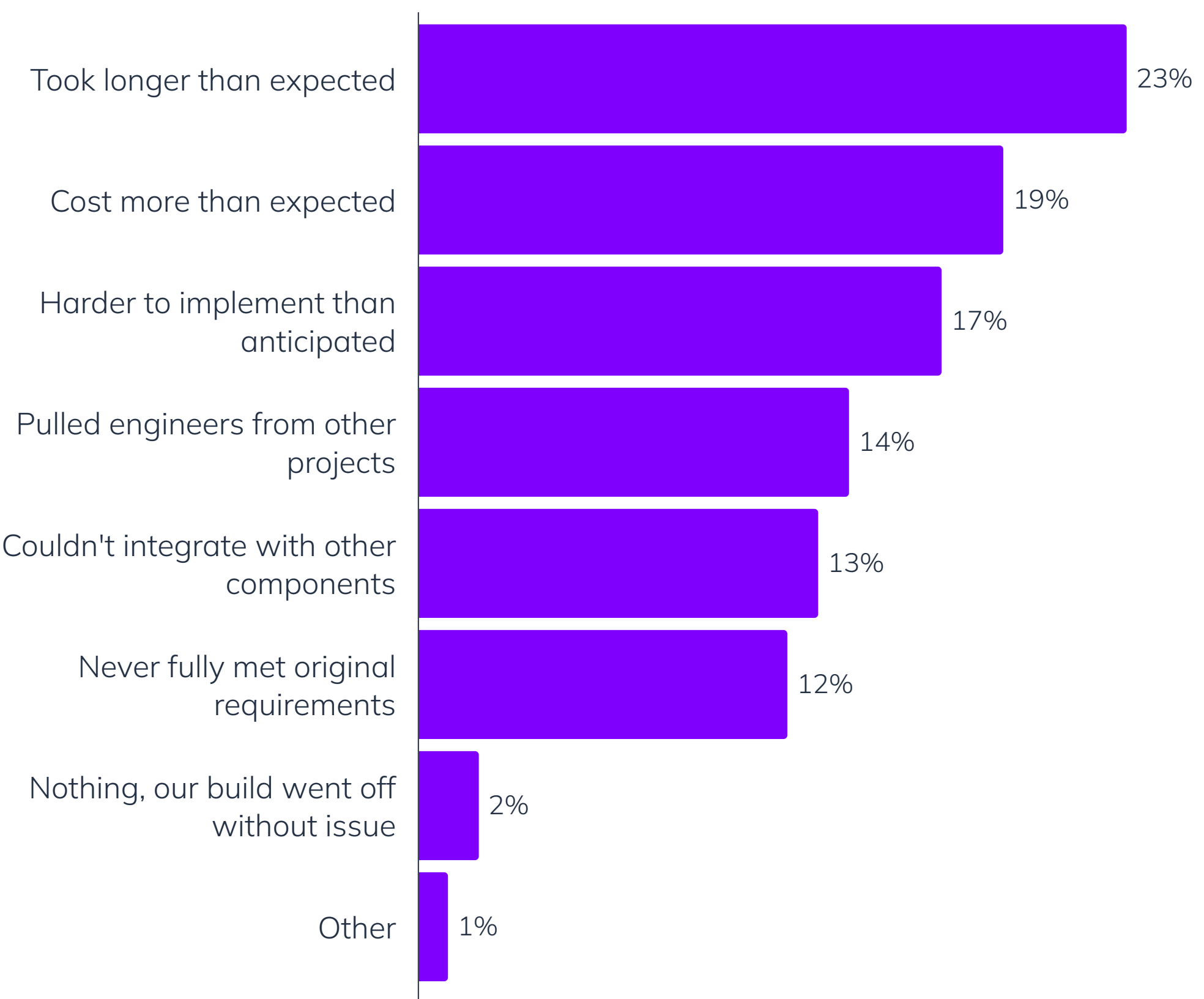
An organization committed to building needs to account for collision management. Their team will need to choose between locking parts of a document to prevent conflicts or implementing more advanced logic to support real-time collaboration.

Beyond that, features such as revision history, track changes, and suggested edits add even more development and maintenance overhead. Together, these components expand the complexity and long-term cost of maintaining an in-house editor.

