

Microsoft Learn: Quality Documentation

Foundational Study with Competitive Insights across AWS, GCP, and MS Docs

August 2025

Prepared by:
Jeremy Heyer

Agenda

01 Background & Key Findings

02 Study Design & Flow

03 Survey Portion

- a) Methodology
- b) Before we begin question
- c) MaxDiff Findings
- d) Free Selection questions
- e) Survey Summary

04 Comparative Portion

- a) Methodology
- b) General Findings
- c) Concept-Article Findings
- d) How-to Findings
- e) Troubleshooting Findings

05 What is Quality Documentation?

06 Key Recommendations & Next Steps

07 Appendix

Background & Key Findings

About this Research

Background

In the evolving landscape of technical documentation, AI developers rely on resources that not only inform but also streamline their workflows. As developers engage in unstructured, real-time learning, they depend on documentation that is clear, thorough, and highly accessible. Competitors such as Google, AWS, and Stripe have raised the bar with user-centric, interactive content, making it crucial to assess how Microsoft Docs measures up.

This study employs a three-method approach. First, a quantitative MaxDiff exercise will quantify the relative importance of specific documentation attributes (e.g., completeness, accuracy, readability, etc.). Second, a hands-on comparative analysis will investigate how developers navigate, locate, and engage with critical doc types—conceptual-article overviews, how-to guides, and troubleshooting resources—on AWS, GCP, and Microsoft Docs. Third, in-depth interviews will surface behavioral insights and perceptions around the concept of quality.

Through this research, we aim to explore how AI developers define documentation quality, where friction occurs, and what improvements would enhance usability. The findings will guide strategies to refine Microsoft Docs, ensuring it aligns with developer expectations and industry-leading standards for technical content.

Research Questions

- How do AI developers define “quality” in technical documentation? Which factors make docs useful, engaging, and effective?
- Which attributes—completeness, relevancy, readability, etc.—carry the most weight in AI developers’ evaluations of documentation?
- How do AI developers perceive UX affordances—such as navigational tools, layout elements, and interactive features—across conceptual overviews, how-to guides, and troubleshooting pages on Microsoft Docs, AWS, and GCP?
- How does overall AI developer satisfaction with AWS and GCP documentation compare to Microsoft Docs? What unique strengths and weaknesses surface for each platform?
- What challenges do AI developers face when using technical documentation in real-world workflows? Where do they encounter gaps, confusion, or inefficiencies?
- What strategic enhancements could Microsoft Docs implement to better support AI developers and become their preferred source for technical reference?

Participants

11 Users:

- AI Developers:
 - A mix of new to Azure (<6 months) & experienced with Azure (>6 months)
 - Experience with Azure docs and ecosystem
 - Experience with various technical documentation platforms (e.g., Stripe, Oracle, IBM, AWS, GCP, etc.)
 - Experience with Cloud Platforms
 - Experience building LLMs/Copilot/AI experiences
- A mix of genders, geo, roles, etc.

Method

- Survey (13 questions total including MaxDiff)
- Comparative Analysis across doc page types (3) and platforms (AWS, GCP, MSFT):
 - Concept-article
 - How-to
 - Troubleshooting
- Interviews around documentation quality

Previous Research

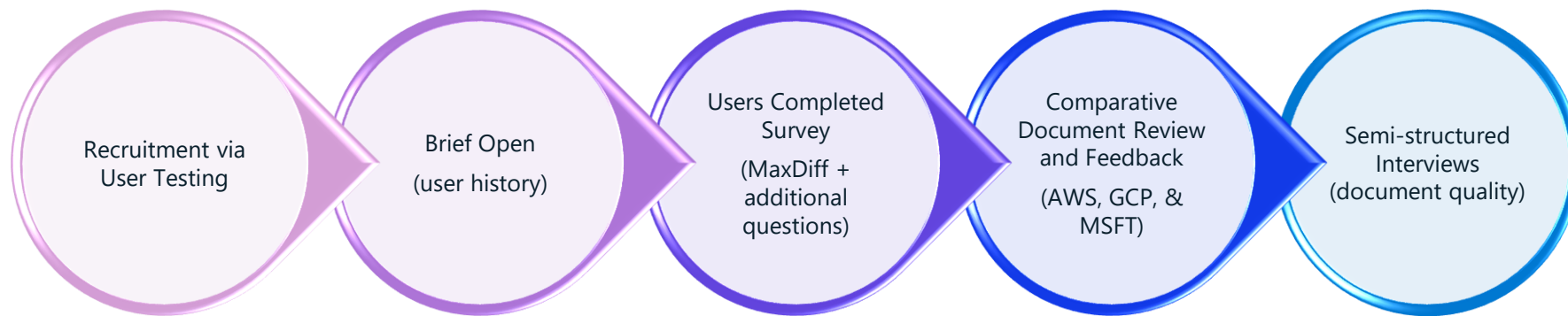
- [Ask Learn Experience on Docs](#)
- [FY25Q4 MS Learn Intercept Survey](#)

Key Findings

- 1. Microsoft docs are good, but not great yet:** Though technically accurate and thorough, Microsoft docs were often perceived as lacking key elements like code snippets, visual diagrams, and intuitive navigation.
- 2. Examples and Visual models emerged as users' top documentation priorities:** Whether new to Azure or experienced, all users identified examples (code samples, use cases) and easily digestible visuals (diagrams, flowcharts) as top priorities when engaging with technical documentation.
- 3. Microsoft docs were never recognized as the top preference compared to competitors:** Despite some positive performance and praise for robustness, Microsoft docs never ranked as the top experience due to high cognitive load, minimal visuals, and page complexity.
- 4. Amazon was preferred for two of three doc types due to its clean layout, visual-first design, and intuitive UX affordances:** Throughout the study, Amazon ranked as the most preferred technical doc across types, praised for its clean layout, clear visuals, and UX features like 'Focus mode' and a surfaced PDF download button.
- 5. Amazon's AI feature 'Summarize page' and 'Focus mode' played a key role in driving document preference:** When uncertain about the preferable doc, most users cited 'Summarize page' and 'Focus mode' as decisive features, noting their absence in competitors—even when technically available.
- 6. Quality documentation is defined by clarity, simplicity, strong visuals, and actionable guidance:** When asked what defined quality docs, users emphasized low cognitive load, clear problem framing, and simplicity—seeking guidance that's direct, visual, and free from information hunting.

