tiny

The State of Rich Text Editors

2022 Developer Survey Results

In the second year of The State of Rich Text Editors Survey, hundreds of developers shared their thoughts on how they use rich text editor platforms, their tools, integrations used and what's most important to them in their daily work.

- **FUNDAMENTALS MATTER THE MOST**, not fancy features
- **A GREAT USER EXPERIENCE** remains **THE KEY TO A WINNING** formula
- **UX** remains an important focus as the most frequent **RTE CHALLENGE**
- **INNOVATIVE FEATURES ARE LESS IMPORTANT** than user and developer-friendliness
- **MOST USERS ARE SETTLED IN THEIR RICH TEXT EDITOR CHOICE,** but fail to understand the available customization opportunities

It’s the fundamentals that matter

For a piece of tech that looks basic, a rich text editor (RTE) is incredibly complex. On the inside.

In fact, it’s one of the most complex interface components: both visually and technically.

An example is the ubiquitous ‘Enter’ key. In an average RTE, that single key performs over 100 different actions — depending on where you are in the editor interface, and what you’re doing. That doesn’t take into account any browser compatibility issues, or dependencies that you’ve built into your product, so the final tally could easily be double that number of base actions it performs.

Then there’s the ‘Bold’ button: it can have 40+ different calls and interactions.

Clearly, for a rich text editor to perform, the fundamentals need to be right.

The fundamentals of rich text editing

So, what is a rich text editor? It sounds like a circular definition, but a rich text editor is an interface for editing ‘rich text’. It’s a crucial component within every organization’s tech stack which enables rich text editing capabilities within an application — no matter the type, use case or device.

The term ‘rich text’ refers to the styling applied to the text — over and above plain text — by virtue of it containing formatting information: bold, italic, strikethrough, different fonts and colours, headings, ordered lists, unordered lists, and its ability to add rich media to the content being created.

Most rich text editors are also WYSIWYG editors (What You See Is What You Get), so you see the rich text as you’re writing it in the editor. Historically they’ve borrowed advancements from desktop and browser-based word processing, and improvements in content management systems. In fact, they’ve synthesized so many advancements that WYSIWYG editors have become the primary means of writing blog posts, building and maintaining document management systems, writing academic and learning content, editing CRM text fields and CMS-based websites, as well as composing emails. That’s a lot of hats to wear.

However, when the fundamentals don’t work as they should, your rich text editor causes more pain than pleasure.

Many developers underestimate their complexity, and fail to grasp that just about everything inside them is really hard to achieve. It’s said that they’re like a complicated Swiss watch with many unseen parts working in synchrony, and they’re absolutely never ever, a once-and-done project.

Never a truer word was said.

Who’s who?

1

Developers are intrinsic to a rich text editors’ health

It’s true. Rich text editors (RTEs) are a core component of almost every tech tool we use today. CMS, DMS, LMS, Email, Messaging, Workflow, CRMs and the list goes on… the roles they play reach into every hidden corner of an organization’s tech (and martech) stack and the amount of time developers are spending *with* them, *on* them and *in* them isn’t declining. In fact, it appears to be growing.

Who responded?

Just like last year, primarily it's full stack developers who are working with rich text editors, despite usually being considered a front-end UI component and/or API. This could be because of the general abundance of full-stack developers or perhaps because a rich text editor requires access and the skills to build connections between some back-end aspects of products and apps.

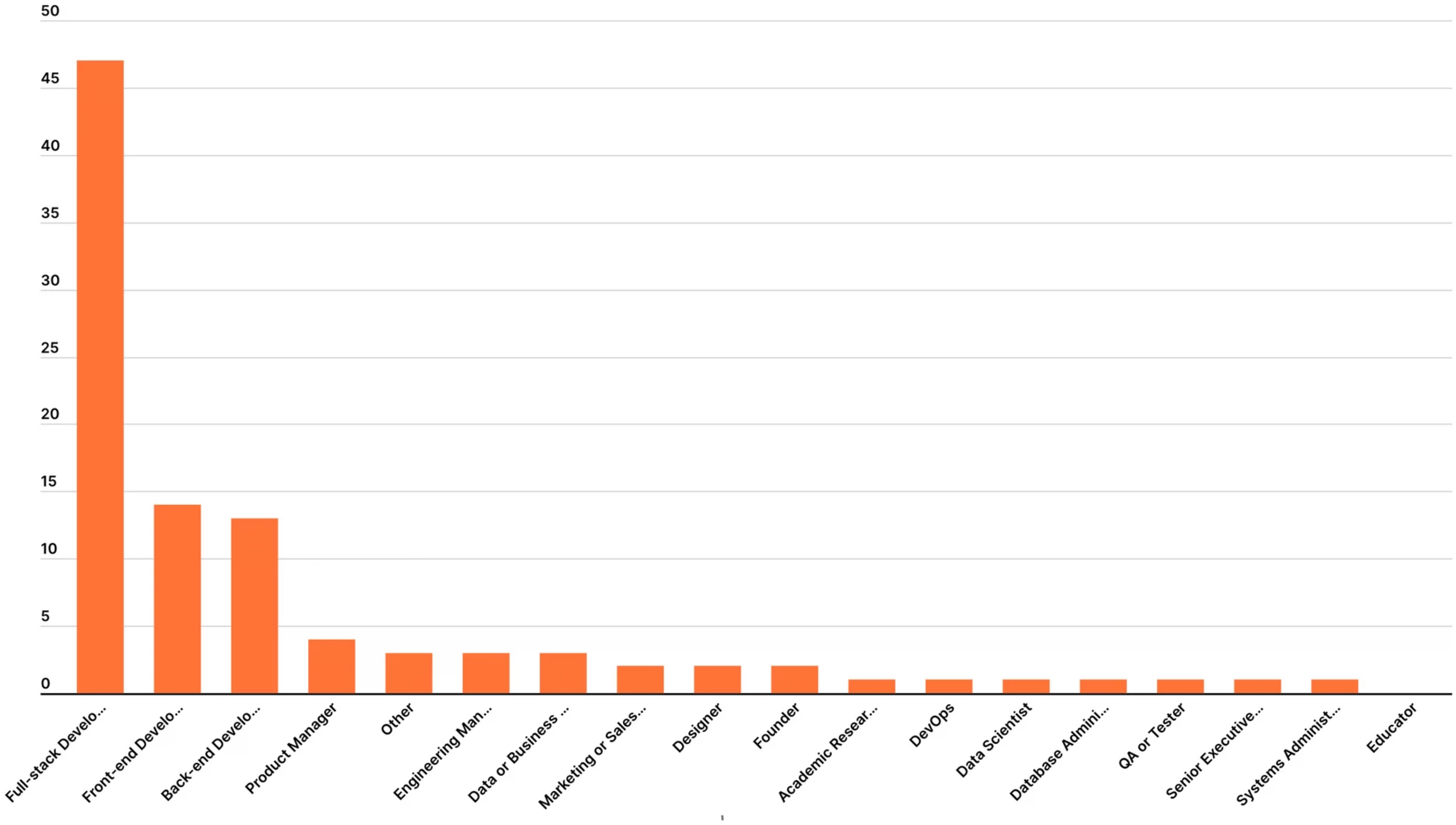
CHANGES FROM 2021

Digging into the details of the raw data:

A SHIFT OCCURRED with more engineers[1] responding — 76.72% in 2022 compared to 63.79% in 202

MAY BE INFLUENCED BY A LARGE DROP in the self-selections of "Other" — down from 39 in 2021 to 9 in 2022

ADDITIONAL JOB TITLES WERE ADDED to the list of title options, which could account for the change



On what basis are you employed?

The majority of respondents working with rich text editors are employed full-time, and in-house. This could indicate there's more in-house importance being placed on editing components as part of the core functionality within the product/applications being developed. And it may also suggest that many companies now see the value in investing in the creation of a great authoring experience, no matter the platform or device.

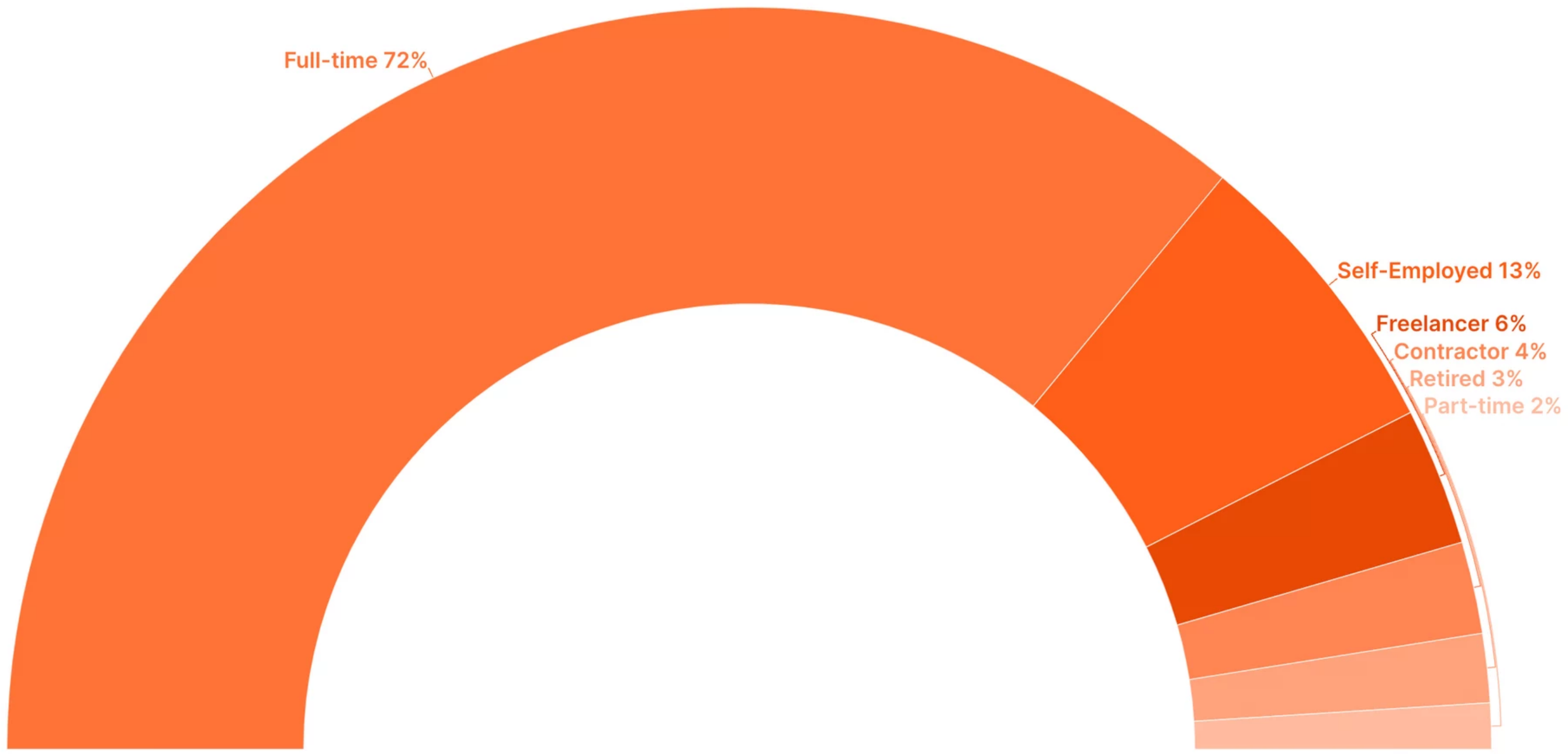
CHANGES FROM 2021

Digging into the details of the raw data:

THERE ARE SIGNIFICANTLY MORE people in full-time employment — 53% in 2021 versus 72% in 2022

SELF-EMPLOYMENT IS STILL THE SECOND most popular employment basis, but declined compared to 2021 — dropping by 4%

THE OPTION FOR "OTHER" WAS REMOVED as a basis for employment, due to the poor data quality received in 2021



How many years experience do you have in developing with RTEs?

A rich text editor may sound boring to some, but it’s the lifeblood of so many organizations around the world. From development to marketing teams, content creators, publishers and legal teams, a rich text editor is intertwined with almost everyone’s workflow. And as far as managing rich text editors, it appears that’s no different — it’s no longer just senior developers using them.

In the last twelve months, the change from RTE work mainly being done by senior developers may have been kicked off by companies embracing digital transformation and needing their senior talent to work on building more flexible tech stacks. Likewise, the 'Great Resignation’ phenomenon may have contributed to the reshuffle, but regardless of the reason, it appears there’s a shift underway towards mid-career developers more often utilizing RTEs, than senior oriented developers.

CHANGES FROM 2021

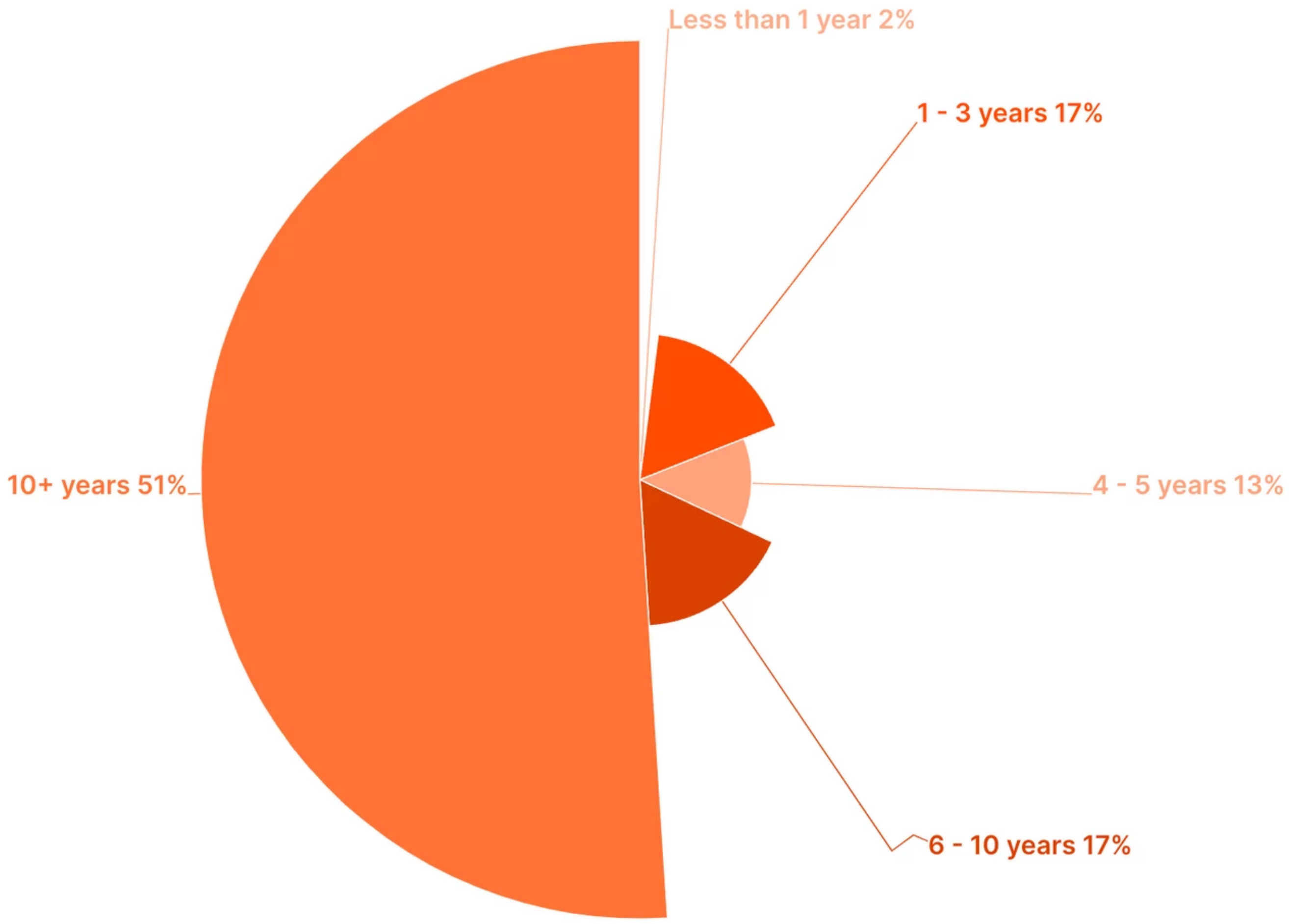
Digging into the details of the raw data:

SHIFT AWAY FROM fresh/new engineers utilizing RTEs

7% DECREASE in highly experienced (10+ years) engineers working with RTEs

MORE MID-LEVEL ENGINEERS working directly with RTEs

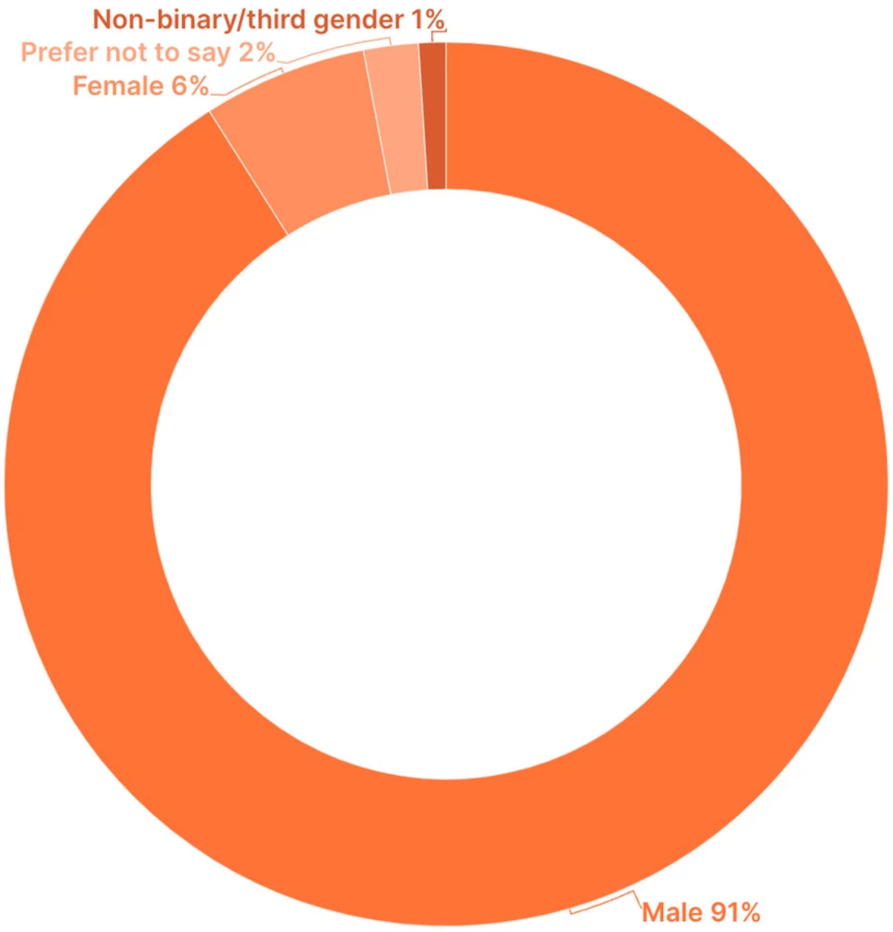
The shift to mid-career experience levels could also indicate the advances made by many RTEs — simplifying integrations, configs and setups — could have allowed senior engineers to step back from the work of integrating the RTE into their product, to utilize their skills on more complex work.