All these challenges divert critical engineering resources from other tasks. For builders, these present opportunity costs where engineers could work on other value-add capabilities, instead of building what is, in reality, a completely new product. With all these issues, it's surprising that 30% choose to build in-house.
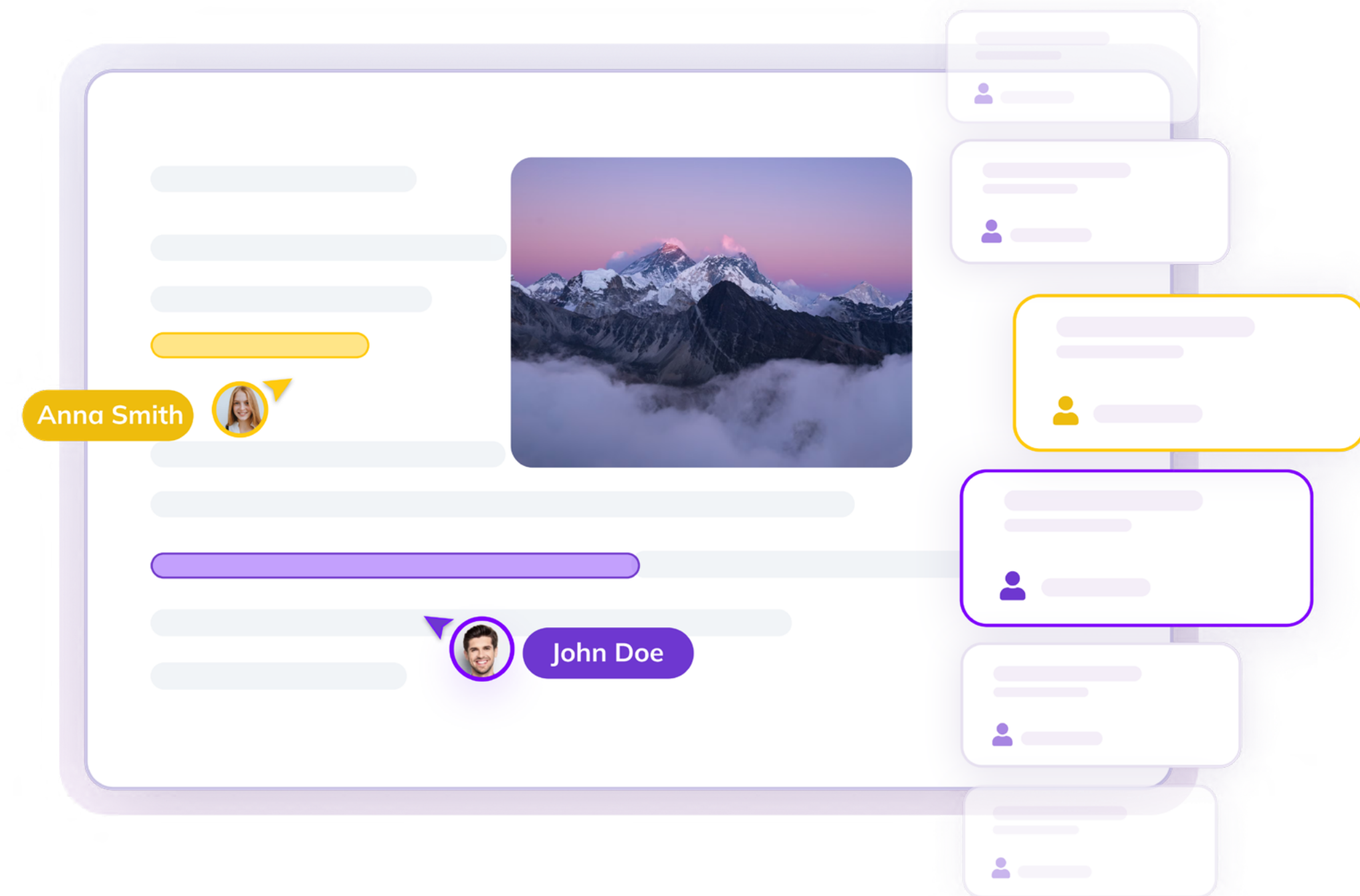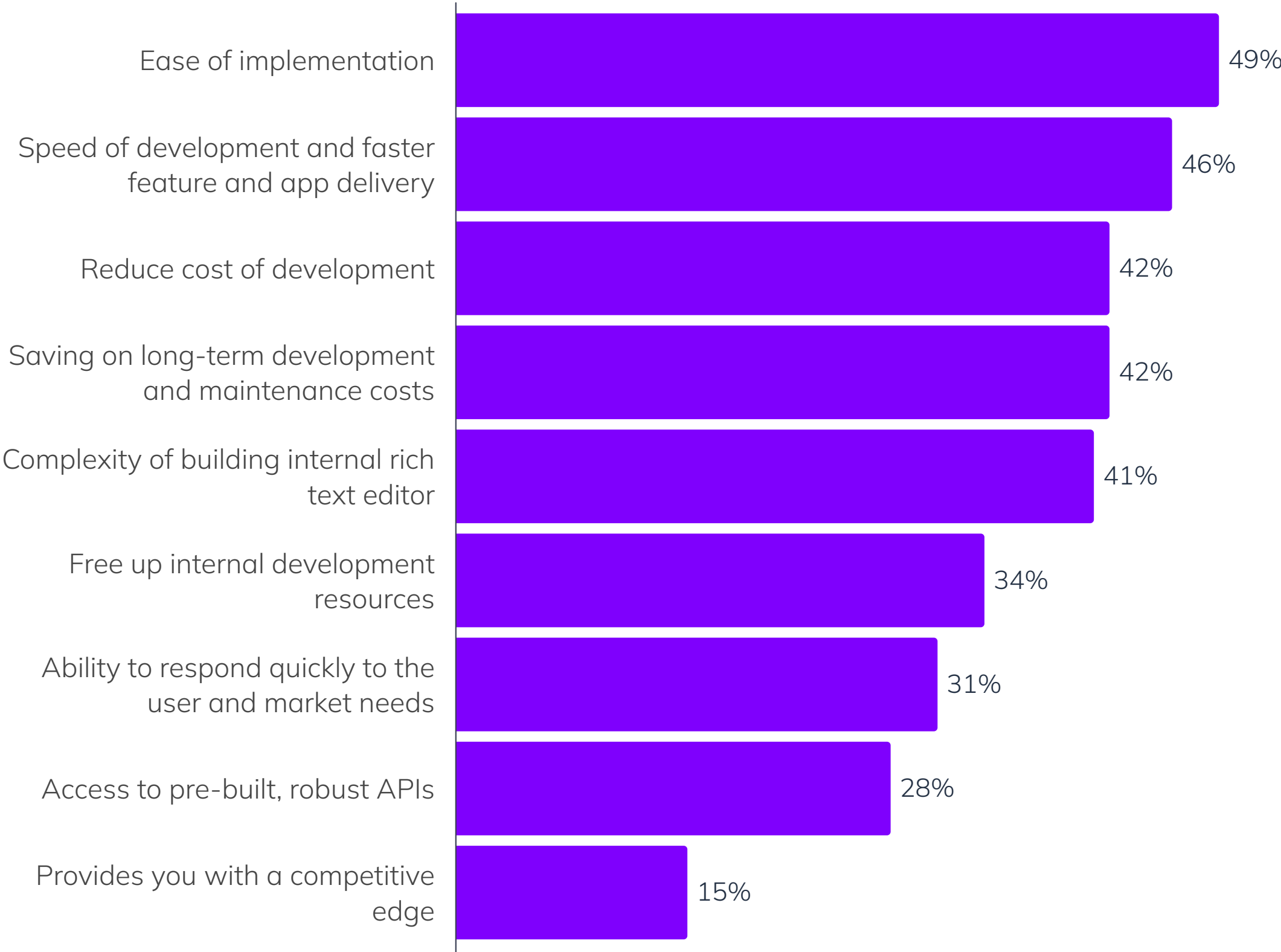
What about buying?

Twice as many respondents decided to buy a pre-made, composable rich text editor component. Well-designed RTEs offer deep customization that rivals building in-house at a fraction of the development overhead. The right RTEs can offer a bespoke experience, allowing you to pick and choose which features or plug-ins to add while offering the ability to extend the RTE with your own development work. In short, buying doesn't mean sacrificing control—these tools are usually built with developers in mind, all but eliminating implementation and interface constraints.

What did the data show? When asked about the reasons for buying a ready-made editor, ease of implementation was the top response at 49%, while avoiding the complexity of building an internal editor sat at 41%. Several answers centered on speed, with 46% of survey respondents citing fast implementation as a top reason and 31% claiming it helped them respond to market needs faster. Finally, financial choices play a role, with lowered upfront development and ongoing maintenance costs coming in at 42%

Fig. 13: What drives your choice to use a ready-made editor instead of building your own?

Category	Percentage
Ease of implementation	49%
Speed of development and faster feature and app delivery	46%
Reduce cost of development	42%
Saving on long-term development and maintenance costs	42%
Complexity of building internal rich text editor	41%
Free up internal development resources	34%
Ability to respond quickly to the user and market needs	31%
Access to pre-built, robust APIs	28%
Provides you with a competitive edge	15%

We also noted a connection with the organizational maturity curve. As organizations grow, they often recognize that the core functionality of RTEs is best left to experts developing best-of-breed solutions. Vendors can spend time keeping up with the latest features, especially frequent developments in AI, leaving your engineering teams to focus on creating new, value-added functionality in your applications.