Study Design & Flow

Study Design & Flow



Survey Portion

- ▶ a) Methodology
- b) Before We Begin Question Results
- c) MaxDiff Findings
- d) Free Selection & User Satisfaction
- e) Survey Summary



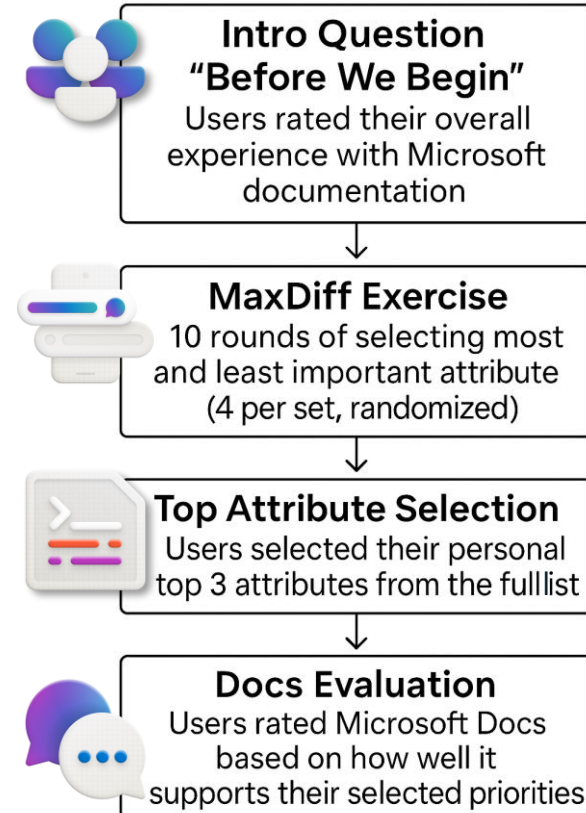
Methodology

Survey Format:

- 13-question survey administered via Qualtrics.
- Moderator launched and shared screen; users completed independently via screen control.

User Experience Sequence:

- Intro Question “Before We Begin” — Users rated their overall experience with Microsoft documentation.
- MaxDiff Exercise – 10 rounds of selecting most and least important attributes (4 per set, randomized).
- Top Attribute Selection – Users selected their personal top 3 attributes from the full list.
- Docs Evaluation – Users rated Microsoft Docs based on how well it supports their selected priorities.

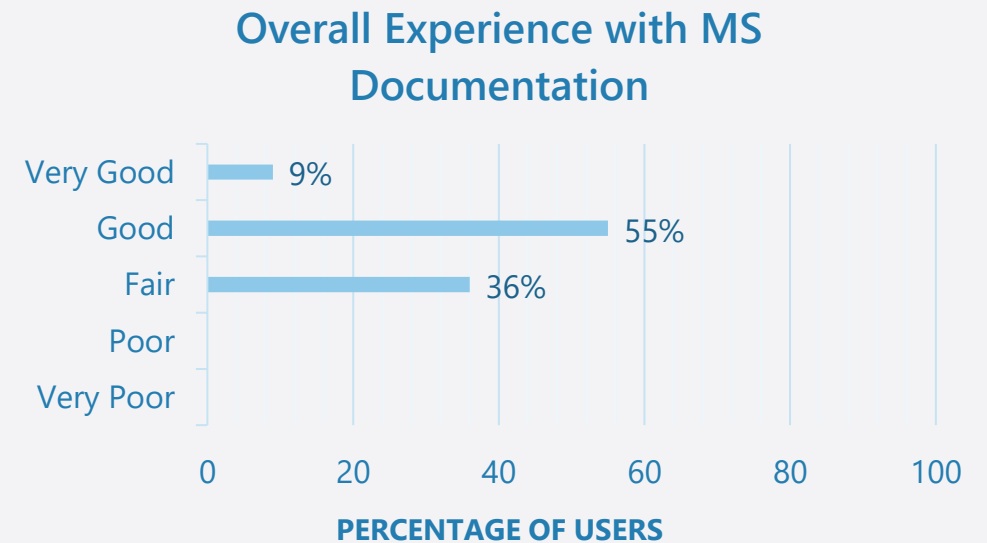


Survey Portion

- a) Methodology
- ▶ b) Before We Begin Question Results
- c) MaxDiff Findings
- d) Free Selection & User Satisfaction
- e) Survey Summary

Most users reported a Good (55%) or Fair (36%) experience on MS Docs, citing cognitive load and search friction as barriers to great and search friction as barriers to great

- Users cite cognitive load and search friction as barriers to deeper satisfaction. While respondents found the documentation technically correct and acknowledged its robust content, they struggled to locate and absorb relevant information quickly.
- Navigating to the right answers required disproportionate effort—particularly when trying to view content in context or follow logical paths across documentation layers ([FY25Q4 Learn Intercept Survey](#)).
- Unprimed feedback reveals intrinsic pain points. This question was conducted prior to any comparative evaluation (e.g., AWS, GCP, MSFT), meaning responses reflect raw, unframed experience with Microsoft Docs.
- The message is clear: even for technical users, correctness alone doesn't guarantee usability.



N=11

Survey Portion

- a) Methodology
- b) Before We Begin Question Results
- ▶ c) MaxDiff Findings
- d) Free Selection & User Satisfaction
- e) Survey Summary