What gender do you identify as?

The responses to this new question reflected the overall industry, and was heavily skewed towards those that identify as male. When the data is broken out and cross referenced to those identifying as working in an engineering[2] roles, there's further consolidation of male responses.

Overall the results are unsurprising, especially given that the long standing industry trend has repeatedly shown that those who identify as male are a much larger cohort than female.



How often do people really develop with their editor?

Many of us who are end users of rich text editors, bounce in and out of them all day long. Whether you're writing content, entering customer data in your CRM or taking notes for a project, there are plenty of great WYSIWYG editors to help make your work easier and more productive.

Historical assumptions have been that developers integrate their rich text editing component and only intermittently interact with it when an upgrade is needed. These results overturn that assumption for the second year running. If they weren't interacting with the editor, we'd expect developers to have chosen 'Rarely' but instead they've selected 'Sometimes', which infers that many developers are touching their editor much more frequently than expected.

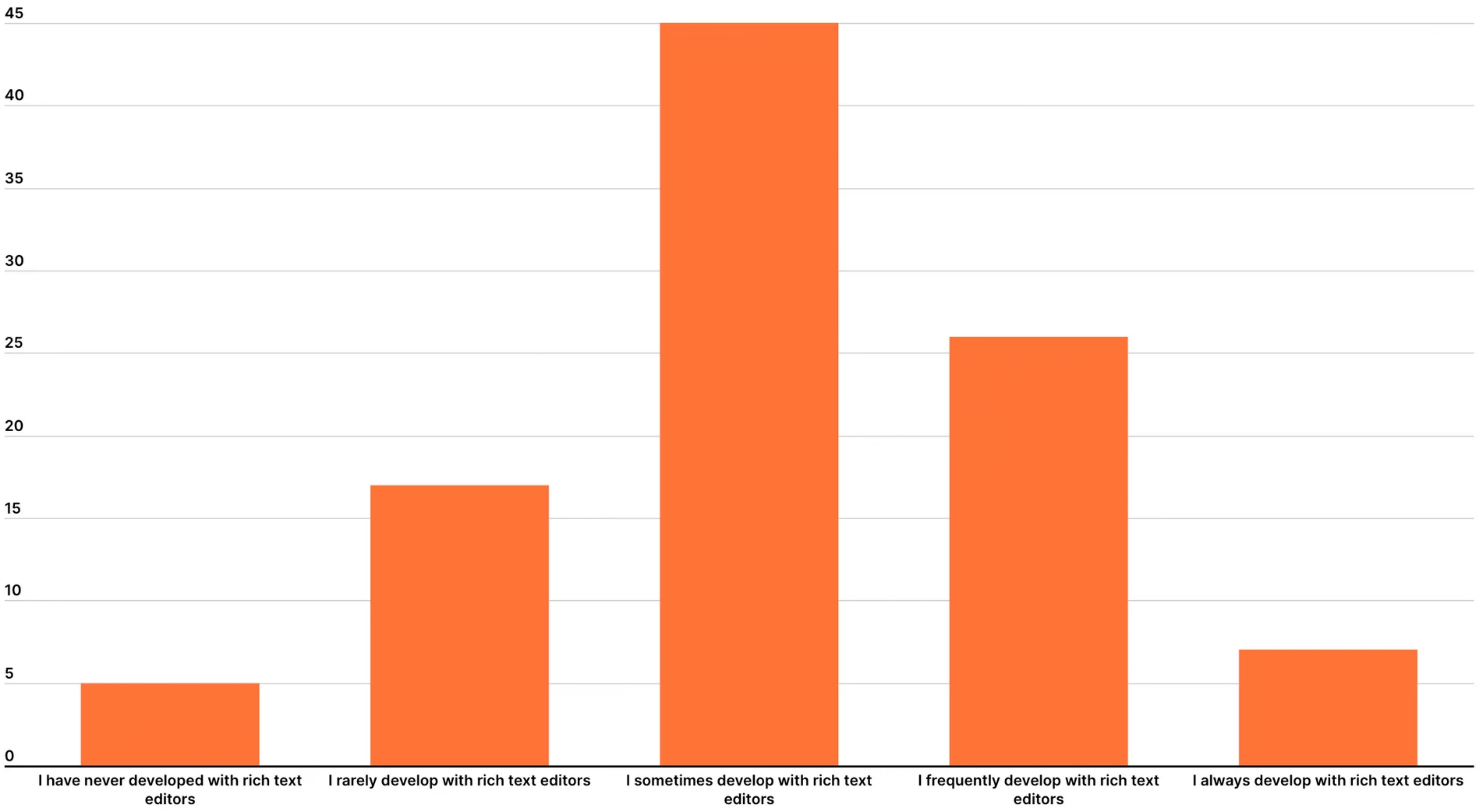
CHANGES FROM 2021

Digging into the details of the raw data:

THERE'S A NOTICEABLE SHIFT in 2022, where more people are 'Sometimes' and 'Frequently' working with RTEs

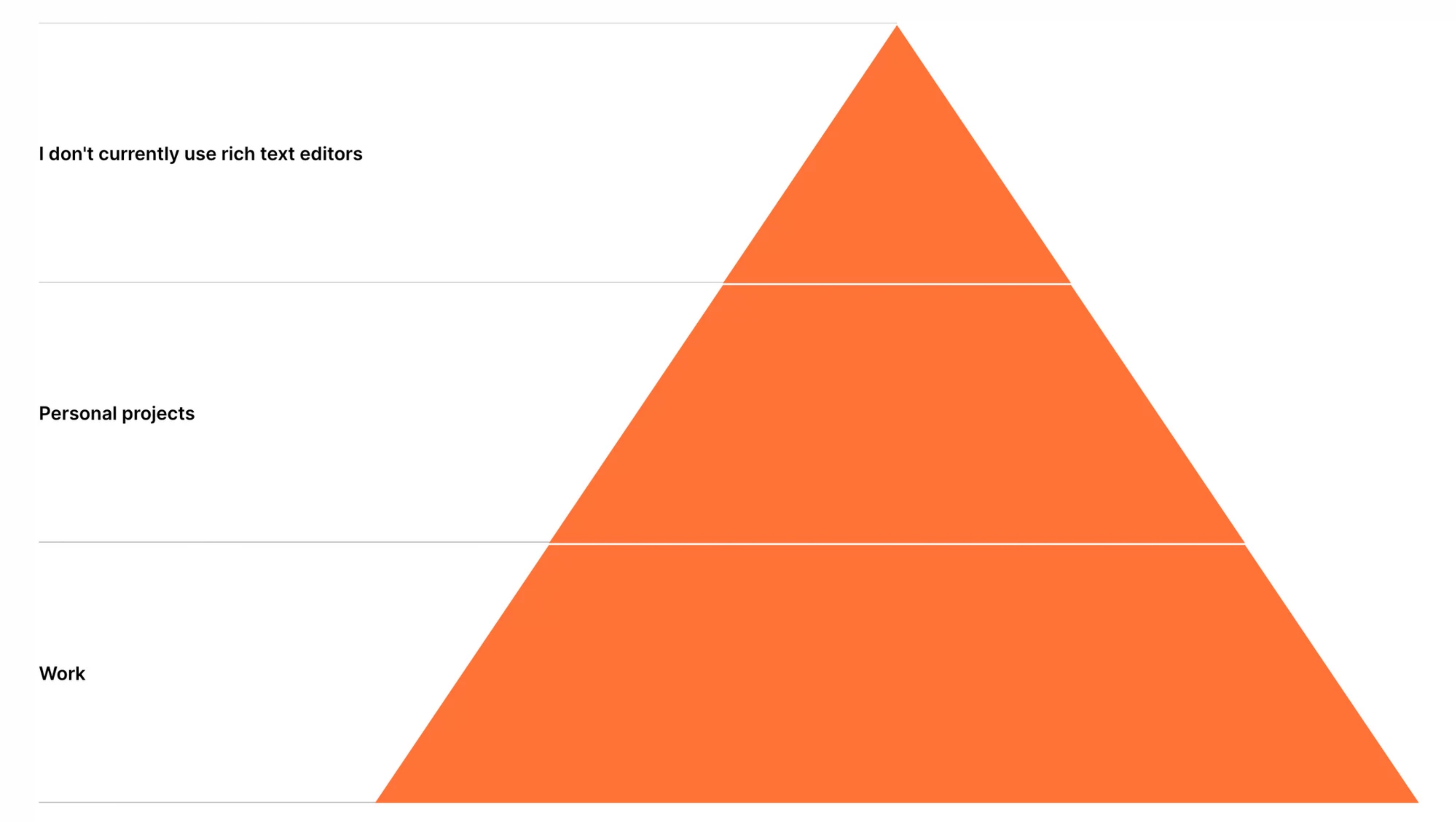
In turn, **THERE'S A 6% DROP** in 'Never' and 6% drop 'Always' developing with RTEs

RESULTS COULD BE SKEWED by a greater number of self-identified 'Engineers' in 2022 (by %), and those roles having a higher likelihood of interacting with RTEs



For what purpose do people use their editor?

It’s clear from this second year repeat result that RTEs are predominantly a workplace tool, with only a small percentage using them for personal projects. The result is also likely reflective of the full-time, in-house skew of the respondents, where you would assume the majority only interact with an RTE at work.



Stacks a-plenty…

2

But, how have the tech stack and text editor usage stats changed?

Transformation and speed-to-market have made the old belief in custom-building all your own technology redundant. Neither budget, nor developer resources, are elastic. So a more agile approach has emerged — that assembles rich text editor components into a flexible, reusable tech stack.

That's allowed developers to integrate rich text editors into their applications in various ways. New opportunities have opened by stitching and re-stitching together APIs, customizing parts in-house and utilizing an open source core editor that your dev team can use as a framework for customization. The possibilities are endless.

What’s your industry?

In 2022 a greater diversity of industries was reflected in the responses. This could suggest that RTEs are becoming more entwined within software of every kind, and are becoming industry agnostic. It could also suggest more industries are realizing they can upgrade the RTE in their tech stack and have therefore broadened their use case application.

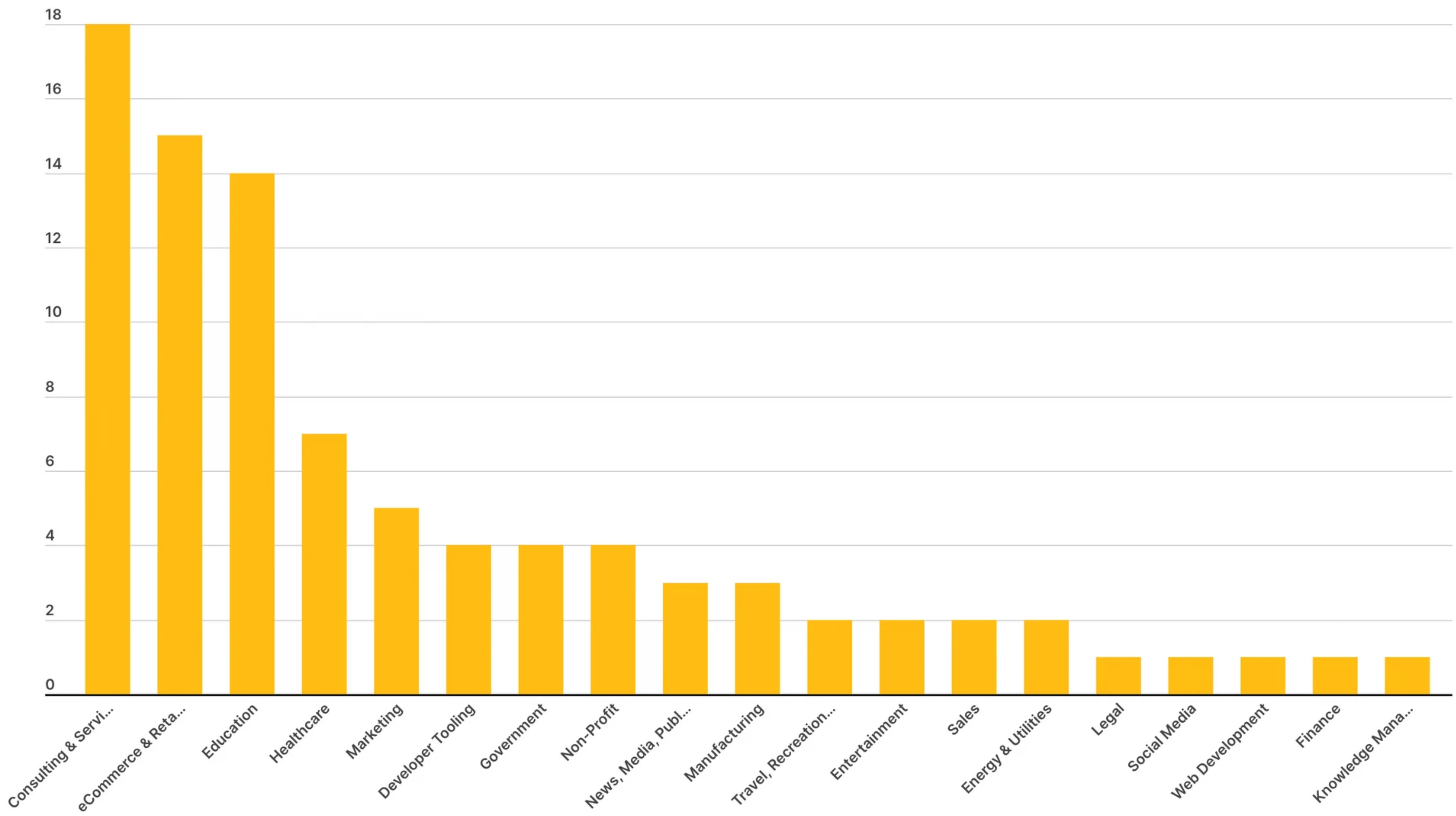
CHANGES FROM 2021

Digging into the details of the raw data:

A SIGNIFICANT CHANGE from 2021 to 2022, after ‘Computer Software’ was removed from the list of options due to poor (industry-specific) data being obtained in 2021

IN 2021 ECOMMERCE & RETAIL WAS MID-RANGE, but in 2022 has risen to second place

HIGHER PERCENTAGE OF ‘CONSULTING & SERVICES’ RESPONSES, however, may be reflective of a pseudo ‘Unsure’ classification



What kind of applications are being built?

Continuing last year's trend and their historically close relationship, CMS was by far the most popular use case for an RTE. Although this is to be expected, given that CMSs and websites are the most popular use cases for rich text editors. That said, there's also deeper usage within CMS aligned areas such as Document Management, CRMs, LMS and Workflow, where their advantages are also being identified and organizations are now pursuing the need for rich text editing components.