But it's critical to reiterate that even if you offload some of the initial development work to third-party developers, some RTE components are highly extensible. Developers can focus on putting the finishing touches on a strong user experience and modify the editor as they see fit. Buying isn't just a faster route; it can be a strategic one that compounds over time.
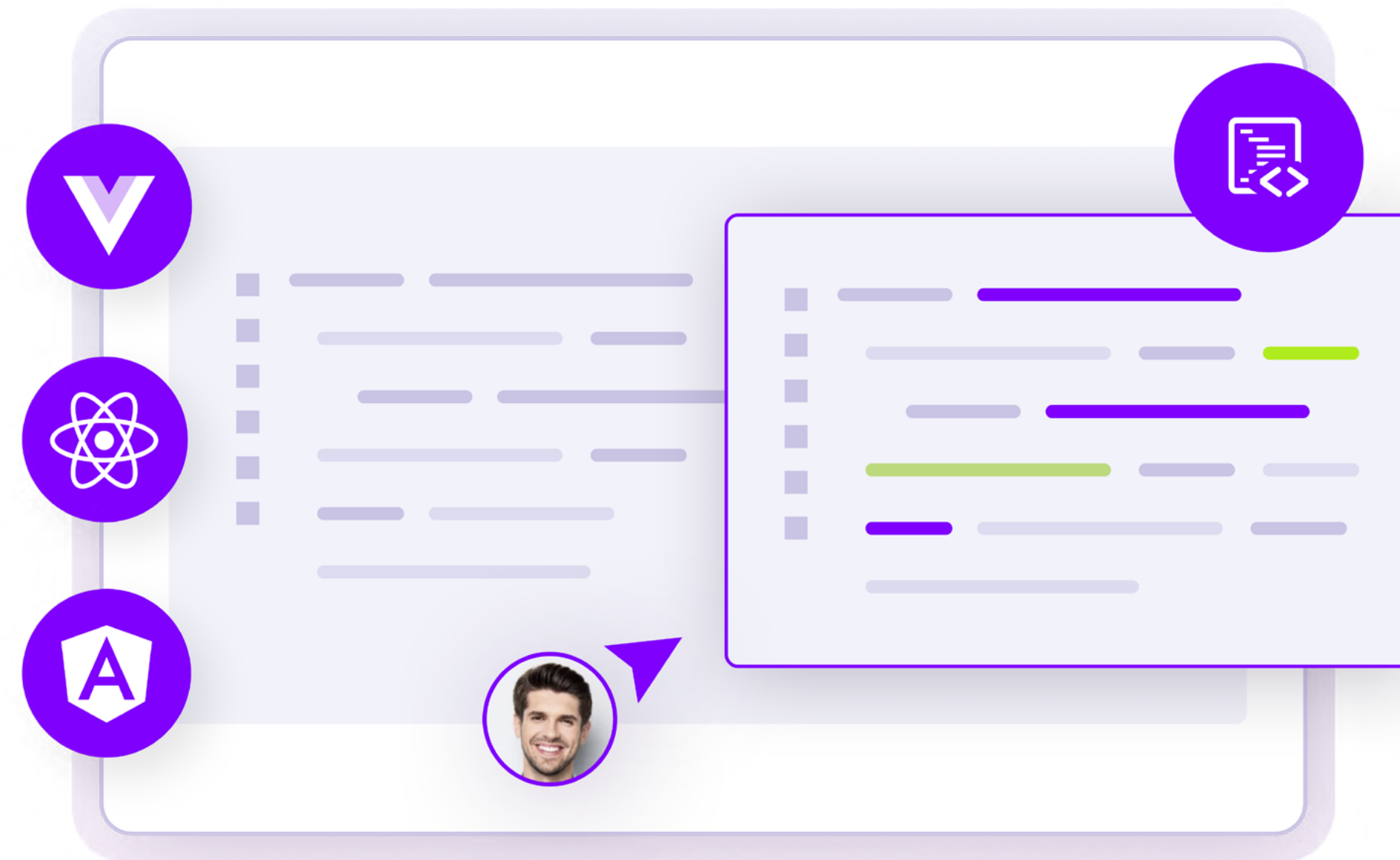
Deployment: Choosing between cloud, on-prem, and hybrid

CKEditor

How you deploy your RTE matters. It can make a difference in labor costs, maintenance, control, and time-to-market. This year, the choice between cloud-hosted and self-hosted gave an edge to self-hosted (38%) over cloud (30%).

When looking at the top reasons for choosing cloud deployment, most respondents pointed to ease of use. Twenty-five percent emphasized the need for software that is available everywhere, which aligns with the growth of distributed teams. Plus, the simplicity of the cloud undergirded other responses, with 18% citing automatic deployments, 17% noting the reduced workload, and 14% claiming scalability made them choose cloud deployment.

By contrast, self-hosting offers greater control. The data about rising security concerns suggests a strong link. For organizations in regulated sectors or those handling sensitive data, maintaining full oversight of infrastructure remains non-negotiable.

Fig. 14: What is your preferred deployment option for your rich text editor?

- Self-hosted
- Cloud
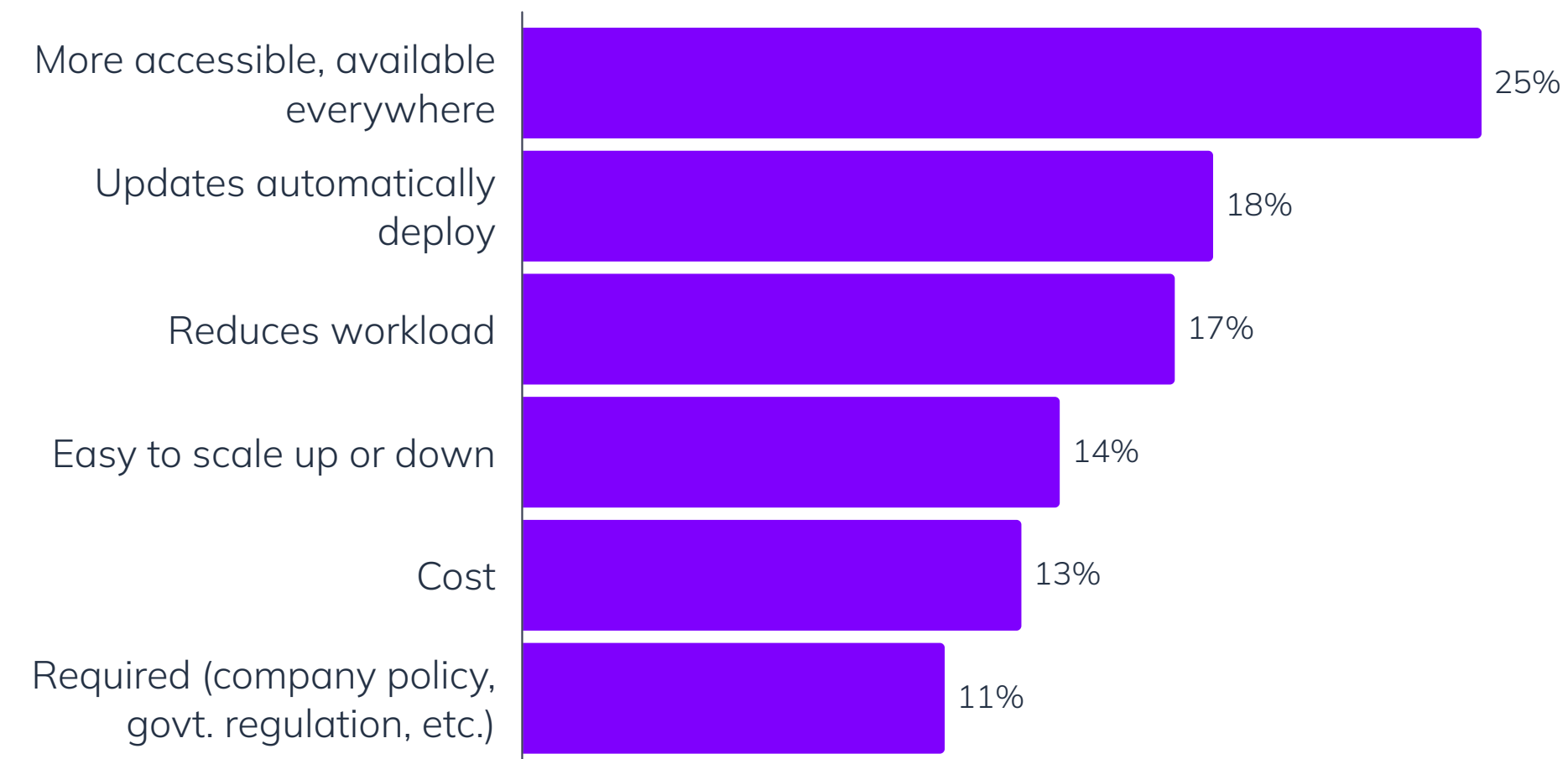- Hybrid
- Not sure