Documentation Attributes & Survey Phrasing

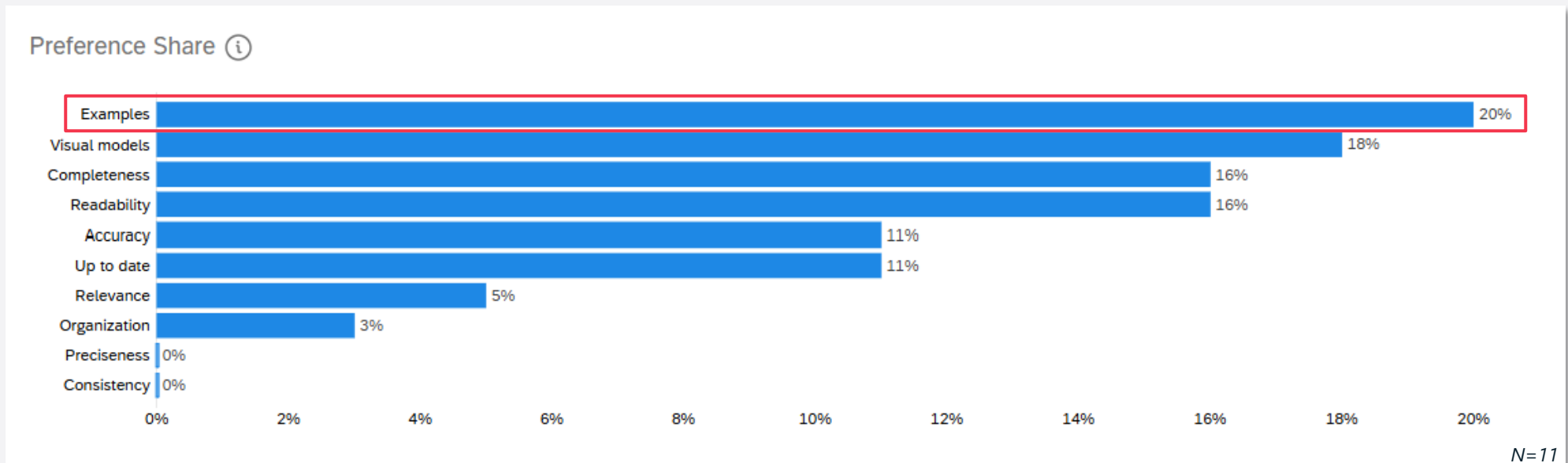
- We drew from Garousi et al. (2013), who identified documentation quality attributes in their study on software developers (*Evaluating Usage and Quality of Technical Software Documentation*).
- Our survey included 10 sets of trade-offs, 4 attributes per set, all randomized.
- The following insights in this section reveal the most important take aways and reflect how often a feature was chosen as most important or least important relative to other attributes.

Attribute	Survey Wording
Completeness	Covers all necessary details needed to understand and use the product or feature.
Organization	Well-structured with clear headings, logical flow, and easy navigation.
Visual models	Includes diagrams, flowcharts, or illustrations that help explain complex concepts.
Relevance	Focused on what I need to know—no unnecessary or outdated information.
Preciseness	Uses exact terms and avoids ambiguity; everything is clearly stated.
Readability	Easy to read, understand (jargon), and skim—even when technical.
Accuracy	Technically correct and aligned with the current product behavior.
Consistency	Uses consistent terminology, structure, and formatting across documents.
Up to date	Reflects the latest version of the product, API, or toolset.
Examples	Includes code samples or applied examples that show how to use the tool.

Document Attributes to MaxDiff Survey Wording

Developers prioritized actionable examples over consistency—highlighting a clear need for guidance that supports execution

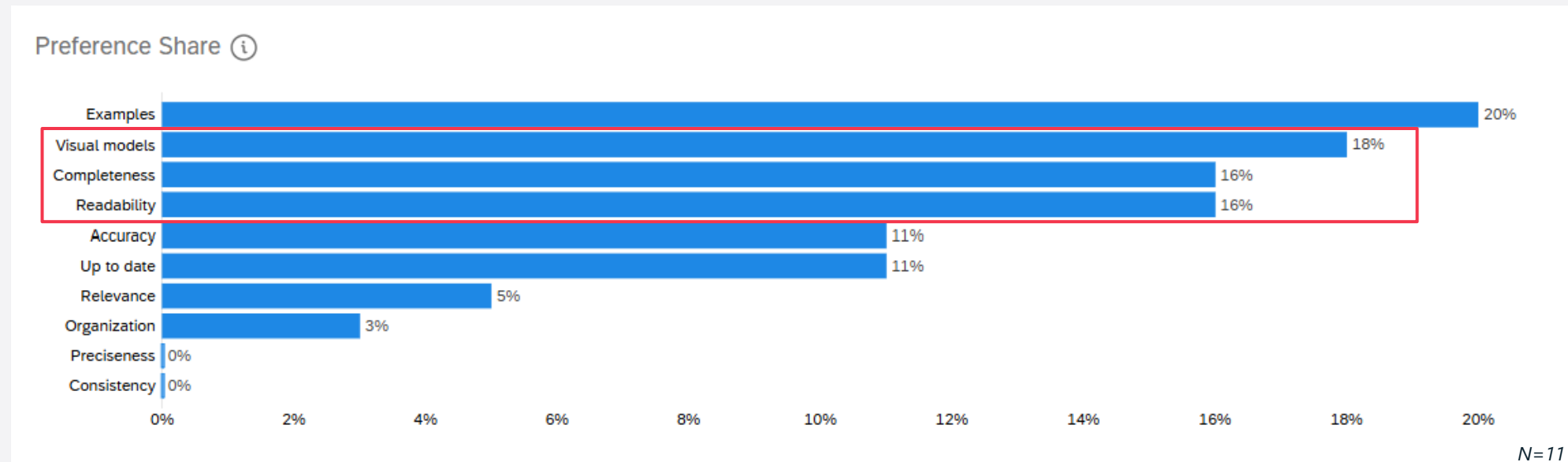
- Users consistently deprioritized 'Consistency' in forced-choice tasks—even after repeated exposure. Developers viewed fast guidance and quick solution-finding as critical.
- This insight aligns with insights from previous research ([Ask Learn Experience on Docs](#)) where prompts around summarization, code snippets and examples emerged as most useful.
- A 0% preference share means the feature was never preferred over others, regardless of how often it was shown, not that it isn't important.



Preference shares reflect responses from 11 users and offers directional insights into documentation quality priorities. Findings should be interpreted as early input rather than statistically representative. These are directional, not definitive.