CHANGES FROM 2021

Digging into the details of the raw data:

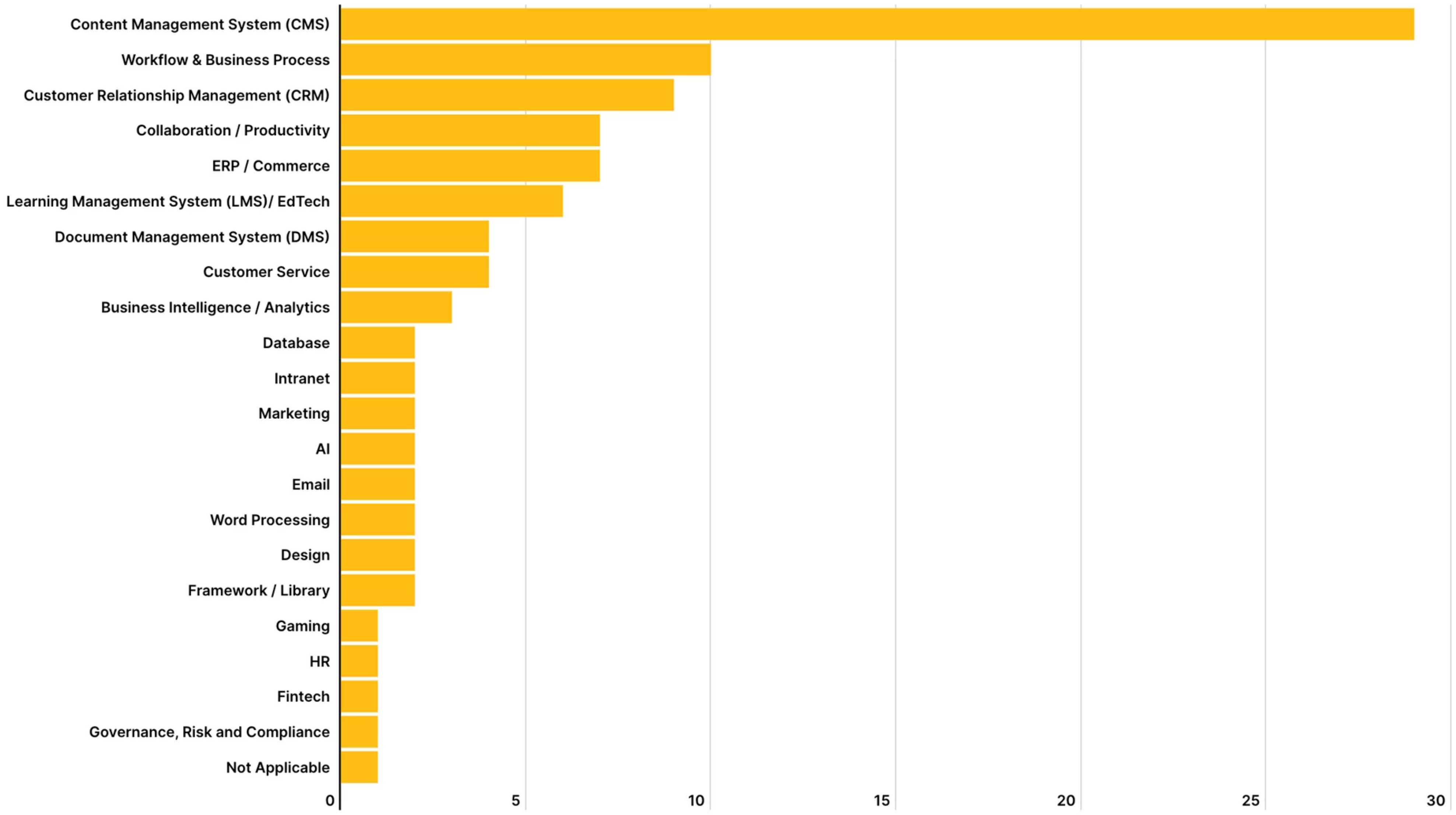
MORE ACCURATE REFLECTION of the market size and potential of various use cases

MARKED INCREASE OF 2% for the Collaboration use case

A 1% INCREASE for the ERP/Commerce use case

THE 'TYPE OF APPLICATION' OPTIONS were changed in 2022 due to poor (use case specific) data being obtained in 2021

THE 'WEBSITE' OPTION WAS ALSO REMOVED to better clarify the CMS use case



What kind of development work does your company do?

The 2022 results highlight that once again the majority of RTE users are primarily either developing SaaS software or websites. This year 'OEM' was added as a separate option, versus commercial software product development, to better gauge if respondents self-identified as makers of components or complete SaaS solutions.

Note: The 2022 survey endeavored to better understand the in-house and custom-tailored applications undertaken, as well as gather more detail on what encompasses 'website' development work. It's likely that future surveys will further refine this question, to more clearly define potential areas of RTE usage.

CHANGES FROM 2021

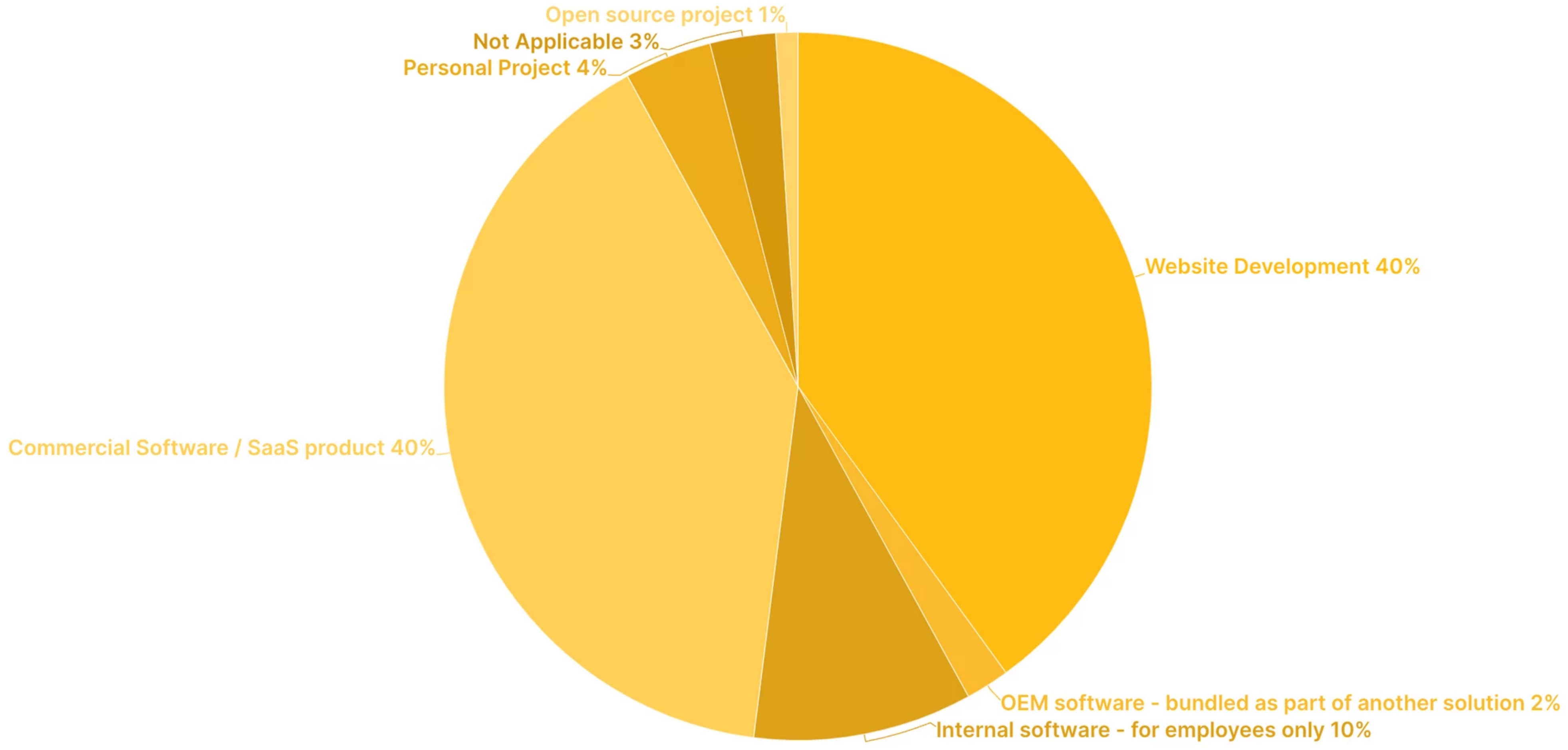
Digging into the details of the raw data:

HIGHER PERCENTAGE IN 2022 focusing on commercial software — going from 26% to 41%

Similarly, **INTERNAL SOFTWARE DECREASED** — from 17% to 10%

WEBSITES INCREASED — from 27% to 31% of respondents

THIS COULD BE A SIGN that more SaaS products are utilizing RTEs



Preferred programming languages

Programming languages are important to understand, and it's vital to understand what language developers favor. This allows RTE makers to better understand the languages they should be compatible with. PHP is still predominant, followed by JavaScript — and while there's a cluster of other languages noted, just as significant is a segment of no-code respondents using rich text editors.

Note: ColdFusion is an Adobe Product that's written in Java.

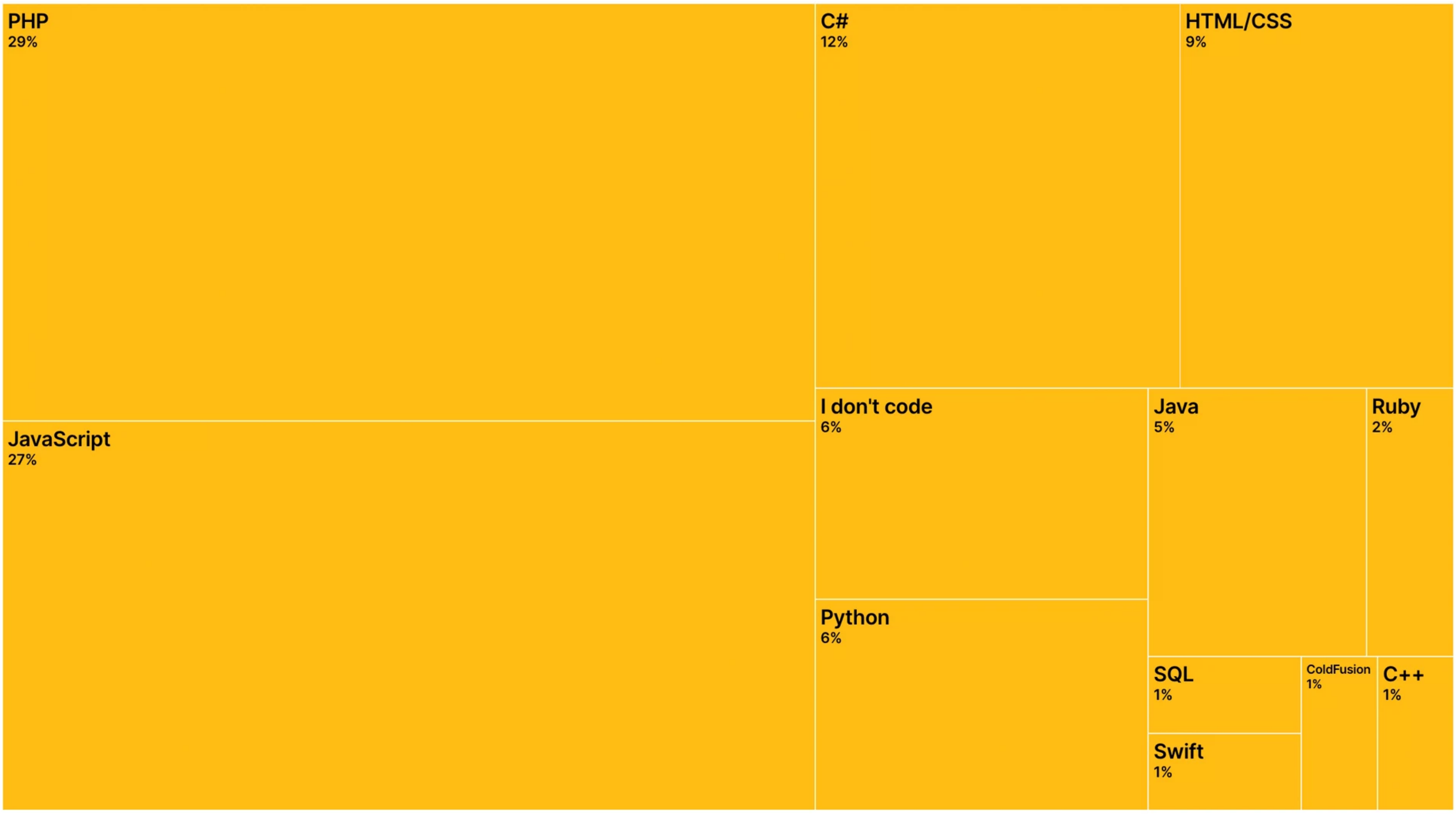
CHANGES FROM 2021

Digging into the details of the raw data:

JAVASCRIPT is becoming more popular

COULD INDICATE that developers are beginning to standardize on more familiar languages

DRAMATIC DECREASE in "niche" languages being selected



What front end framework do you use?

In 2022, 'React' raced ahead of 'Bootstrap' as the preferred framework to use in development work. From the results, it appears there's no variance between the 'Preferred' (see above Q& A) and the 'Used' frameworks. As was the case in 2021, this could signify the importance of APIs for custom work and the need for quality documentation and 24/7 support to answer integration queries from developers.

CHANGES FROM 2021

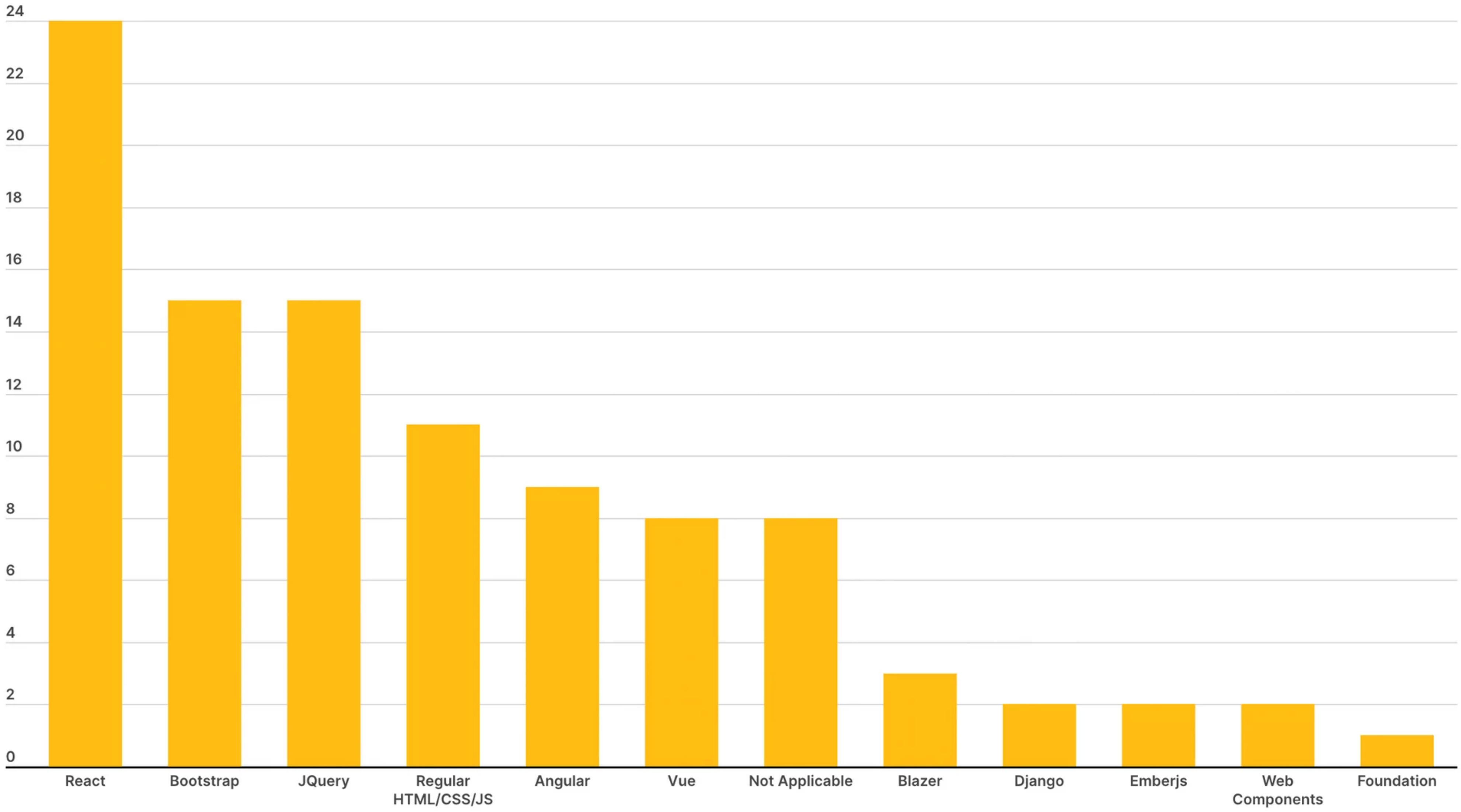
Digging into the details of the raw data:

REACT is a more popular framework choice in 2022 versus 2021

SLIGHT PERCENTAGE INCREASE in JQuery

REGULAR HTML/CSS has slipped in popularity

COULD BE A POTENTIAL STANDARDIZATION of popular frameworks and less use of niche frameworks in products



What’s your weapon of choice?