30%
26%
38%
6%
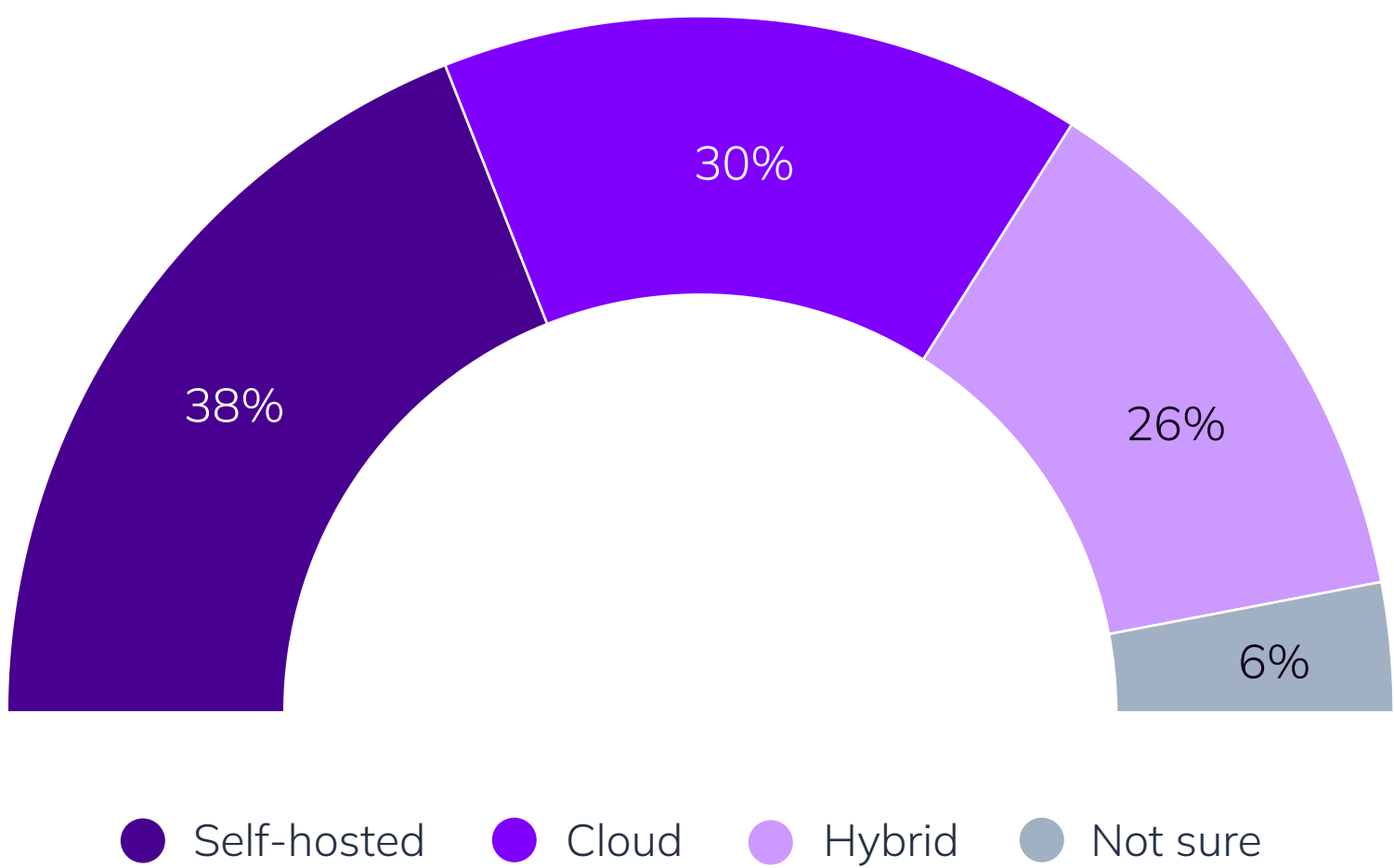
The most notable trend, however, is the growth of hybrid deployments, which rose to 26% of respondents—up from 16% in 2023 and 18% in 2024. Hybrid offers a balance, keeping sensitive data on-premises while still maintaining cloud scalability and agility.

How does this apply to collaboration? Collaboration depends heavily on the infrastructure that supports it. Real-time co-authoring, shared version histories, and AI-assisted editing all rely on low-latency connectivity and continuous access to shared data. Cloud deployments enable this easily, offering instant availability for distributed teams.

Self-hosting can match that functionality, and in some cases, may help control costs in the long term over cloud-based RTEs with usage-based models. However, it does require internal investment in networking and synchronization infrastructure. And with globally distributed teams, it often requires colocating data centers near each office location (or client locations if you're selling a commercial app). If developing apps with a broad user base, there may also be regional data residency requirements, especially with collaboration features that require user registration.

The bottom line is this: Deployment choice reflects organizational priorities. Organizations prioritizing speed or scalability choose cloud, those needing more control opt for on-prem, while many increasingly choose hybrid setups to gain the best of both worlds.