Visual models, Completeness, and Readability stand out as key documentation attributes valued by developers

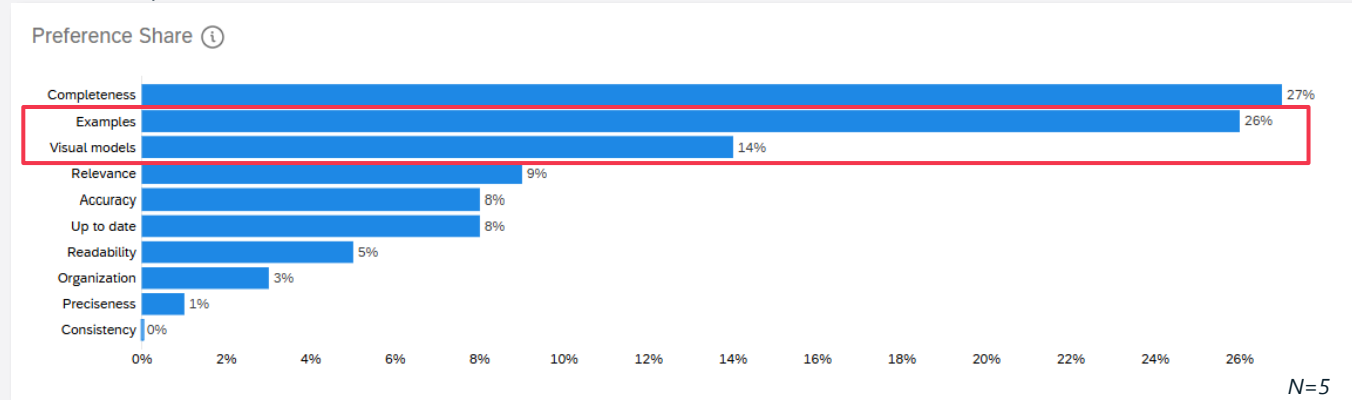
- Diagrams, flowcharts, and other visual aids help distill complex systems into understandable formats, allowing developers to quickly grasp architecture, workflows, and logic without wading through dense text.
- Completeness ensures developers have everything they need to build with confidence—from edge cases and error handling to system constraints and integration details.
- Pairing this thoroughness with Readability—clear language, scannable structure, and intuitive layout—makes technical content approachable, even when it's deeply complex.



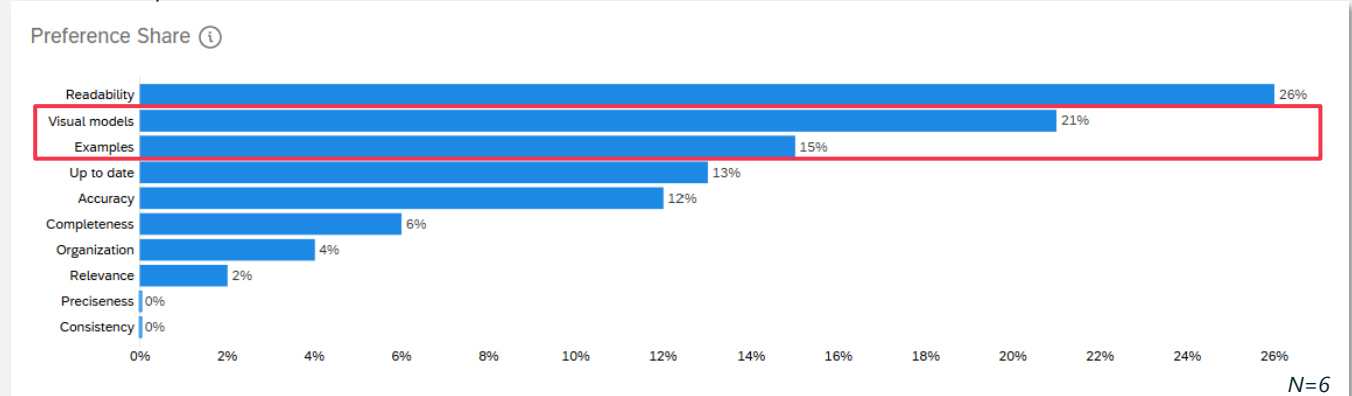
While completeness was most valued by those newer to the platform, users with more experience leaned towards readability as their top concern

- Segmenting Azure developers by tenure—less than 6 months vs. more than 6 months—revealed a consistent pattern: examples and visual models rank among the top three documentation attributes for both groups.
- For newer users, completeness emerges as the top priority. That means documentation must cover all necessary details needed to understand and use the product or feature to support onboarding and reduce friction.
- More experienced users, on the other hand, favor readability. With a stronger mental model, they're better at identifying correct solutions—so clear, scannable content helps them locate answers faster and stay productive.

<6 Months Experience



>6 Months Experience



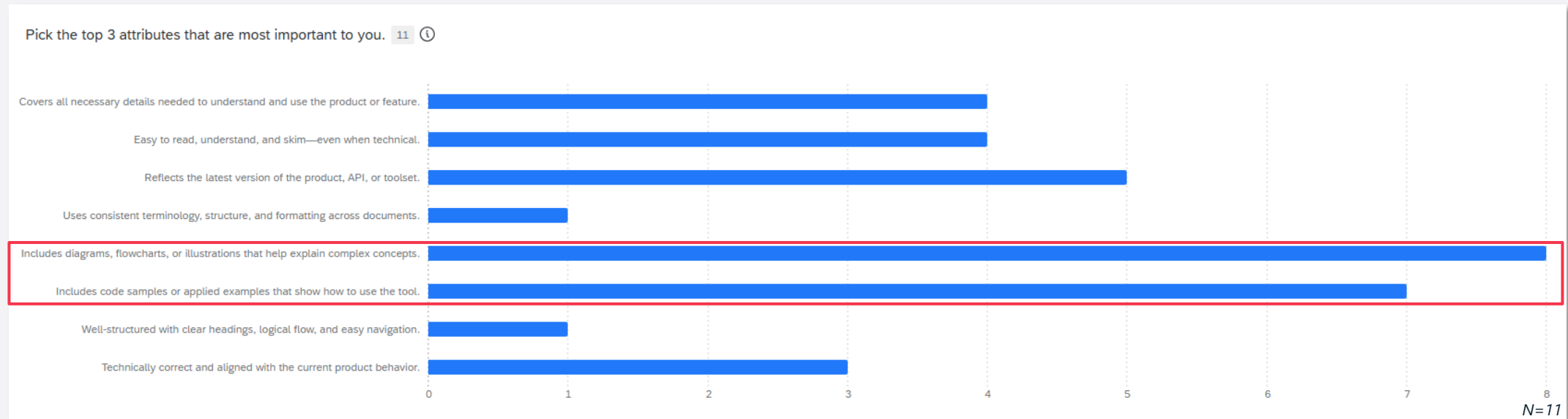
Above: Preference share for Azure users < 6 months. **Below:** Preference share for Azure users > 6 months.