3

Editor, uses and stuff being made

When the web became a publishing platform, content writers, designers, and other non-programmers needed a tool to create web content without needing to learn HTML, the markup language of the web. As a result, web-based rich text editors have become the weapon of choice (or necessity) for many content creators. Most modern WYSIWYG editors make use of the *contenteditable* html attribute, that when set to true, allows the user to directly edit the content of an HTML element.

But with so many different rich text editor options available for developers to choose from, as well as open source (not to mention building their own) it’s always interesting to see what people are really using and the stuff they’re making.

What rich text editor do you use at the moment?

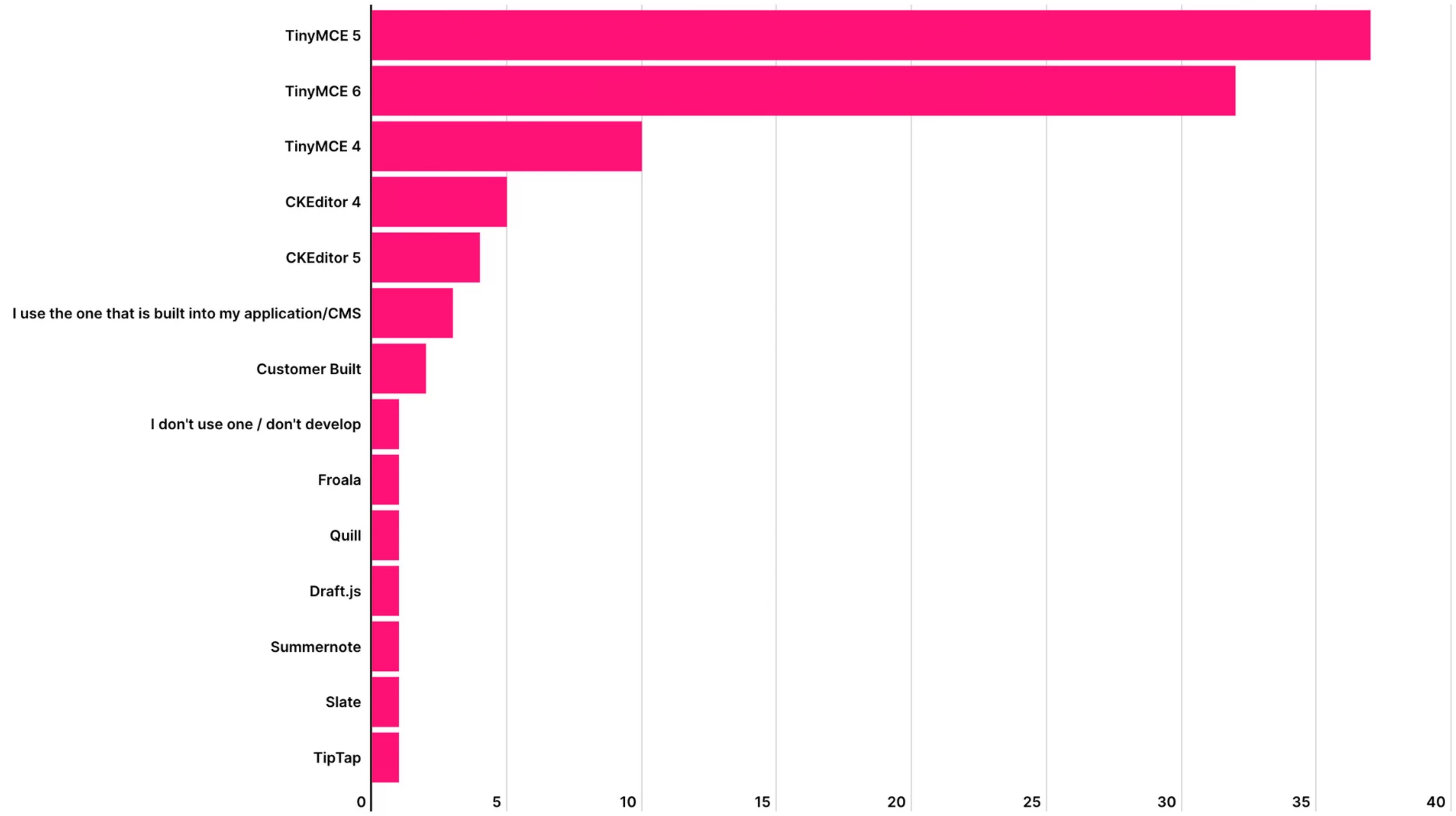
As with last year's results, this question provides a snapshot of the many rich text editors on the market. In 2022 TinyMCE remained the top contender, with CKEditor coming in as the second most popular option. Whilst other minor players are mentioned in the responses, they do not make up sizable numbers.

CHANGES FROM 2021

Digging into the details of the raw data:

CKEDITOR HAS RISEN in terms of popularity, as well as Custom-built

TINYMCE HAS TAKEN AN EXTRA 10% of the market with the introduction of TinyMCE 6



What rich text editors do you use, or have experience in using?

The purpose of this question is to try and understand how many editors people experiment with before selecting a rich text editor to work with.

Similar to last year, it's obvious that both CKEditor and TinyMCE are the quality rich text editor options on the market. During the process people tend to (a) test different versions and editors (b) try the open source editors, before ultimately making a final selection. This could be a sign that people who're in the testing stage, find that the two key editors on the market provide a much better quality experience for both developers and end users.

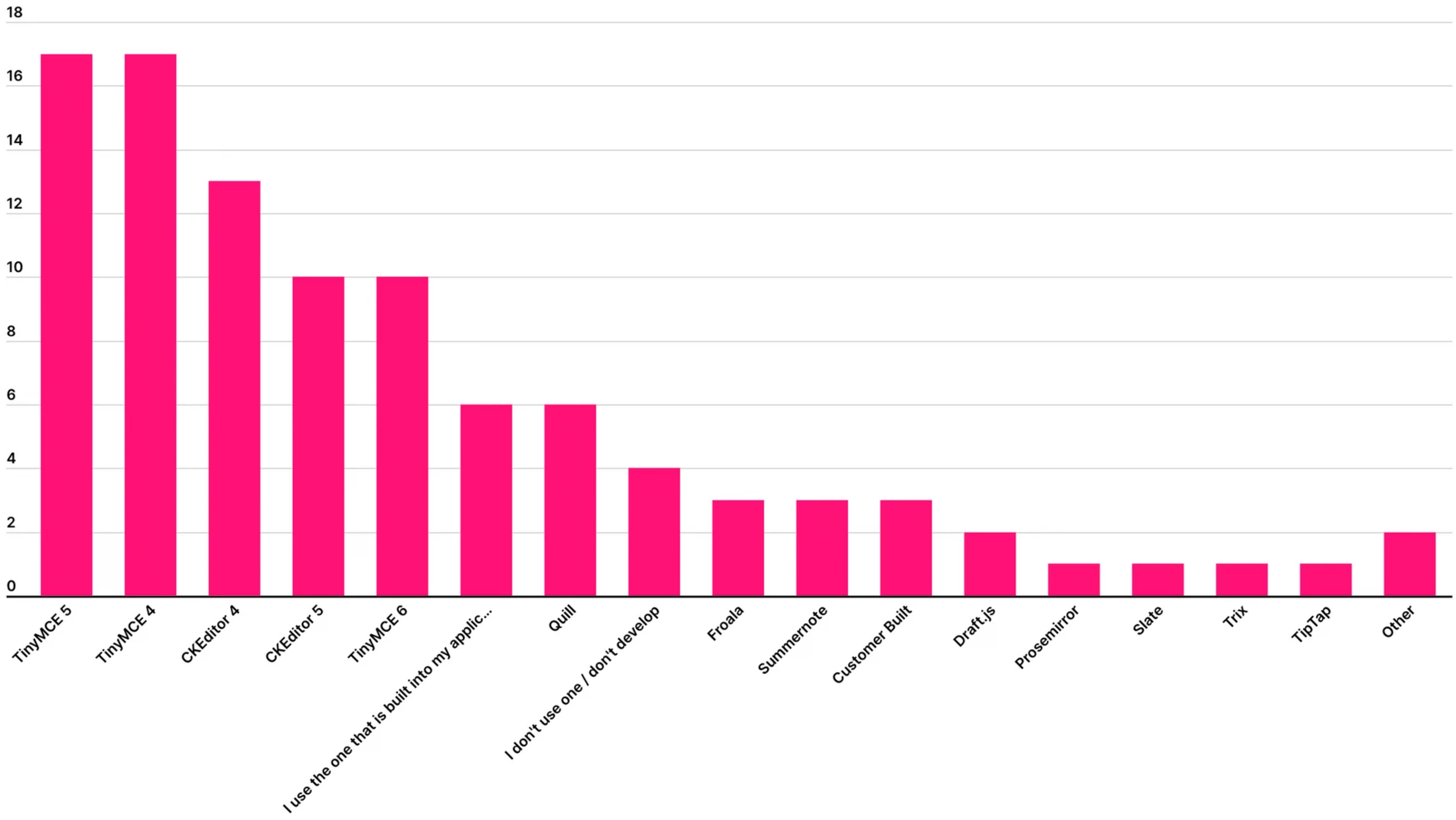
Note: This question allows people to select multiple options.

CHANGES FROM 2021

Digging into the details of the raw data:

THE 2022 RESULTS show a similar trend to 2021

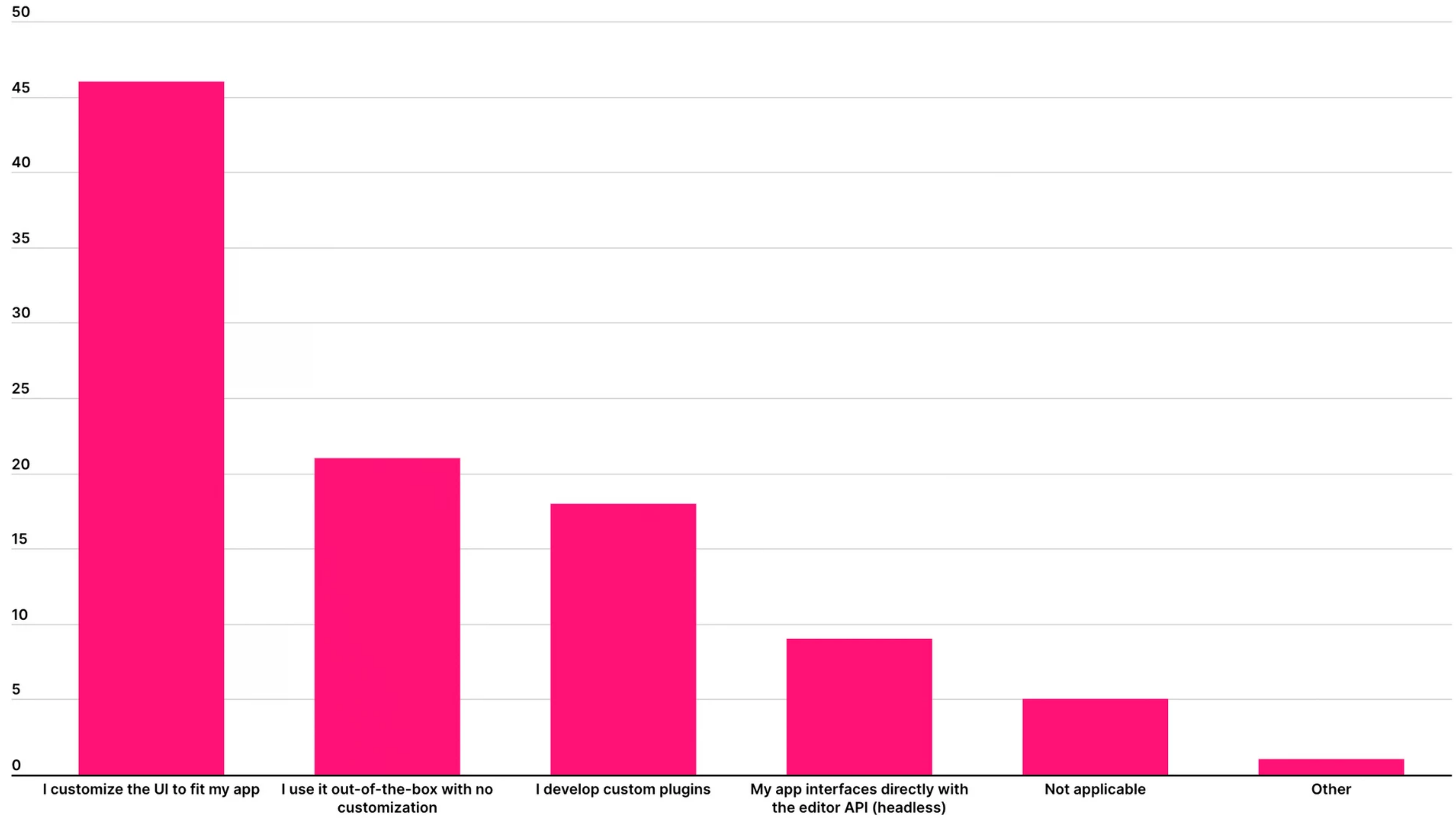
PEOPLE TRY DIFFERENT EDITORS, then decide on either TinyMCE or CKEditor (second choice)



Have you needed to customize your rich text editor?

This new question was added in 2022 to better understand how much customization is undertaken on a rich text editor and how developers interact with various configurations. Most notable in the results is that despite the number of feature-rich editors (and advanced plugins) available, there's a number of developers still working on custom plugins for their RTE instance. In addition to that, almost half look to customize the UI to fit their particular app.

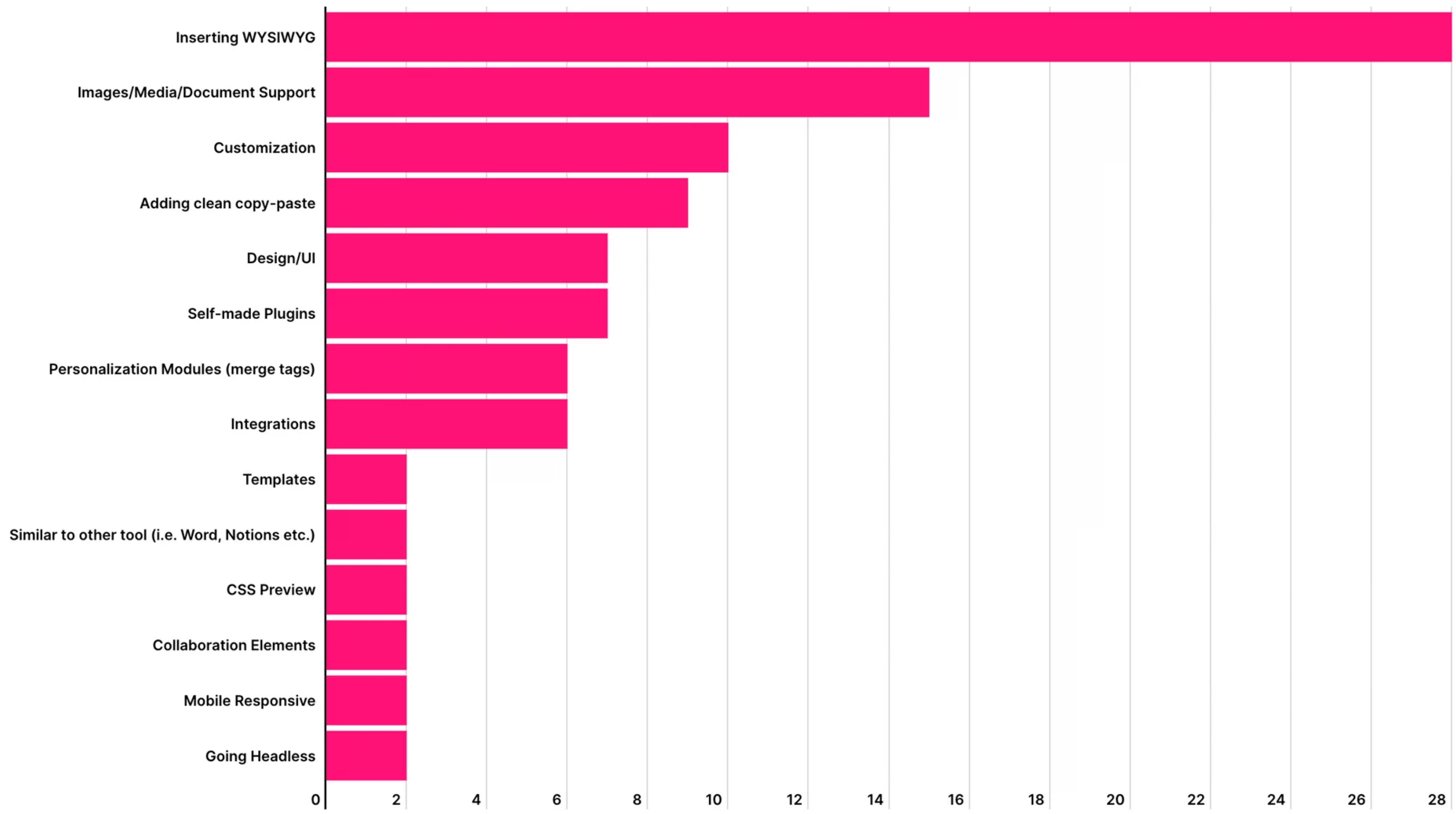
Note: This question allows people to select multiple options.