Fig. 15: Why do you prefer to use cloud services over self-hosting your rich text editor?

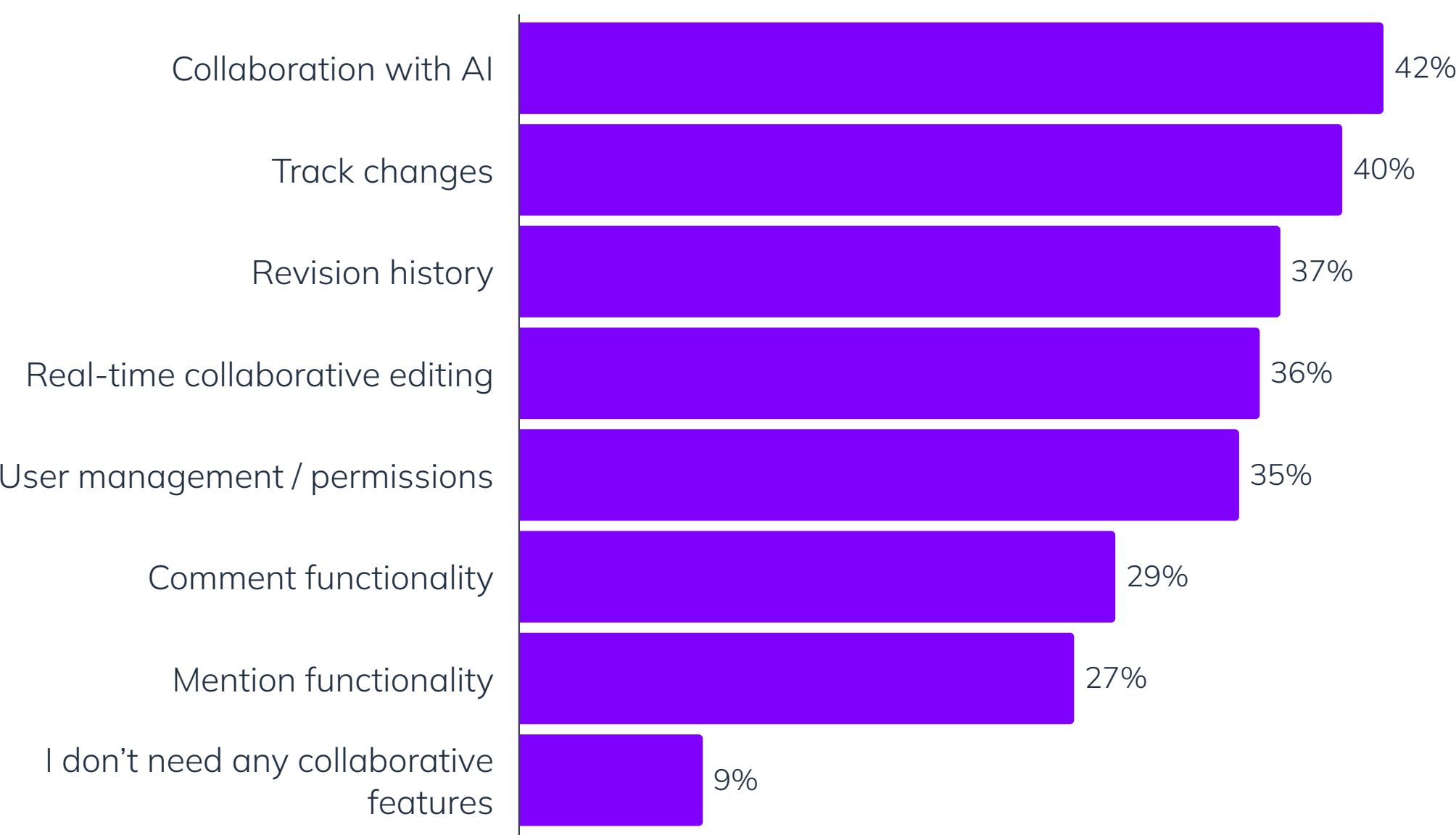| Category	%
More accessible, available everywhere	25%
Updates automatically deploy	18%
Reduces workload	17%
Easy to scale up or down	14%
Cost	13%
Required (company policy, govt. regulation, etc.)	11%

Chapter 4:

The Future of Collaborative Editing

Where is collaborative editing headed?

Change is one of the only constants in life. Although, lately, you can add the law of accelerating change to the list of constants. Respondents were asked to predict the most important features over the next five years for collaborative editing, and while many answers were expected, others were surprising.

The second and third most important collaborative editing features were track changes (40%) and revision history (37%), which makes sense since they're core to collaboration. Real-time collaboration came in next at 36%, demonstrating the need to respond rapidly to project tasks and goals. These numbers are each fairly close to previous years' responses, showing their continued importance.

But two new trends demonstrate where we're headed: the top answer was collaboration with AI at 42%, while respondents answered with user management and permissions at 35%. This indicates an increasing emphasis on security features. These two trends are highly significant, so they deserve their own discussion.

Fig. 16: In the next five years, which of the following collaborative editing features will be critical?