Survey Portion

- a) Methodology
- b) Before We Begin Question Results
- c) MaxDiff Findings
- ▶ d) Free Selection & User Satisfaction
- e) Survey Summary

Even with free rein to pick any attributes, users still favored Examples and Visual models the most

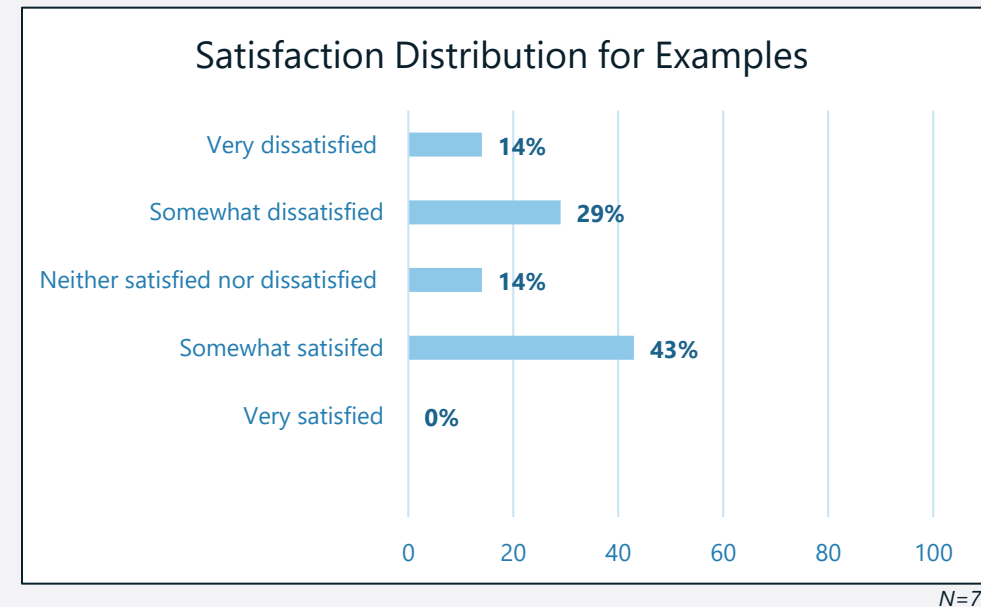
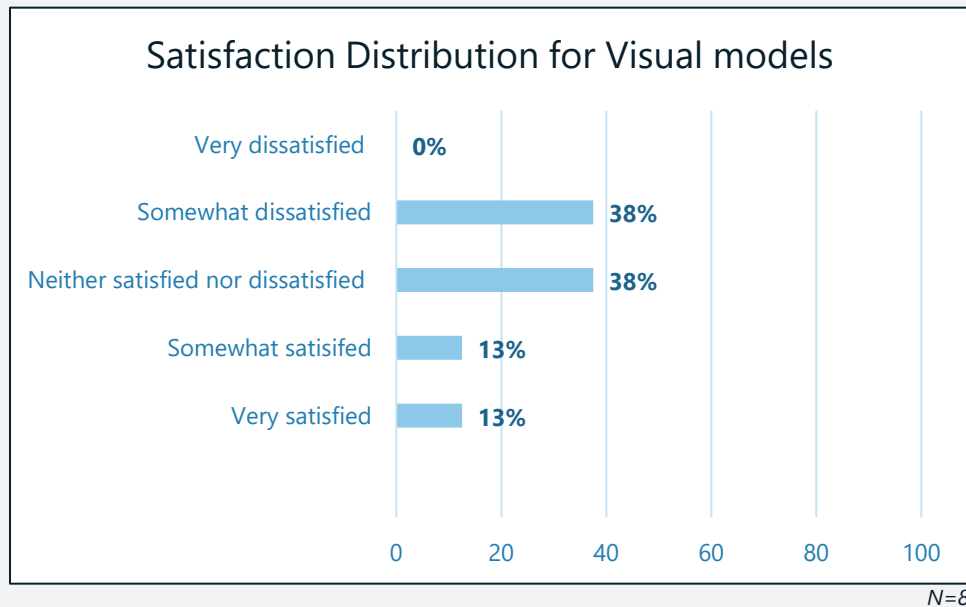
- In the free selection results, users gravitated towards Examples and Visual models.
- This repeated preference points to a natural pull: when given full freedom, users seek clear visuals and the utility to act above all.



Top 3 high priority attributes as free selected (uncontested) by users

High Demand, Low Impact: Visual models and Examples are missing the mark on satisfaction

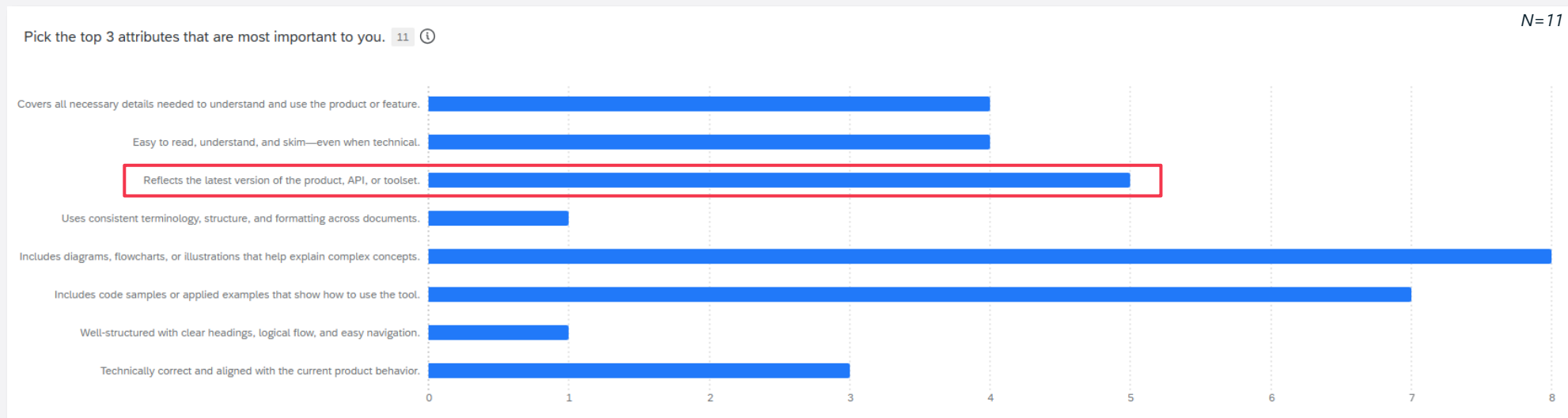
- When prompted to rate their satisfaction with these attributes in MS docs—satisfaction was low.
- Despite being a preferred form of output (content), Visual models satisfaction is tepid: 76% of respondents are either neutral or dissatisfied.
- Examples have slightly stronger satisfaction signals at 43% “somewhat satisfied,” but no one is “very satisfied” and there is a notable 14% very dissatisfied.