What big improvements did you recently implement that positively impacted users?

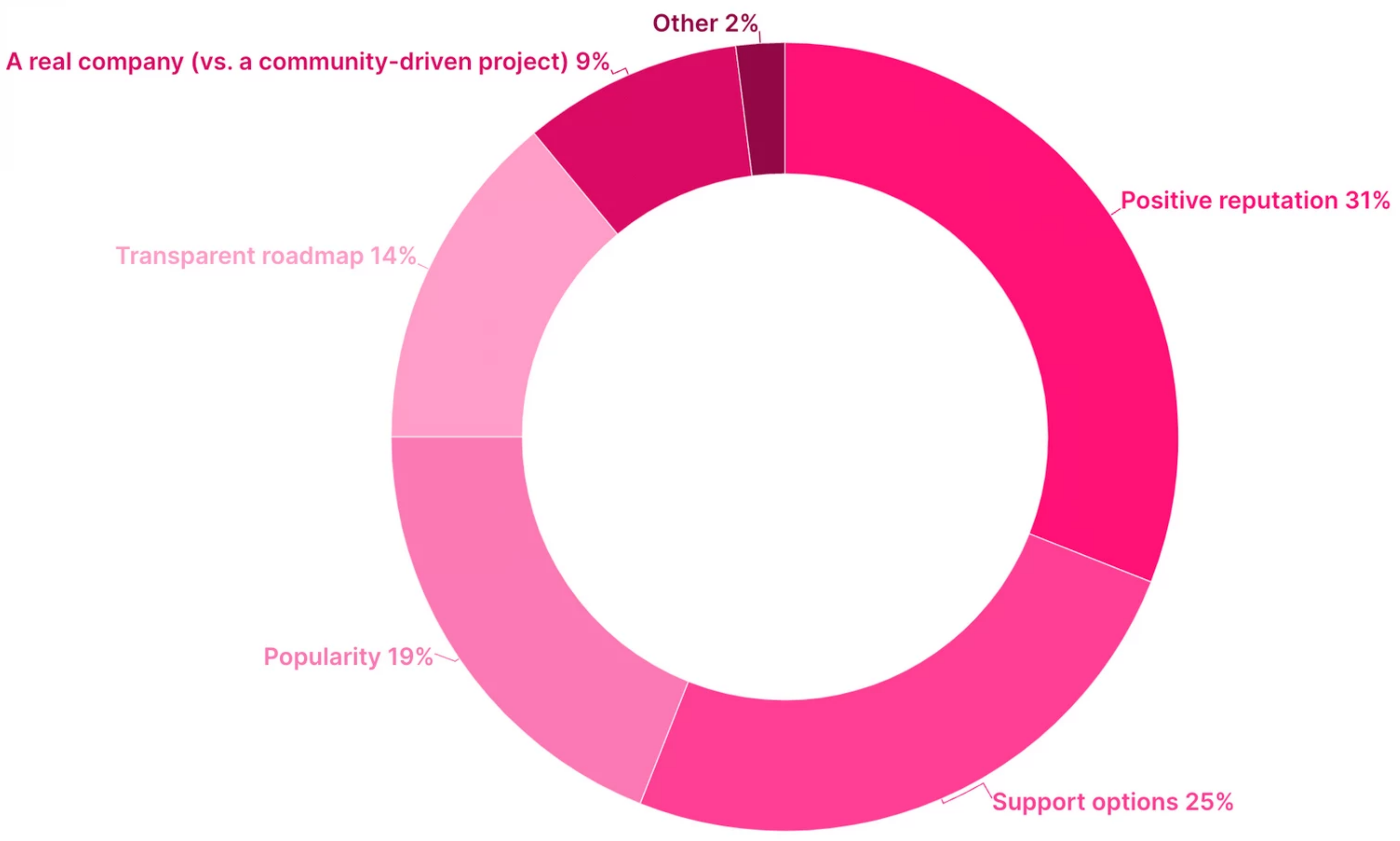
This new open-text question for 2022 was added to encourage more personalized input to the survey results and to understand the biggest things that move the needle for both RTE customers and users. The responses have been summarized and categorized, to help shape the lens of view.

The general implementation of a WYSIWYG editor seems to be the biggest improvement, followed by a focus on upgrading the images/media functionality and document support as ways to improve their users' experience.



What do you value the most while evaluating a third-party software component?

This question was added in 2022 to better understand the driving factors behind the selection of a rich text editor. With ‘Positive reputation’ and ‘Support’ sitting so high in the results, it does indicate that rich text editors must invest in quality documentation, community interactions and professional, easily accessible support to aid developers in their work.



The sticking points...

Choices + challenges — are they causing you more pain than pleasure?

4

Some rich text editors are perfect for experienced developers, while others are more suited to beginners or web content/writers. There's ones that are better for collaboration, copy-paste and document management, while others are scalable and adaptable to suit just about any app — it's obvious that there's clearly no one editor that perfectly solves every situation. The real question is which one suits you?

What's your preferred deployment option for a rich text editor?

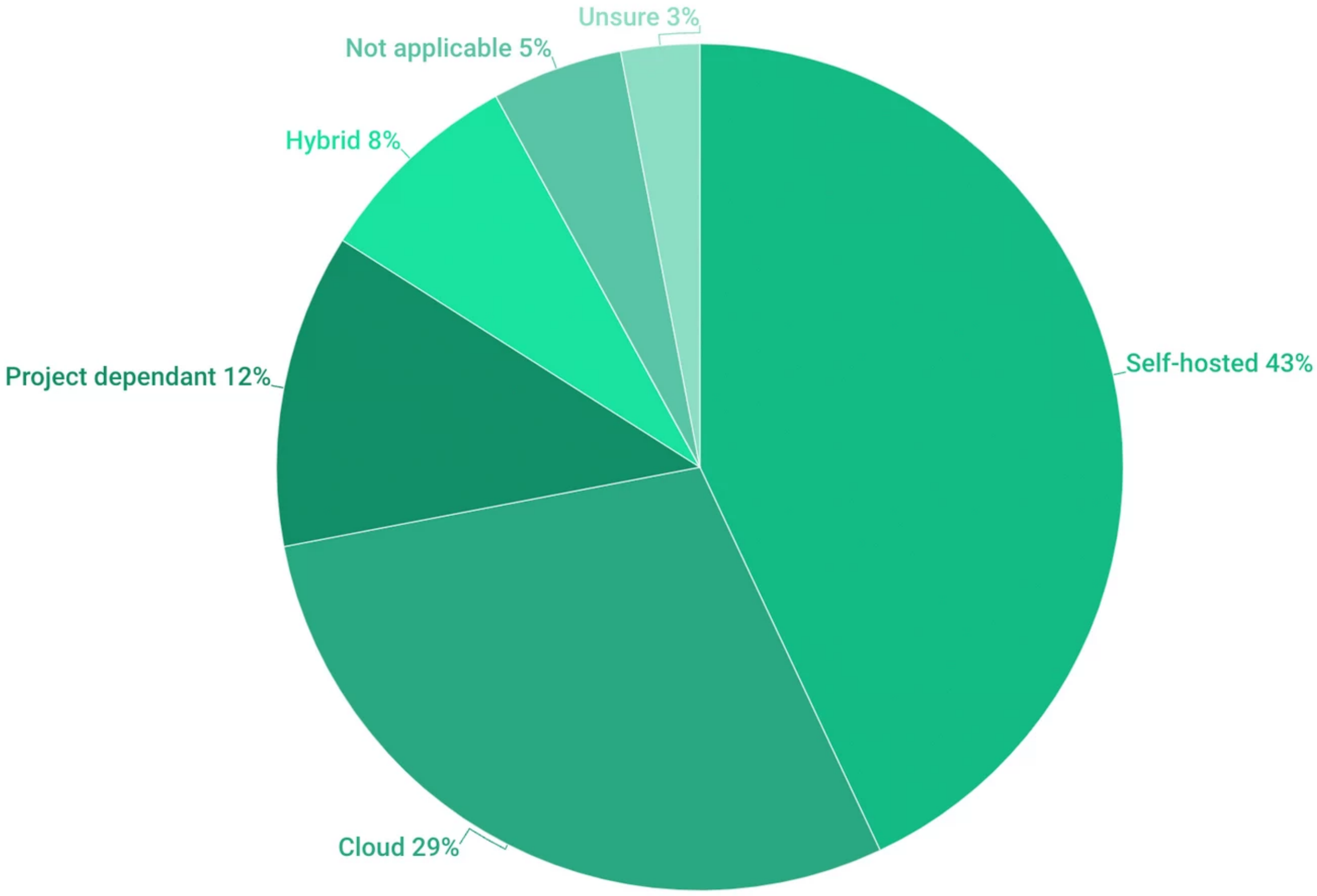
Deployment options are an important gauge of the overall market and speed of transformation being undertaken across the digital landscape. 'Self-Hosted' is still the preferred option, which is curious, given the general race towards Cloud deployment — to meet the speed expectations when undertaking digit transformation.

CHANGES FROM 2021

Digging into the details of the raw data:

SELF-HOSTING has remained the most popular answer across both years

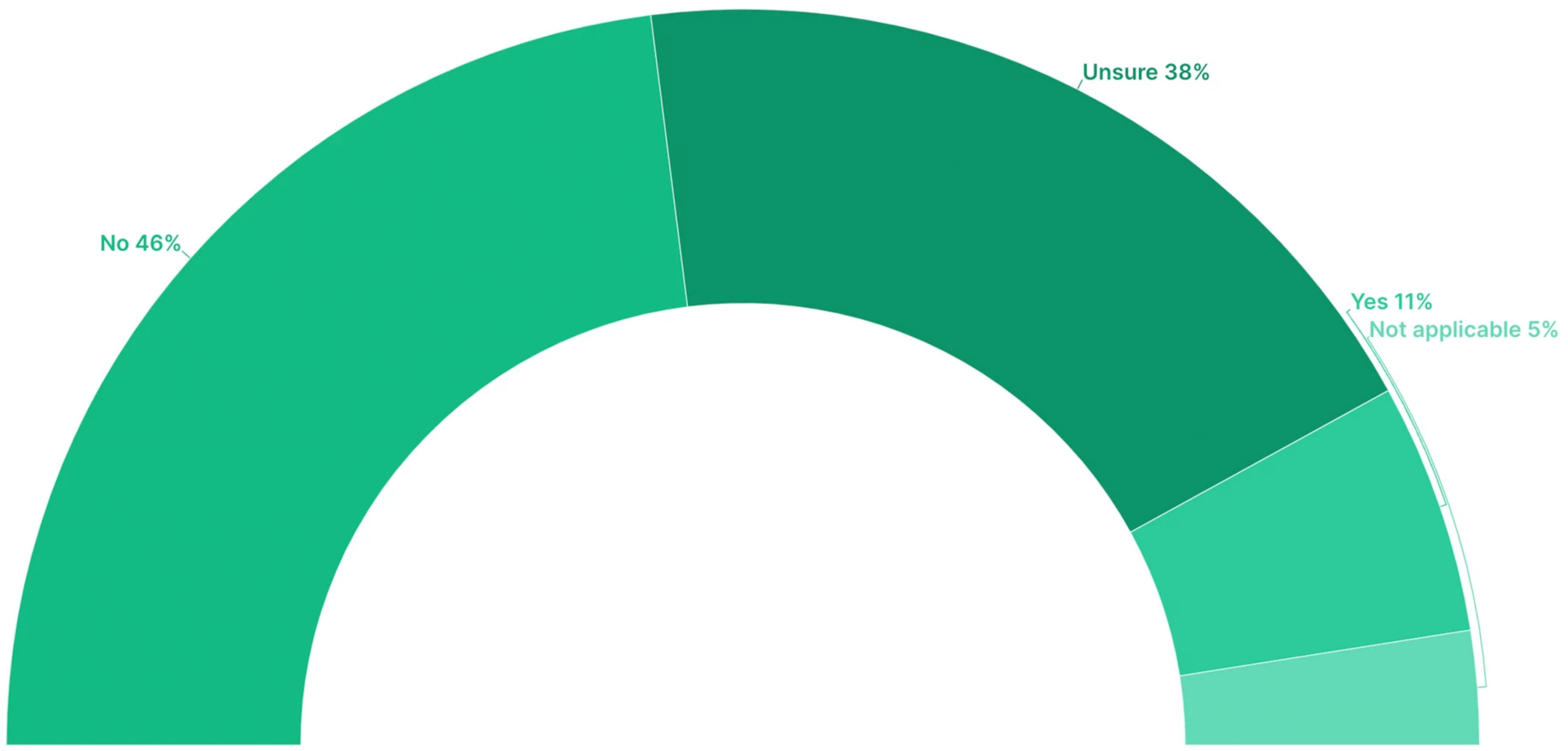
IN 2022 A NEW OPTION WAS ADDED — "Project dependant" — which was the third most popular response



In the next 2 years, are you planning to change the way you deploy your rich text editor?

Closely tied to the context of the preceding question, this response gauged people's readiness for change. Overall, it's clear that the market isn't open to making rapid changes to how they deploy software. However, there's clearly an interest in exploring options — so long as there's no requirement to commit to a specific direction.

That said, with the continued growth in remote working and evolution of Cloud-based rich text editors, there's clearly interest in moving towards a workplace, based in the cloud.



What drives your choice to buy and use a ready-made rich text editor, as opposed to building your own?

It appears the two standout reasons for buying instead of building are ‘Ease of implementation’ as well as ‘Speed of development’, while ’Cost of development’ was a close third response. It’s obvious when time-poor Engineers can rely on a trusted third-party RTE that can be easily implemented and quickly deployed, it makes their job much easier.

The decision to buy is further validated by the knowledge that it takes US$15–28M+ and 115–220 person-years to build the core editor of TinyMCE or CKEditor (excl. Maintenance and extensibility work).

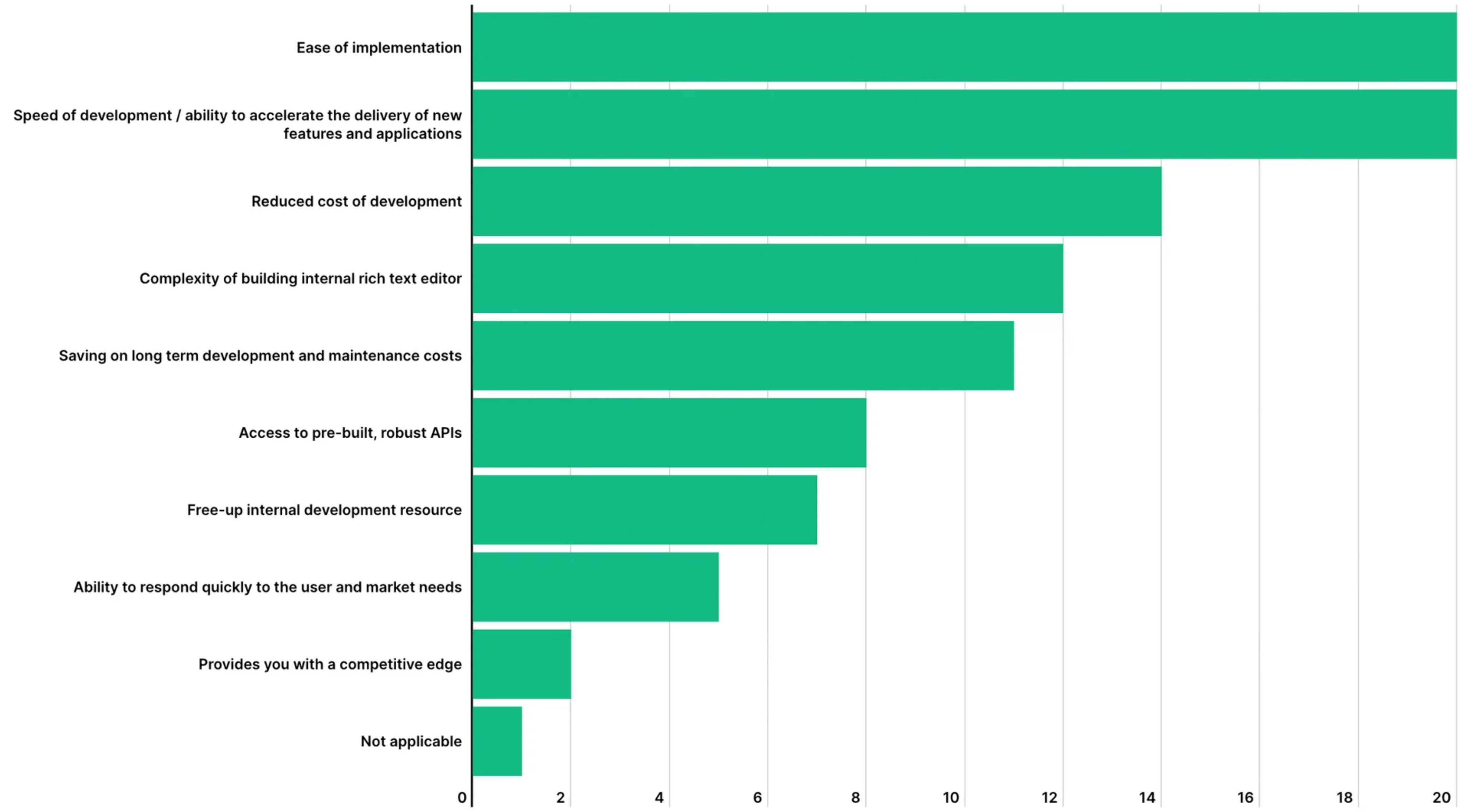
Note: This question allows people to select multiple options.