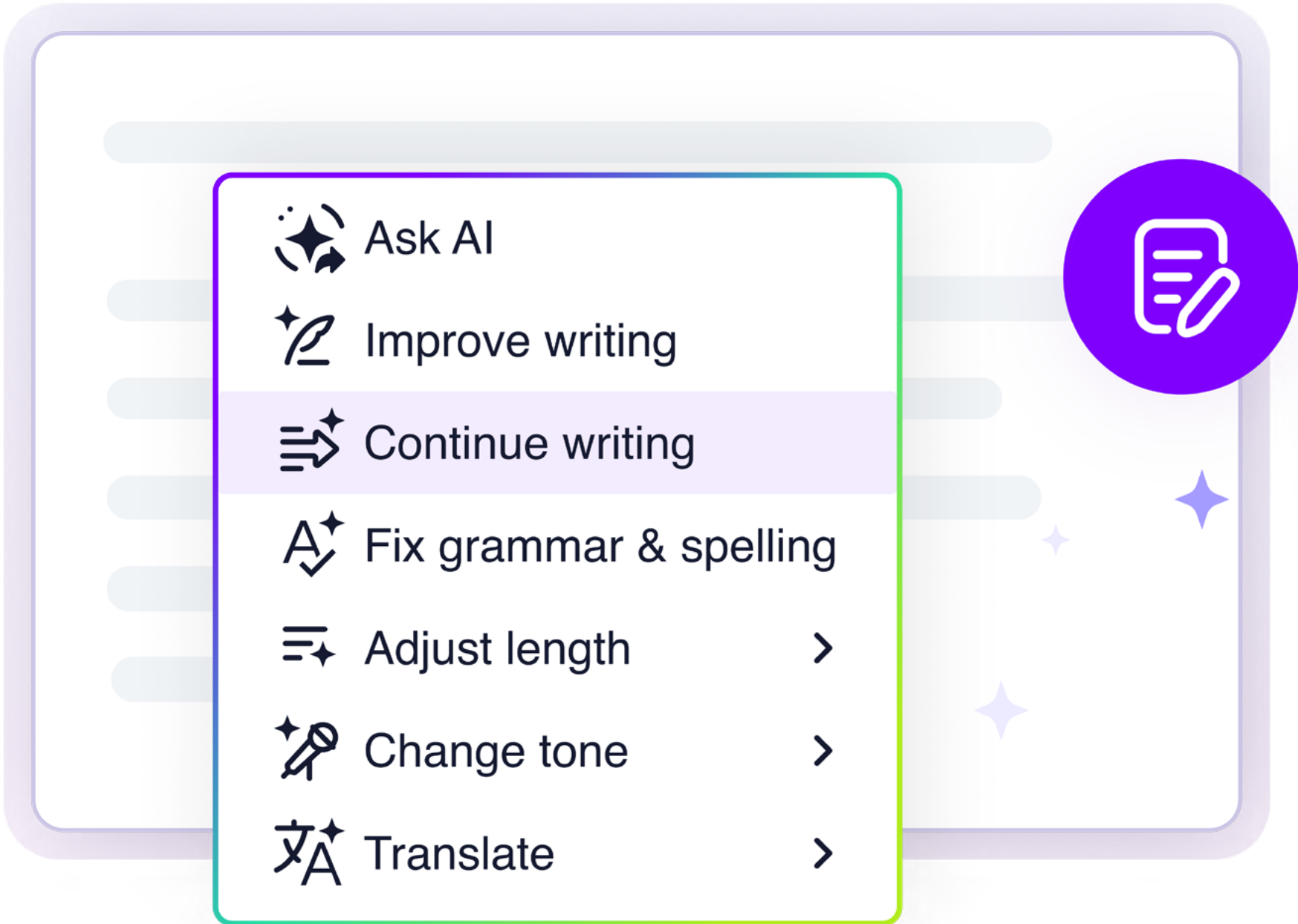| Feature	%
Collaboration with AI	42%
Track changes	40%
Revision history	37%
Real-time collaborative editing	36%
User management / permissions	35%
Comment functionality	29%
Mention functionality	27%
I don't need any collaborative features	9%

The Future of
Collaborative Editing

The role of AI and LLMs

AI has become more deeply embedded into every aspect of our lives, integrating into everything from smartphones, customer service chats, and productivity tools and crisscrossing industries from government to financial services to healthcare and more. So, it's no wonder that AI was touted as one of the biggest predicted shifts in the RTE market over the next few years.

Respondents predicted AI collaboration as the top AI trend overall. This represents a major figure-ground shift in our relationships with LLMs and AIs in organizations of all stripes. With teams stretched thin and asked to do more with less, AI often helps bridge the gap.

Content creation is not easy. Staring at a blank page stricken with writer's block or trying to review your own content can slow down your progress or force errors. Turning AI into a true collaborator means workers can have it generate outlines, suggest places to improve content, do research, or edit text and suggest new wording. AI can finally function as a sidekick, supplementing skillsets and offloading lengthy tasks for users.

Ask AI
Improve writing
Continue writing
Fix grammar & spelling
Adjust length
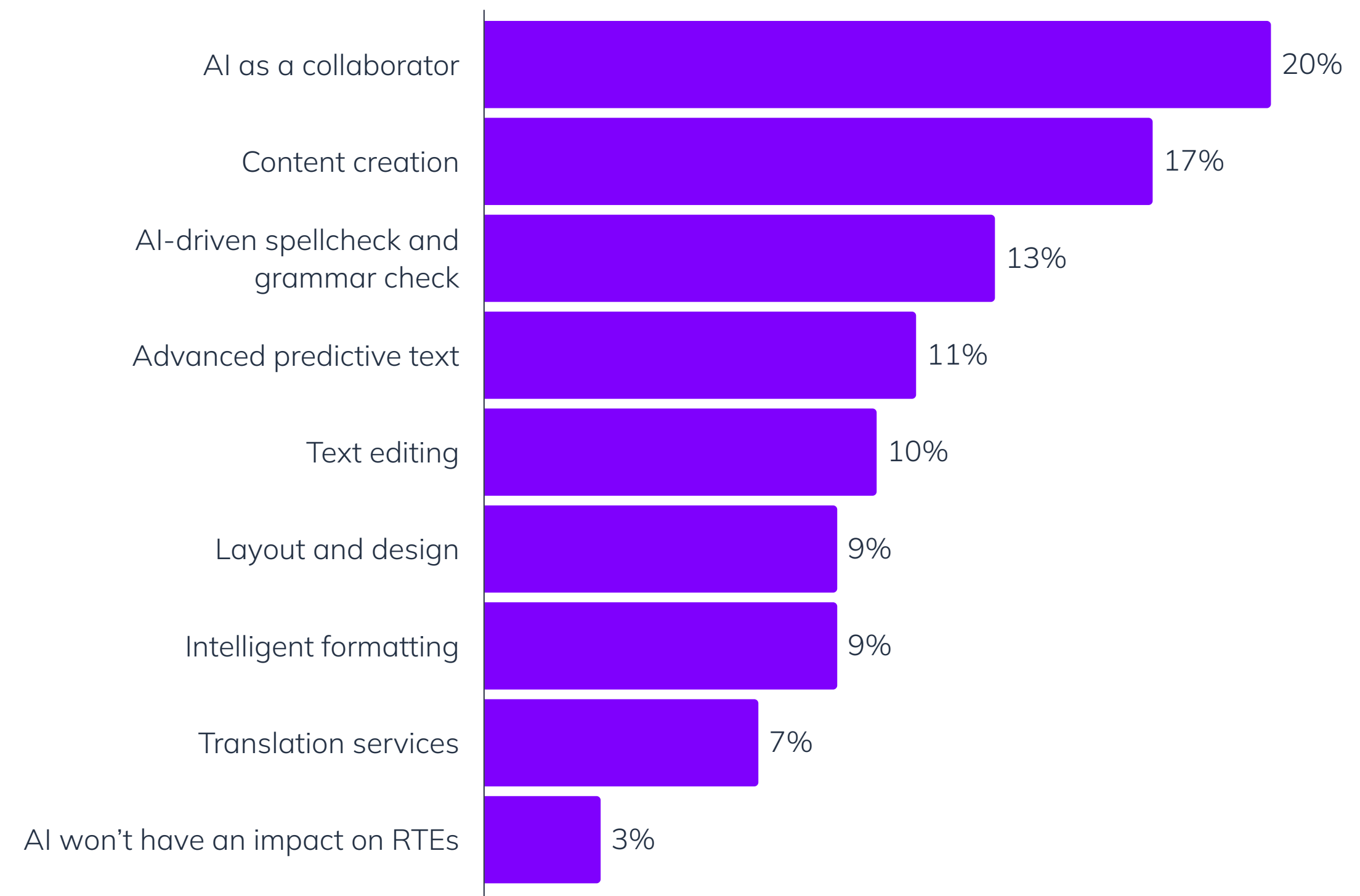Change tone
Translate

The Future of Collaborative Editing

We dug in a bit further, asking participants where they thought AI and LLMs would make the biggest difference. Answers encompassed content-related activities like content creation (17%), spellcheck and grammar (13%), text editing (10%), and translations (7%). Graphic elements came in at 9% for both layout and design and intelligent formatting.