Note: Satisfaction distributions based on users' free selection. Percentages presented are out of identified sample size (n), not the entire study population (n=11)

Freshness didn't win the popularity contest, but it still earned a seat at the table

- When attributes are compared in tight, forced trade-off sets, users pick what really jumps out to them against the others. Freshness or the up-to-date attribute didn't break the top 3 because it was routinely beaten out by stronger contenders in head-to-head choices.
- Yet in unconstrained selections, "up-to-date" surfaced as a consistent pick, highlighting its perceived importance despite limited decisiveness.
- While not a primary driver, freshness plays a meaningful role and should be thoughtfully integrated into documentation when resourcing allows.



Top 3 high priority attributes as free selected (uncontested) by users

Survey Portion

- a) Methodology
- b) Before We Begin Question Results
- c) MaxDiff Findings
- d) Free Selection & User Satisfaction
- ▶ e) Survey Summary



Insight Across Methods

- **Examples** anchor user trust
 - Tops MaxDiff
 - Ranks #2 in free selection → Signals that hands-on, applied guidance is non-negotiable
- **Visual Models** drive conceptual clarity
 - Ranks #2 in MaxDiff
 - Ranks #1 in free selection → Illustrations, diagrams, and flowcharts accelerate understanding
- **Readability & Completeness** unlock widest reach
 - Rank tied #3 in MaxDiff
 - Both attributes land in free picks → Well-structured, thorough docs satisfy skimmers and deep divers alike
- **Up-to-date (freshness)** builds confidence
 - Reflects the latest version of the product, API, or toolset
 - Surfaces *strongly* only in free selection but is a critical trust signal → Users want to feel assured content is current

Comparative Portion

- ▶ a) Methodology
- b) General Findings
- c) Concept-Article Findings
- d) How-to Findings
- e) Troubleshooting Findings

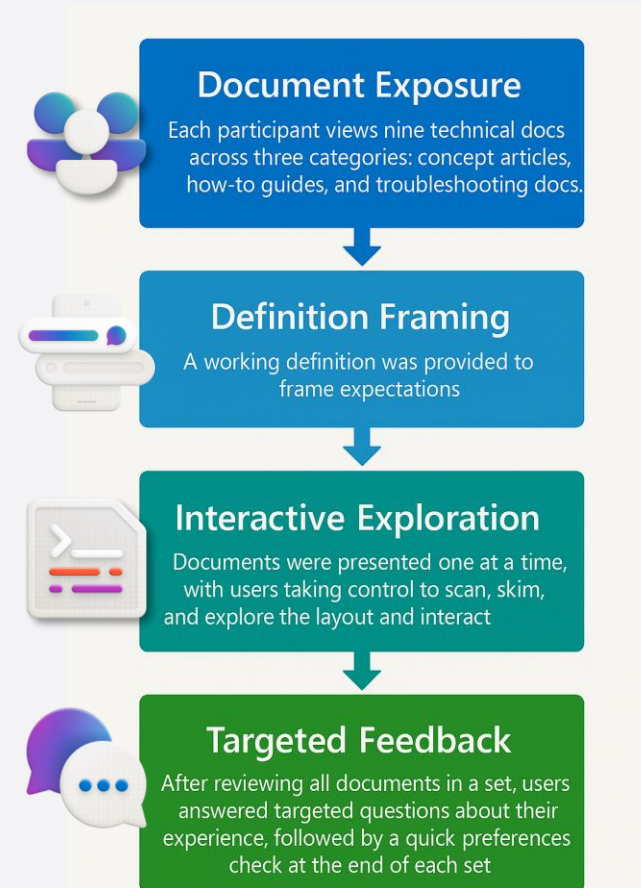


Methodology

Each participant was shown nine technical documents across three categories: concept articles, how-to guides, and troubleshooting docs.

For each set, a working definition was provided to frame expectations. Documents were then presented one at a time, with users taking control to scan, skim, and explore the layout and interactive elements.

After reviewing all documents in a set, users answered targeted questions about their experience, followed by a quick preferences check at the end of each set.



Comparative Portion

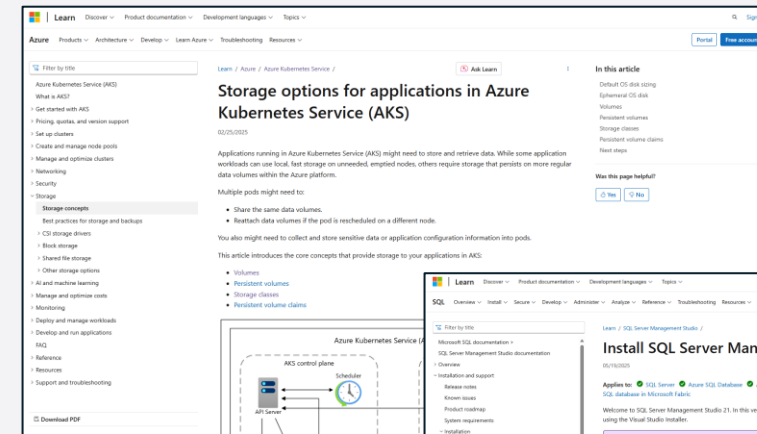
- a) Methodology
- ▶ b) General Findings
- c) Concept-Article Findings
- d) How-to Findings
- e) Troubleshooting Findings

Across all three document types, Microsoft never ranked first in user preferences primarily due to overly complex page layouts, cognitive load, and the inability to find content quickly