CHANGES FROM 2021

Digging into the details of the raw data:

THE 2022 AND 2021 RESULTS ONLY DIFFER by a few percentage points

IN 2022 AN OPTION FOR ‘APIS’ WAS ADDED, but does not seem to rate highly in the decision



What drives your choice to build your own rich text editor (RTE) as opposed to using a ready-made editor?

As was the case in 2021, it appears the majority of respondents wouldn't even consider building their own rich text editor from scratch, or using a framework-like solution. This supports the trend that people are after something simple to integrate and customize, rather than having to do the heavy lifting themselves.

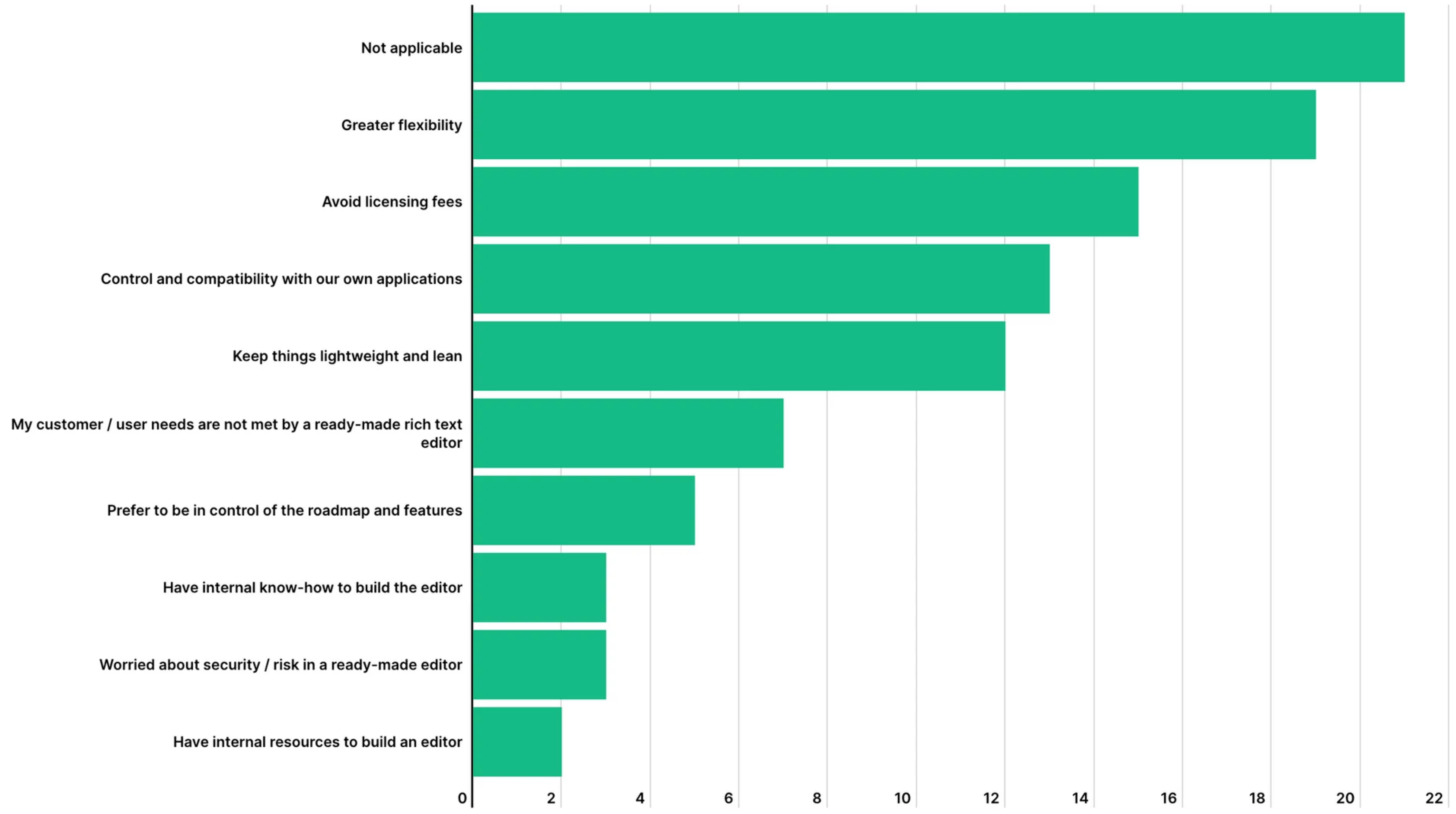
Note: This question allows people to select multiple options.

CHANGES FROM 2021

Digging into the details of the raw data:

COMPARED TO 2021, flexibility and control ranked more highly in 2022

COST MOVED DOWN the list of importance, in 2022



What challenges are you currently experiencing with your rich text editing solution?

When looking at the 2022 results, it seems that developers want a RTE that focuses on a great ‘User experience’ that they know will work when they need it to work.

Note: The below table is done in priority order of major challenges.

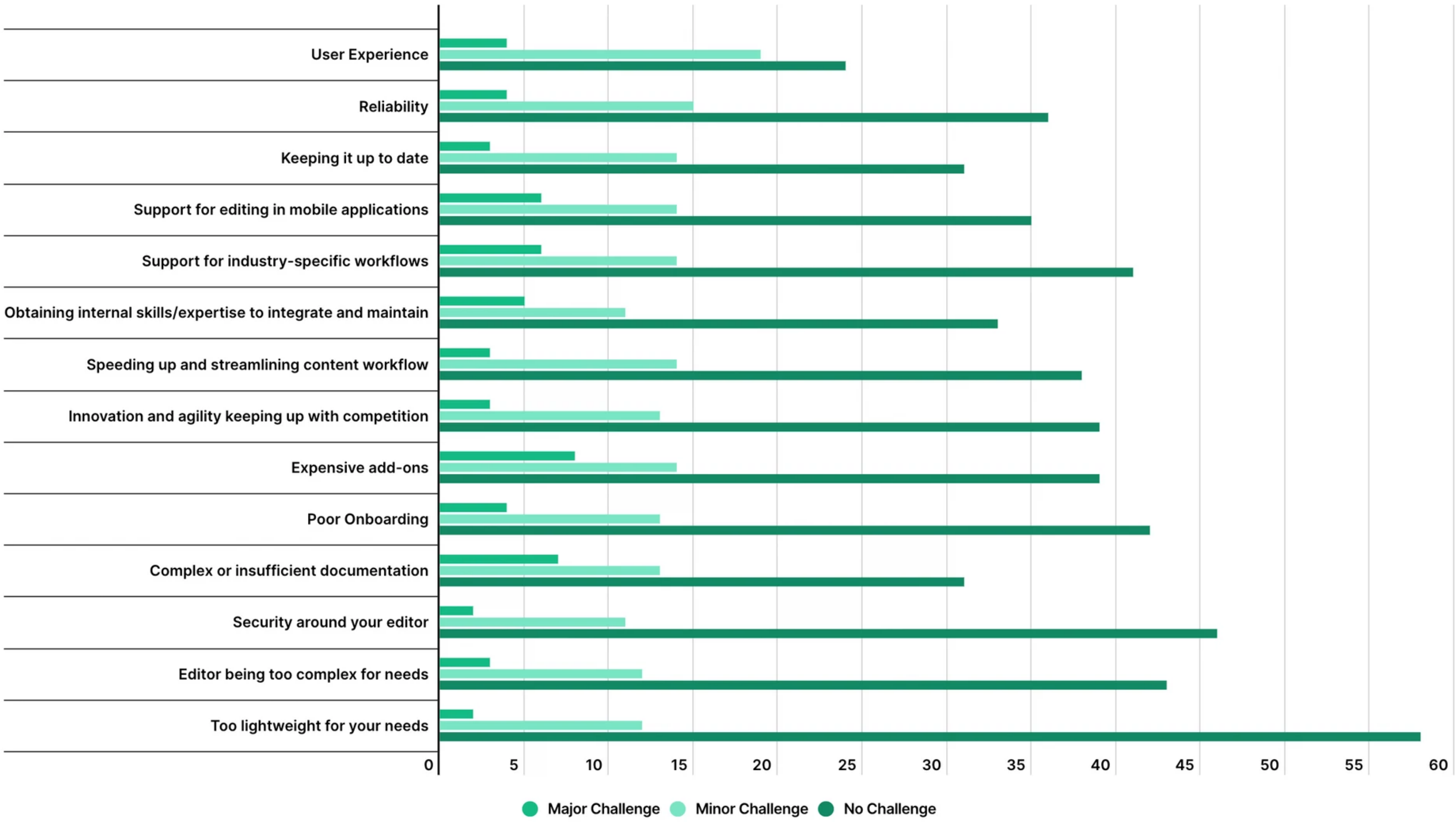
CHANGES FROM 2021

Digging into the details of the raw data:

A SIMILAR TREND to the 2021 results

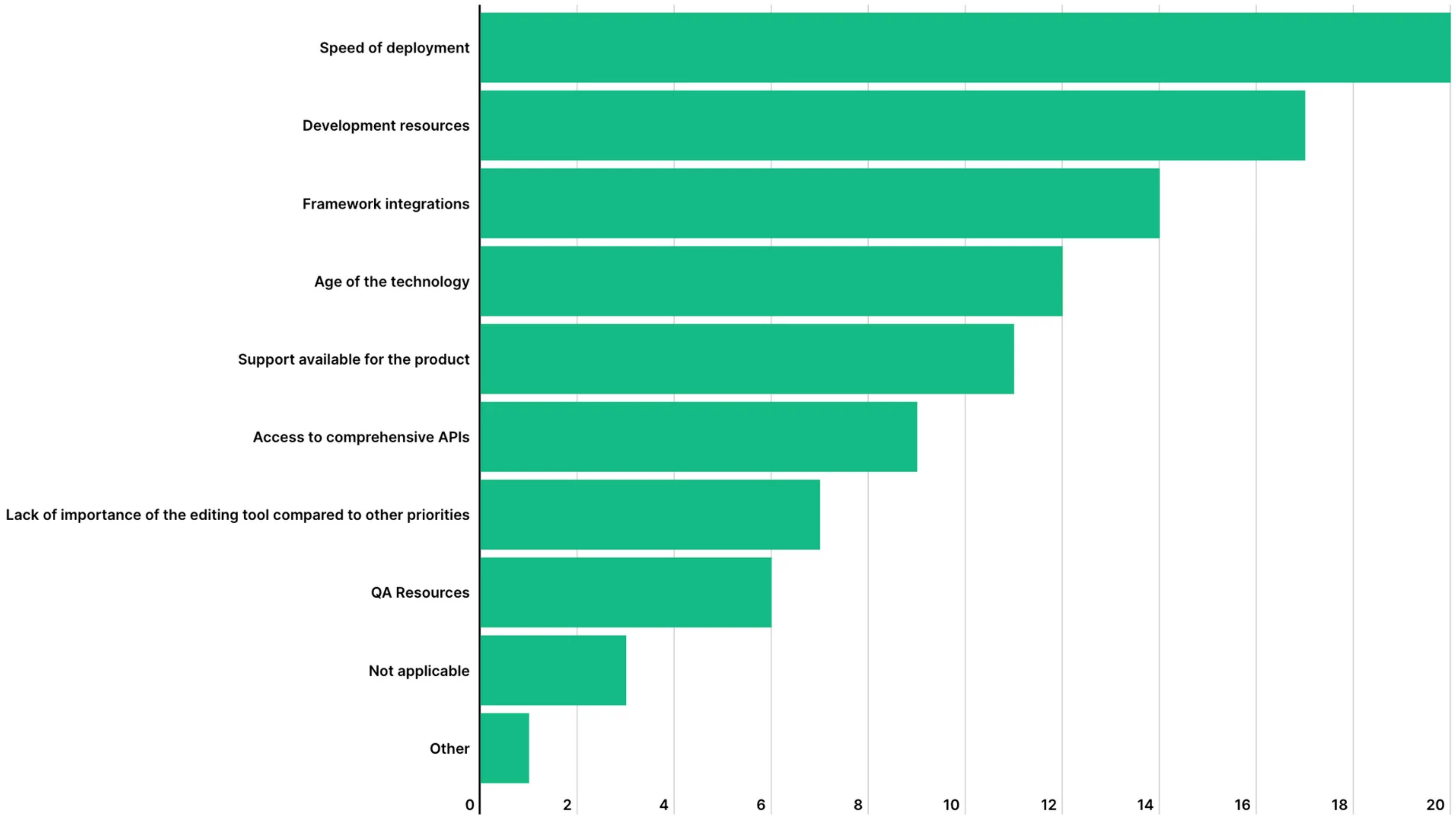
SECURITY REMAINS TO BE CHIEF concern for developers, with their RTE

RESULTS SUGGEST that the fundamentals of a rich text editor are of most importance to developers



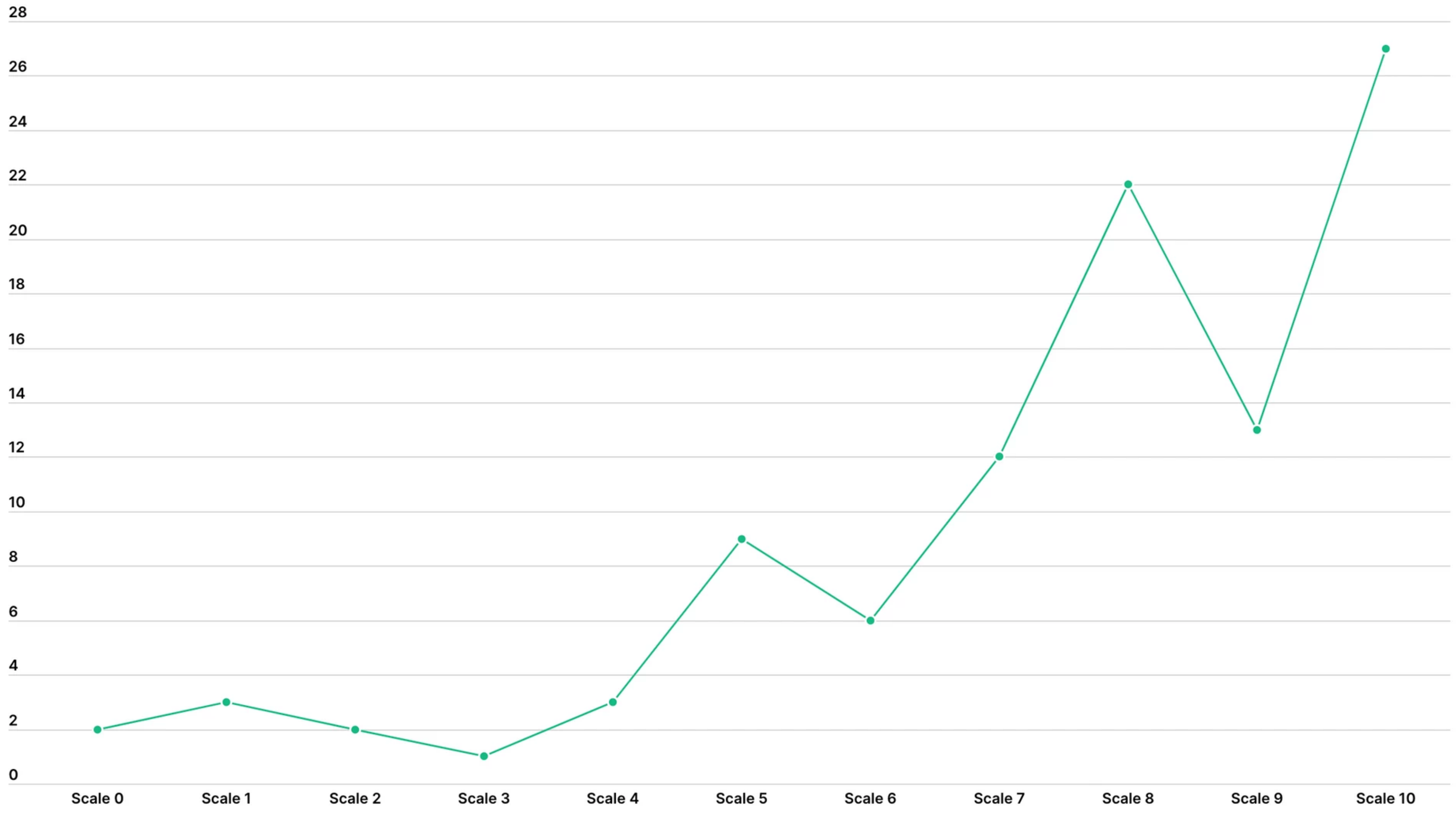
What are the main factors when deciding to upgrade your editor?