But still, the highest-rated answer was AI as a collaborator at 20%, showing that organizations are thinking of AI as an extension of their team. The truth is that collaboration is a non-linear process, and this applies to AI as well. Users may put AI to work by brainstorming, creating a short or long draft, reviewing content, or researching, and, critically, they may zig-zag between these tasks to accomplish their goals.

Companies should be careful here. AI can be an excellent collaborator, but too many users have to flip between AI tools to finish content. This context switching drains some of the productivity gains offered by AI and increases potential mistakes. To combat this, more organizations are turning toward embedded AI that includes all the AI tools needed within the editor. Expect this to become more of the norm over time, as consolidation boosts efficiency and makes for a more pleasant user experience.

Fig. 17: Where do you think LLMs and AI will have the biggest impact on rich text editors in the next year?

| Category	%
AI as a collaborator	20%
Content creation	17%
AI-driven spellcheck and grammar check	13%
Advanced predictive text	11%
Text editing	10%
Layout and design	9%
Intelligent formatting	9%
Translation services	7%
AI won't have an impact on RTEs	3%

Security's importance has increased

Security has shifted from a compliance checkbox to a deciding factor in choosing collaborative editing tools. This year, 36% of organizations building their own RTEs cited security or risk in ready-made editors as their top concern—up from 22% last year.

Why has security become so critical? One potential reason is that complexity expands the attack surface. Many companies use multiple editors across departments, creating inconsistencies in access controls and data handling. Each solution introduces potential vulnerabilities, and respondents now rank user management and permissions among the most critical collaboration features.

AI heightens the challenge. AI assistants increase productivity but can expose sensitive data or create new risks through prompt injection or data leakage. At the same time, AI-driven attacks are cheaper and more easily automated, pushing teams to tighten governance and demand stronger security credentials from RTE vendors.

These concerns are reshaping deployment choices, too. The steady rise in hybrid deployments reflects the balance organizations seek between cloud convenience and on-premise control. In short, collaboration requires trust, and security builds that necessary foundation.
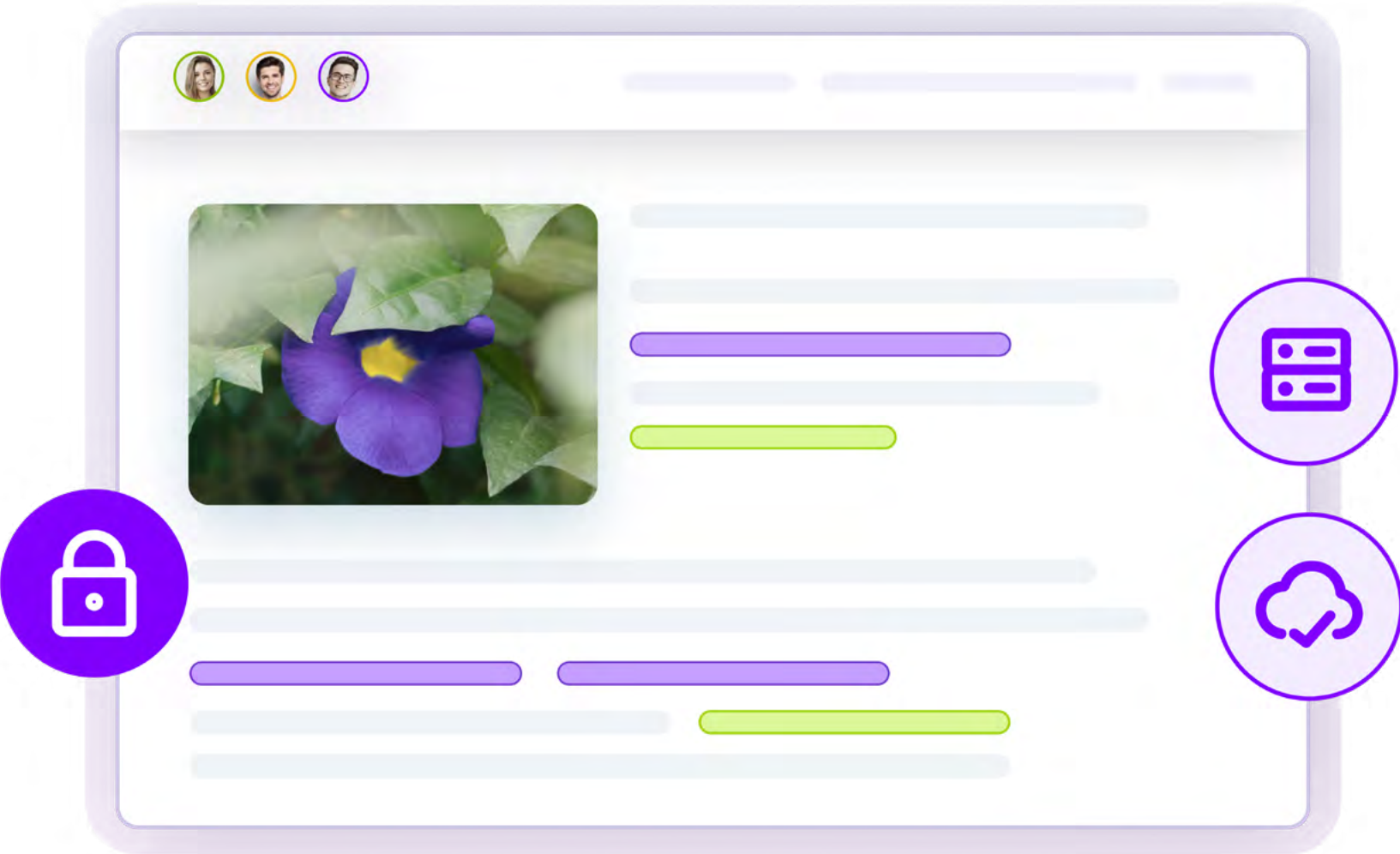
User predictions

Respondents also offered their own insights into the future of both rich text editors and collaborative editing. A few trends clearly emerged from their comments. The top theme, as just discussed, was the role of AI and LLMs. Here are a few of the other trends that rose to the surface:

RTE as an ecosystem
Rich text editors—and collaboration by extension—no longer exist only at the app level. Instead, we're seeing people refer to them as part of a broader universe of tools that connect to their infrastructure and users' actual day-to-day workflows.

One respondent stated we should see:

> Out of the box integrations with major frameworks.
> No more hacky custom solutions to upgrade or replace
> old systems.