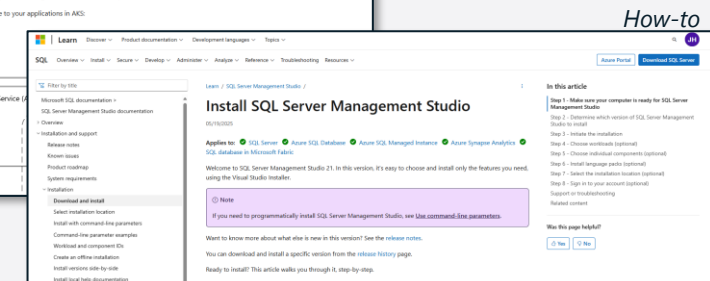
- Throughout the study, users were asked to identify which experience best aligned to their expectations of various doc types.
- For each type, all users preferred Google or Amazon over Microsoft.
- Primary reasons for this was cognitive load, over complex page layouts, and inability to quickly find content without extra actions (Ctrl+F).
- Low satisfaction with Microsoft doc types, in particular troubleshooting and how-to surfaced in [FY25Q4 Learn Intercept Survey](#) where both ranked towards the bottom.

Recommendation:

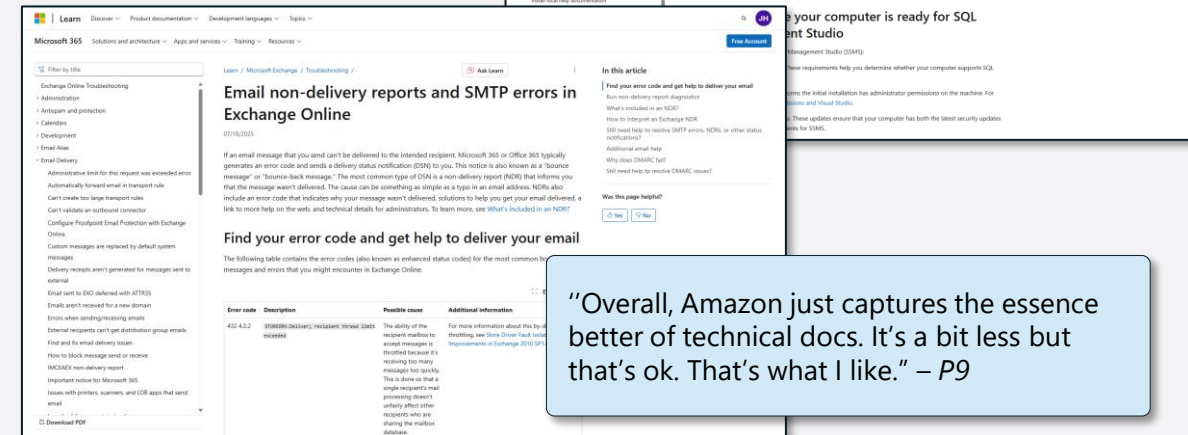
- Explore design solutions that decrease page density such as 'Focus mode'.



Concept-article



How-to



Troubleshooting

“Overall, Amazon just captures the essence better of technical docs. It’s a bit less but that’s ok. That’s what I like.” – P9

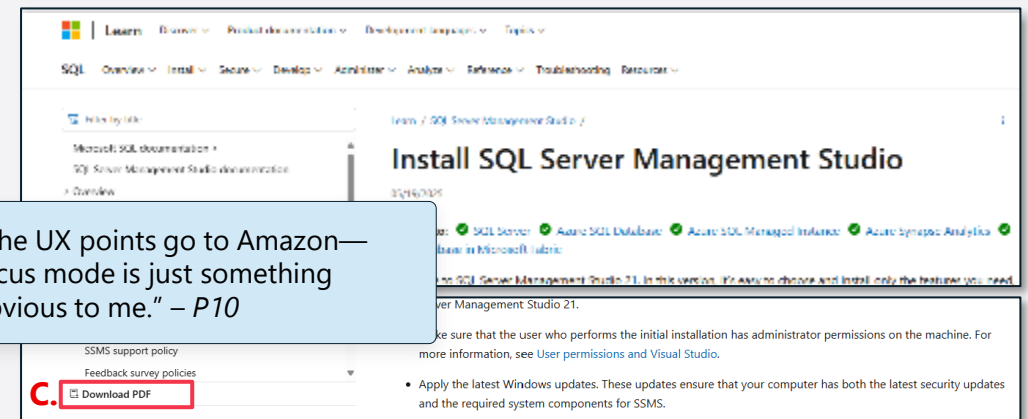
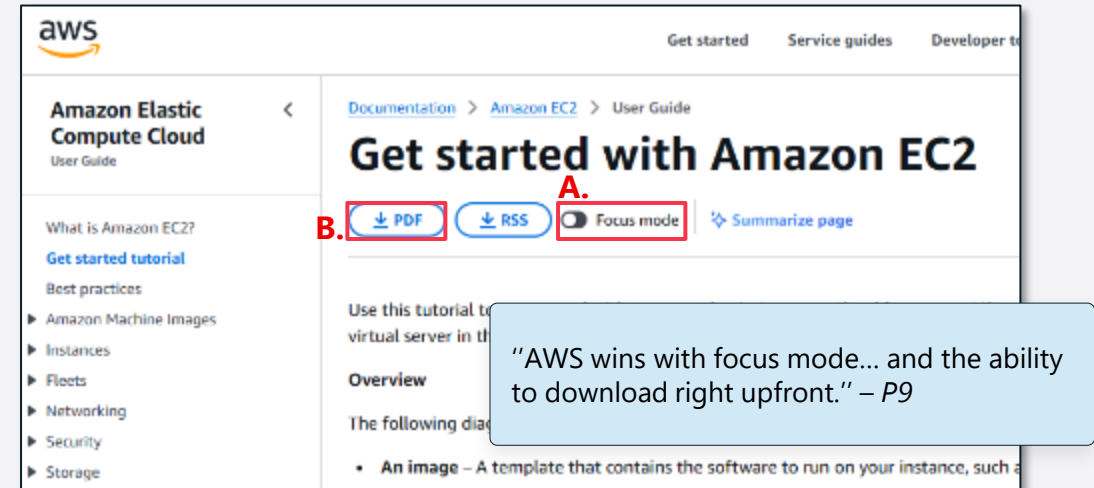
The presence of 'Focus mode' and PDF download at the top of the page played a significant role in Amazon's overall preference across the study

- For users, the ability to quickly expand the page content via 'Focus mode' [A] was critical to readability and being able to quickly skim page content.
- This clear CTA removed cog load and page complexity by streamlining the presentation of content and data.
- Additionally, the placement and ability to quickly download a doc [B] was considered valuable and needed for content digestion.
- For Microsoft, the download button is nested in the bottom left corner [C] and was often missed. For Google, a user must Ctrl+P and then save the document.
- Current design initiatives have implemented 'Focus mode' into Learn, showing a strong alignment to user needs.