This is a new question in 2022, to help the overall RTE market better understand what factors influence the decision to upgrade or change an editor. It appears that ‘Speed of deployment’ and ‘Development resources’ available are key influencers, with ‘Framework integrations', ‘Age of tech' and ‘Support’ being important secondary factors in the decision making.



On a scale of 1–10 how important is it that you ship bug fixes for your rich text editor, in a timely manner to your users?

This new question in 2022, helps the overall market understand how important bug fixing is to the RTE community. Based on the results, it seems that addressing bugs within the RTE platform are amongst the most important things for RTE creators to promptly take action in their software. This indicates builders need to carefully weigh new features/enhancements against the cost of not fixing crucial bugs within an editing system.



Future trends

5

It’s still a ‘keep it simple’ credos

In a world where ‘shiny new objects’ attract our attention, the winning rich text editor formula is surprisingly clear. No innovations, no fancy features, just keep it simple and deliver a great user experience.

What are the most important features needed in a rich text editor?

The 2022 results show that developers rank ‘Security’, ‘Responsivity’ and ‘Customizability’ as ‘Extremely important’ in their rich text editor. The second most notable result was that new, cool, and fancy features still aren’t a top priority.

Developer friendly features remain important — getting started quicker, being able to edit the HTML code and customizing the editor.

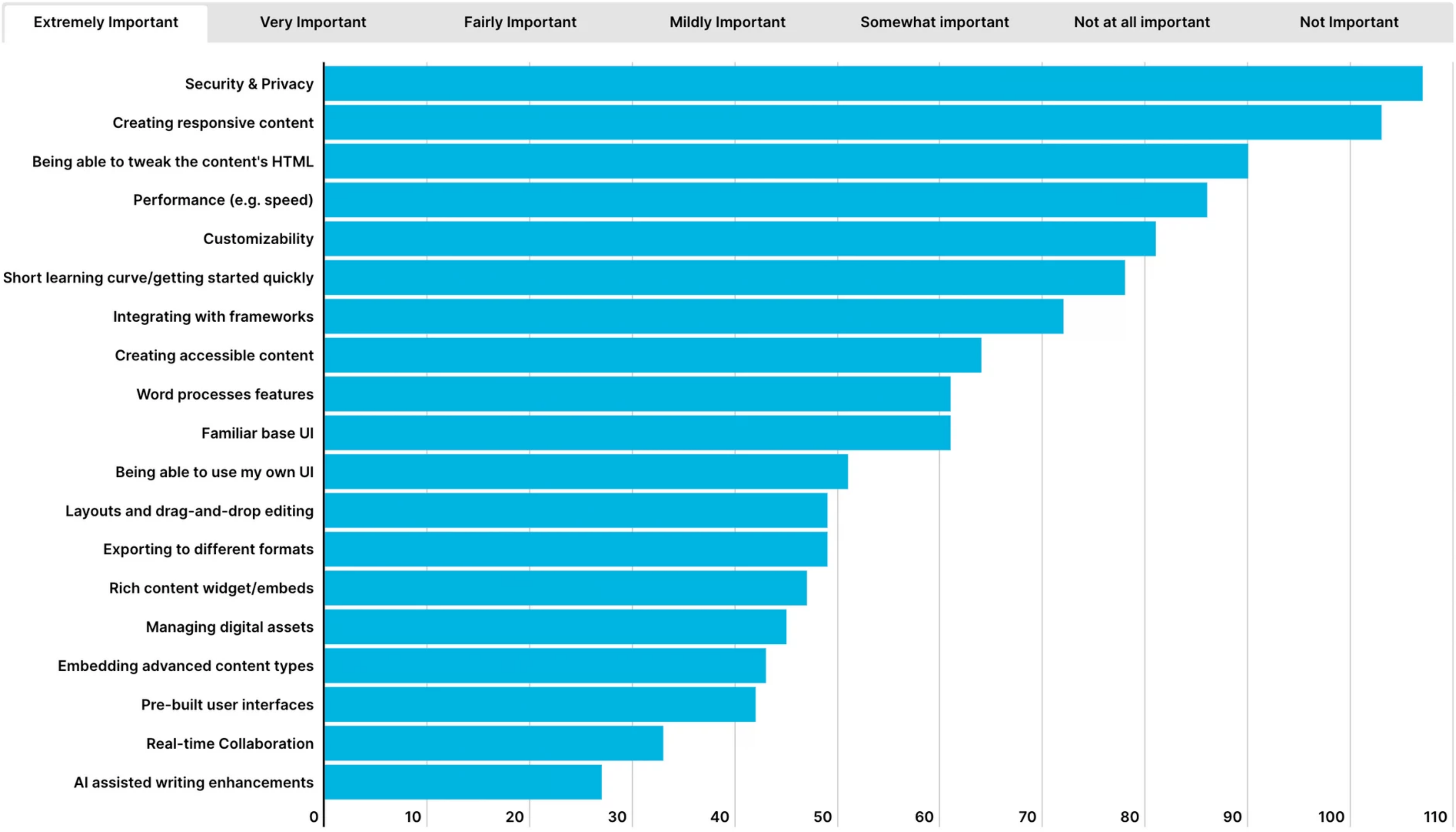
CHANGES FROM 2021

Digging into the details of the raw data:

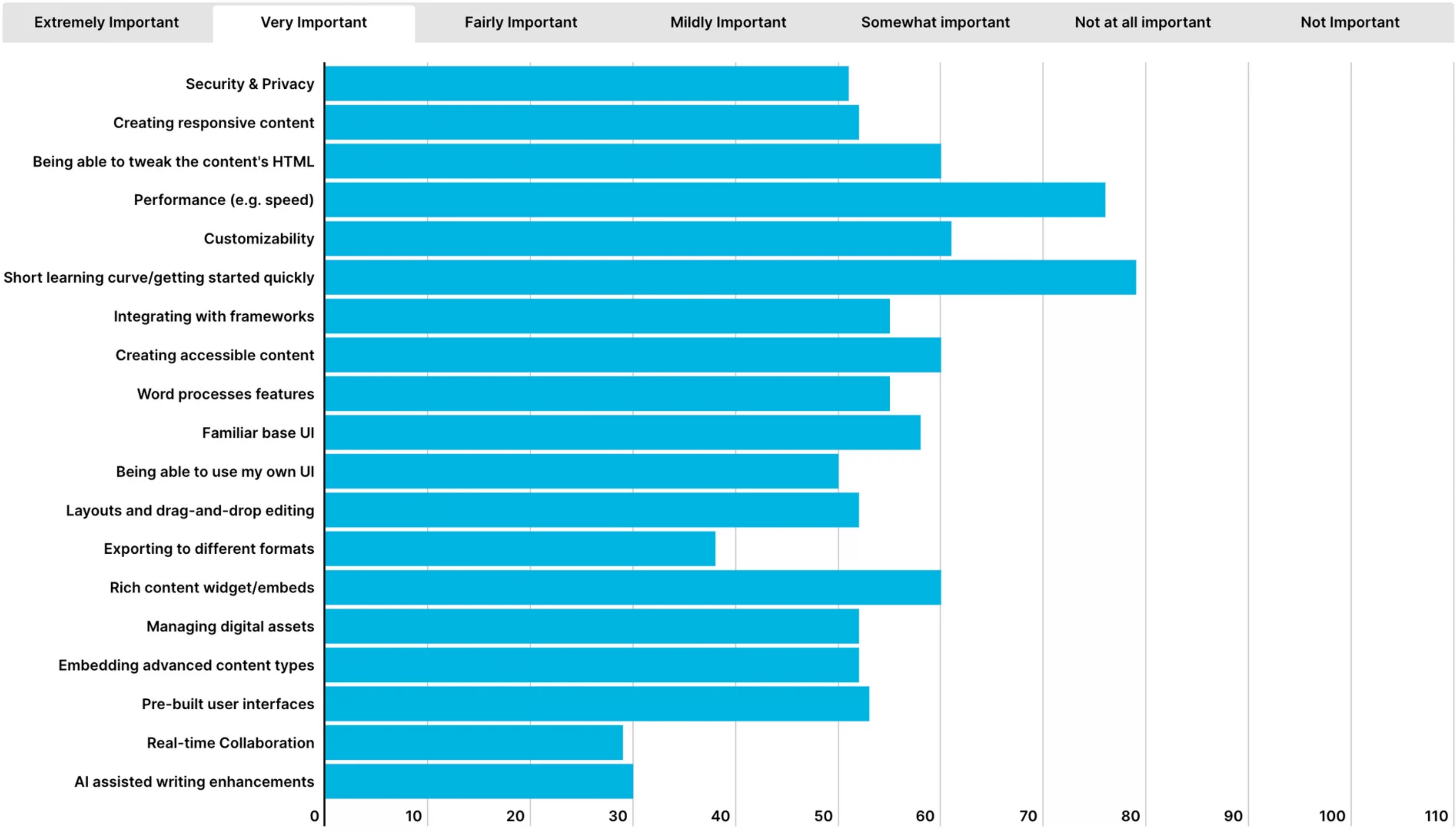
FEW CHANGES FROM THE 2021 RESULTS, where Security ranked the highest, followed by a lack of interest in new features

SEVERAL NEW OPTIONS WERE ADDED IN 2022 — around areas such as UI — but only ranked as ‘Mildly important’ for respondents

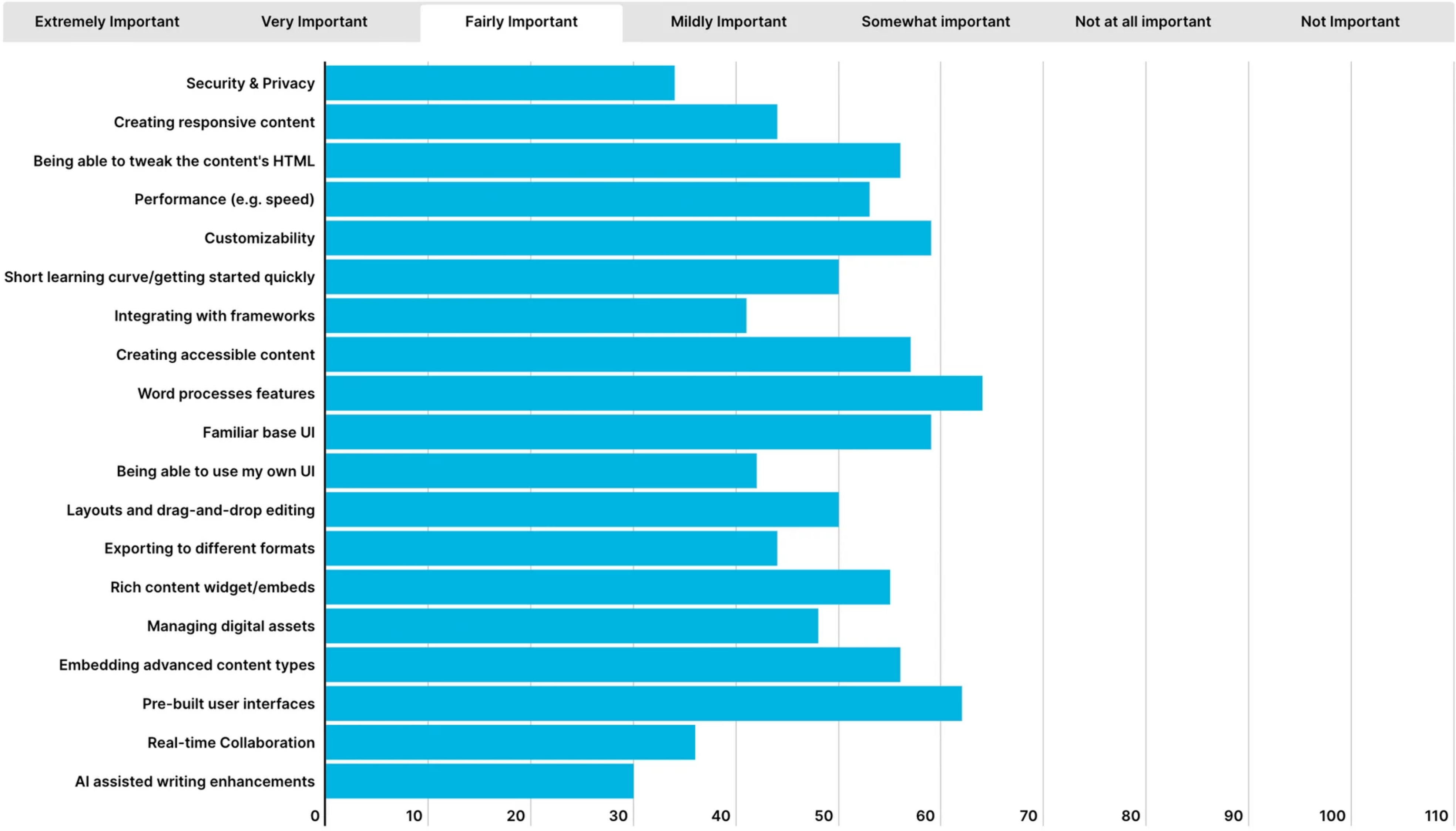
CONFIRMS THE 2021 DATA that focusing on the core fundamentals of the RTE system seems to be of key importance to the RTE market



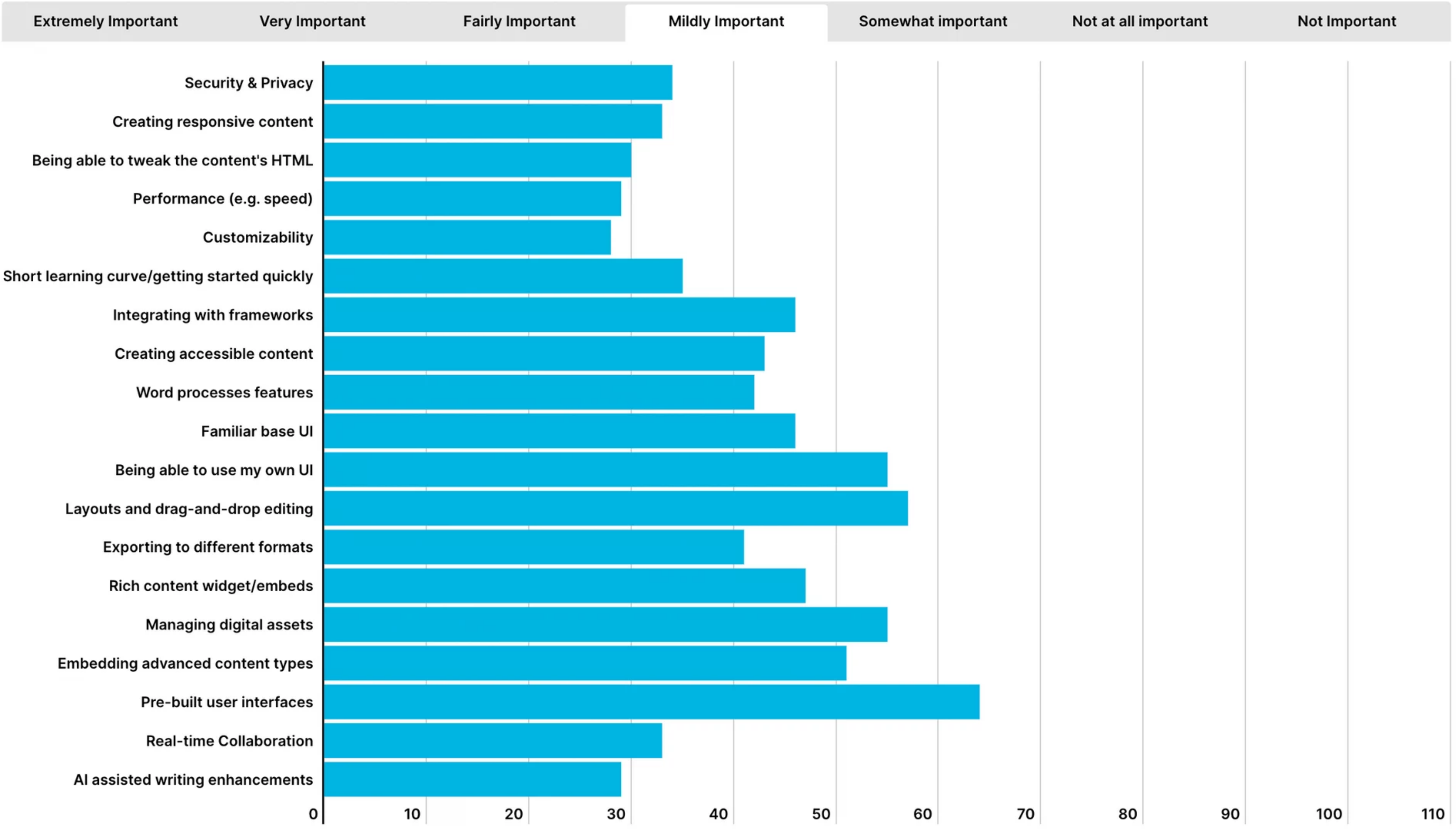
Number of responses. Click tabs to change results view