This shows the emphasis on blending these solutions into the wider technical landscape of a given organization.

Another respondent claimed:

> RTEs will integrate seamlessly into entire workflows
> instead of being isolated tools.

This demonstrates not only the need for easy technical connectivity but also the ability to sit firmly within the natural daily workstreams of end users.

Nothing happens in a vacuum, as collaboration occurs not only at the app level but across entire workstreams and lifecycles. Collaboration needs to work within a web of technology in an organization.

User predictions

CKEditor

Multi-device support

Collaboration should be available to anyone at any time. To that end, we'll likely see a growth of multi-device support for teams. One respondent mentioned:

> It might be possible to achieve seamless multi-device synchronization using any device for smooth editing.

Shifting RTE support to fit across browsers, devices, and screen sizes allows users to edit and respond to content whether they're at their desk or scrolling on their phones at the airport. This further collapses review cycle time, bolsters productivity, and speeds up output.

Expansion of advanced collaboration tools

Since the data showed renewed enthusiasm surrounding collaboration in general, it shouldn't come as a surprise that several respondents spoke about changes in collaboration tools themselves.

These reflect a shift towards more advanced, specialized functionality, with one respondent claiming:

> I believe collaboration features will include more interactive elements.

This interactivity could mean that collaboration will become more action oriented in the editor. Features won't just capture feedback, they'll enable live interaction directly in the content.

Another respondent mentioned:

> Collaboration will be asynchronous and borderless.

This backs up many of the other predictions in this section, showing that people want to be able to work where they are, on whatever device they have handy.

Whatever the answer, we expect to see collaboration grow more sophisticated, from AI-enhanced tools to seamless, multi-device transitions.

Conclusion
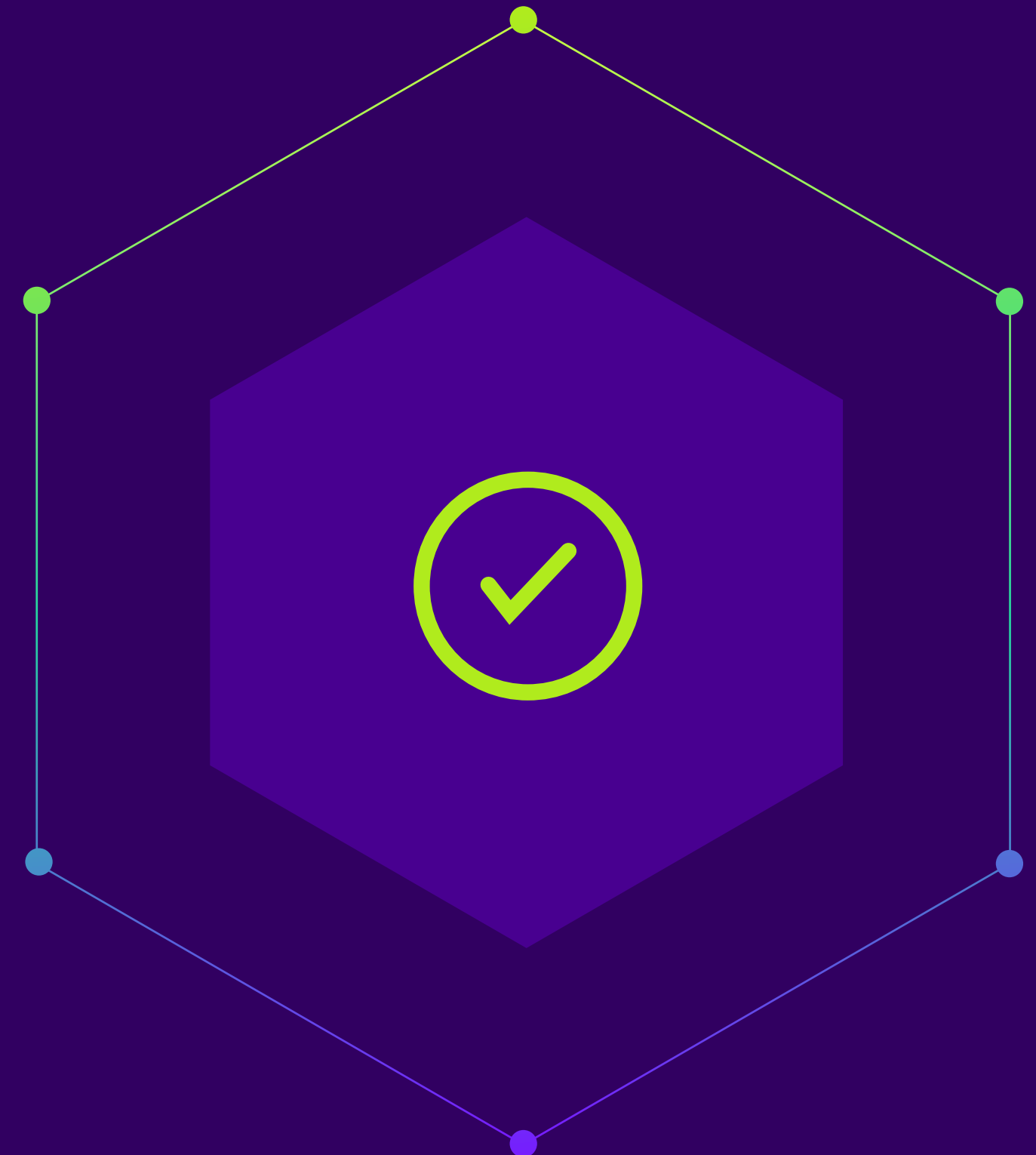
A new frontier in collaboration

The cornerstone of work is, and always has been, collaboration. As humans, we've succeeded when we worked together, whether that was early humans dividing tasks in a hunt, industrial workers dividing portions of their work on an assembly line, or modern workers drafting reports with data from multiple teams to guide future strategic decisions. When organizations succeed, they do so through the strength of their teams and their ability to work together.

This year, we saw a true renewal of collaboration. That renewal is directly connected to artificial intelligence. AI gained mainstream prominence only a few years ago, but now, after years of investments, we're starting to see the dividends emerge. AI has delivered a new boon to collaborative editing, turning from a technology tool to a full-blown collaborative partner and an extension of a team. This fundamentally changes the face of collaborative editing for the long haul.

As the contributions of humans and AI have blended into a partnership, rich text editors have become a meeting place. They've become central hubs for teams across time zones and devices. All of this redefines how teams write, review, and ship content, and they improve both speed and quality.

As collaboration continues to mature, users will further expect to sidestep disconnected workflows and demand more real-time collaboration, AI assistance, and streamlined experiences in their RTEs, all without sacrificing security. Teams will expect more control over where they deploy as well, whether that's cloud, self-hosted, or hybrid.

One thing is clear: we're entering a new phase of collaborative editing. Communication is the glue of every organization and workflow, making it imperative for teams to act now in embracing future-forward collaborative editing capabilities.

CKEditor