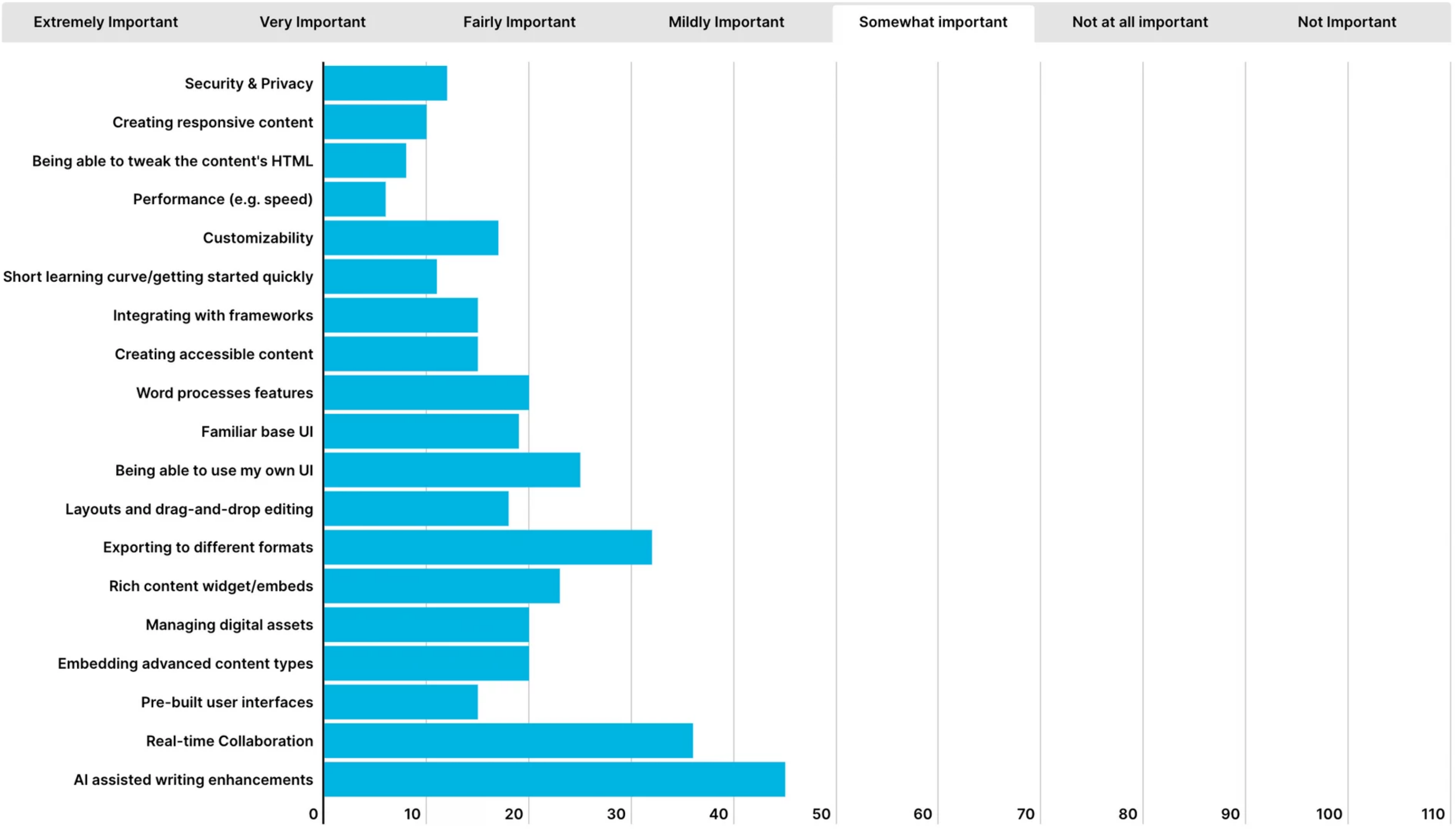
Number of responses. Click tabs to change results view



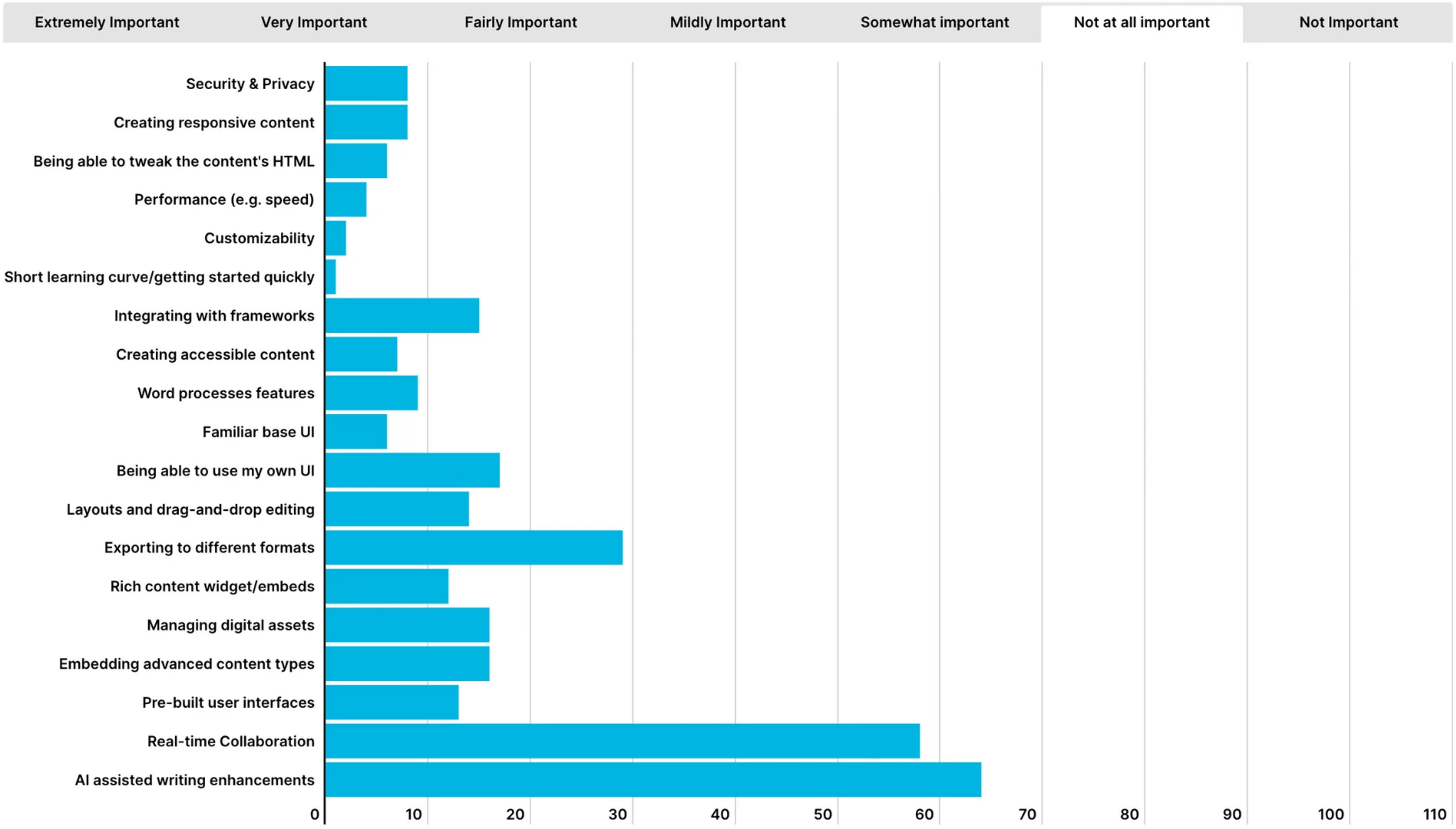
Number of responses. Click tabs to change results view



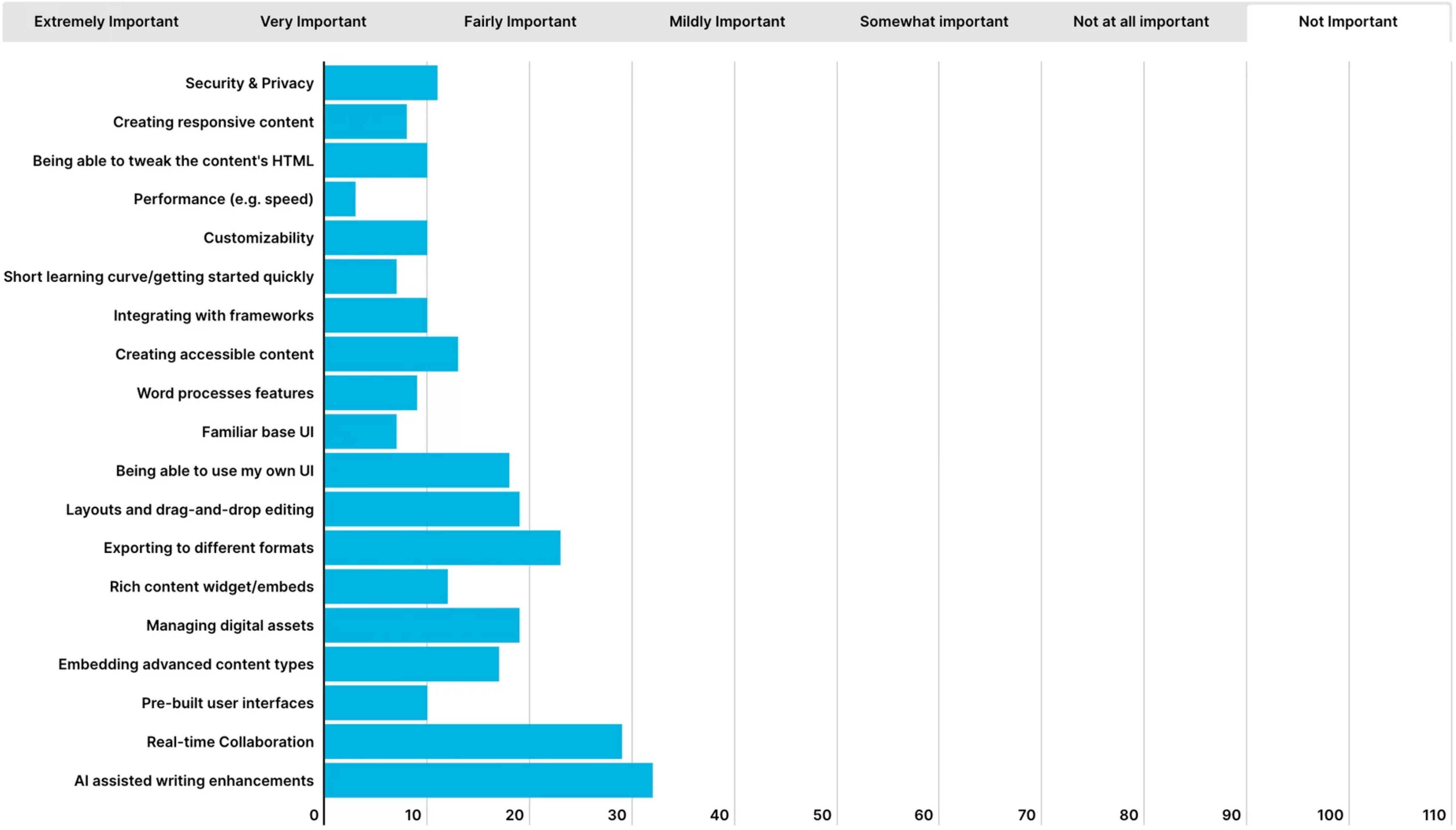
Number of responses. Click tabs to change results view



Number of responses. Click tabs to change results view



Number of responses. Click tabs to change results view



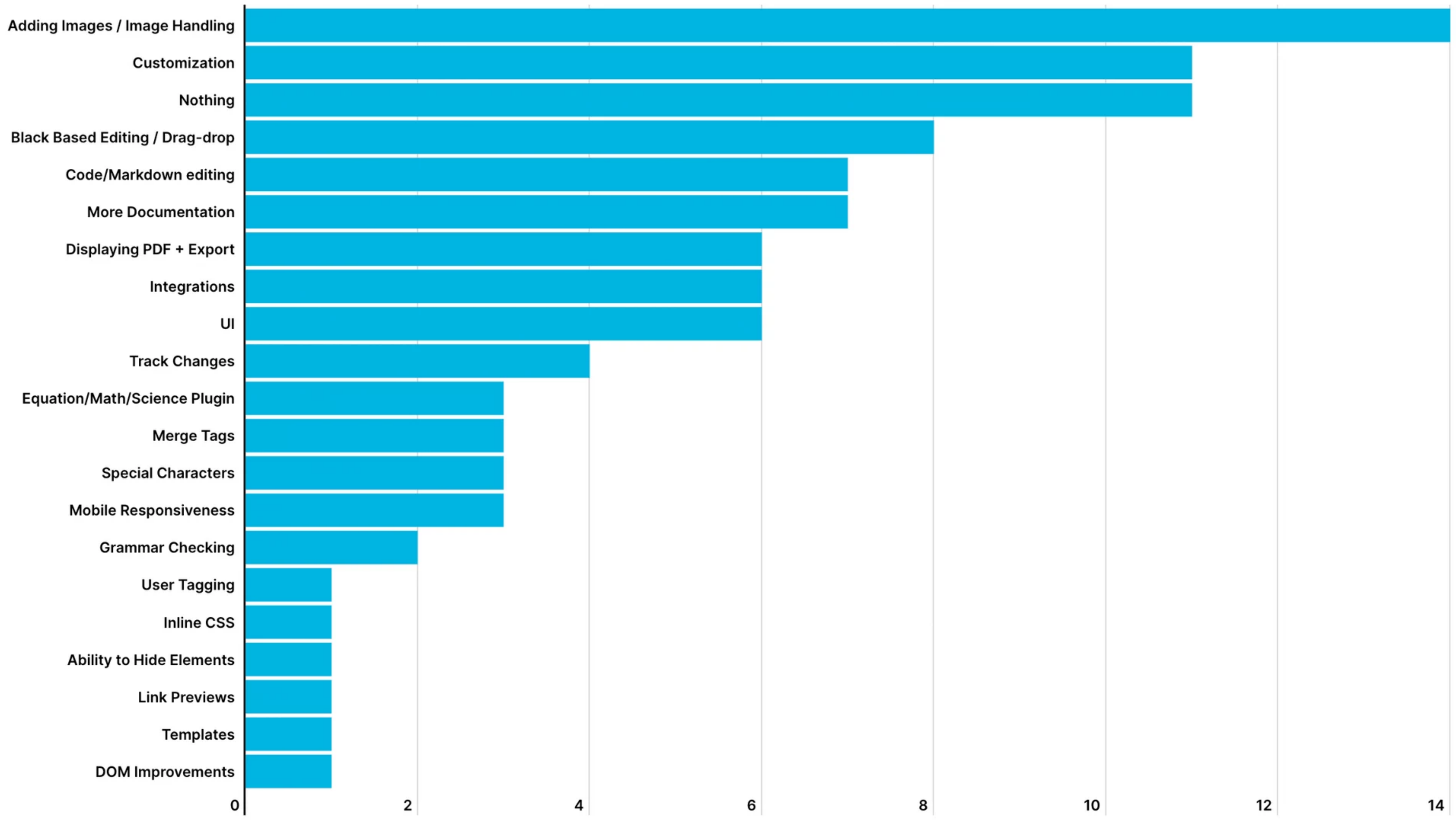
Number of responses. Click tabs to change results view

Based on your experience, what's the most important thing missing from your rich text editor?

This was a new open-text field question in 2022, to better understand what (desired) features were missing from RTEs. Based on what's known about the functionality of most editors on the market, they're all highly customizable. However, the fact respondents have noted 'Customization' as something that's missing could signal that poor documentation and low knowledge of the existence of certain advanced features is widespread.

Images rank as the most popular feature to improve — with respondents looking for more advanced ways to get images into their application, rather than what they're able to do with the images.

The responses have been summarized and categorized, to help shape the lens of view.



TINY TECHNOLOGIES

Tiny is the creator of **TinyMCE, the world's most trusted WYSIWYG component** that enables rich text editing capabilities within an application. Scalable, adaptable and reusable, it powers 100M+ projects worldwide and more than 1.5M+ developers use it to add velocity to their tech stacks, so they can build and ship their projects faster.

There's tens of thousands of market-leading applications powered by Tiny globally. It's helped SaaS companies, large enterprises, content creators and publishers to launch, grow and scale their businesses, reduce their development and technical debt burdens, minimize ongoing support tickets and boost the productivity of their users.