Recommendation:

- Consider surfacing the 'Download PDF' entry point towards the top of the page providing quick access and improved visibility.
- Explore providing a 'Focus mode' granting users the ability to clear the ToC and expand the page content removing clutter.

Amazon: PDF download and Focus mode



Microsoft: PDF download

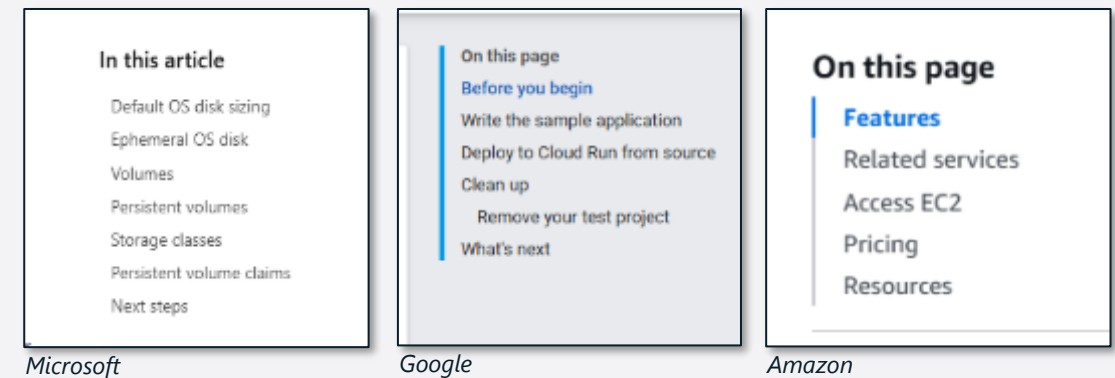
Right-rail structure outperforms ToCs with faster navigation, better context, and leaner layout

- Users gravitated toward right-rail elements like 'In this article' and 'On this page,' consistently describing them as cleaner and more intuitive than traditional ToCs. Despite having similar structural goals, right-rail navigation offered a faster, less cluttered experience.
- While sticky ToCs provided baseline value, especially within-document, their utility tapered quickly. Users cited difficulty scanning dense hierarchies and a lack of spatial clarity compared to the lightweight, streamlined anchors in the right rail.
- The effectiveness of right rail content can be attributed to:
 - **Always-visible** anchors make jumping between sections lightning fast
 - **Contextual awareness** helps users understand where they are in the page hierarchy
 - **Compact structure** takes up less mental space than a sprawling ToC

Recommendation:

- Explore design solutions that auto-collapse the 'Table of Contents' on page load or provide users a cleaner view of core content while minimizing visual clutter.

"I might look at the ToC initially but, in the end, I go right to the 'in this article' section." – P5



"I only interact with it [ToC] if it's a long doc, otherwise it's the side area here [right-rail]." – P3

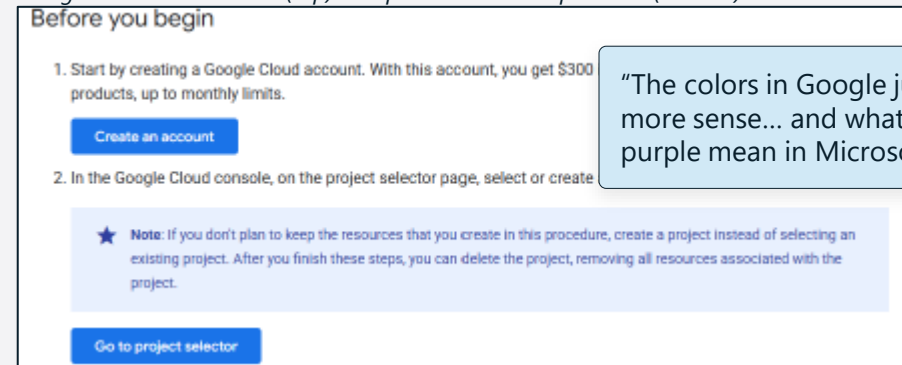
Color-coded elements — such as warnings, entry points, links, and editable code — had a strong influence on users' perceptions of ease of use and readability

- When navigating any of the doc pages, users were drawn to color-coded content as it created a natural separation within the doc, decreasing cognitive load and providing strong focus on actions to take.
- All platforms provided color-coded notes and warnings which were considered critical. However, all users stated that Google's use of color was preferred as notes, buttons and editable code snippets [A] had strong contrast and led to direct actions, actions that led to potentially solving their problem.
- Microsoft user of color was commended however some users questioned the assignment of color and action. For example, 'purple' for notes [B] and 'blue' for important stating that they didn't seem to align to any sense of urgency.

Recommendation:

- Consider aligning color choices with user expectations to enhance clarity and perceived urgency.
- Experiment with color treatments that reinforce the interactivity and actionability of key elements.

Google: Blue note bubble (top) and pink edit code experience (bottom)



"The colors in Google just make more sense... and what does the purple mean in Microsoft?" – P8

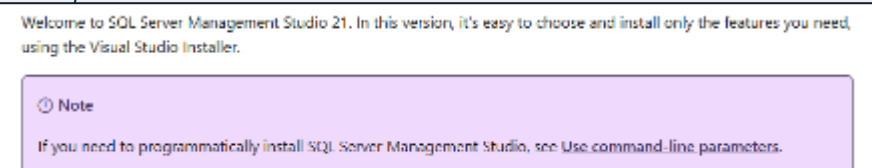
A.

```
gcloud config set project asdfasd
```

"I like the blue buttons and the interactivity of the '+' icons... The colors make it easy to read. They are helpful." – P11

Microsoft: Note indicator

B.



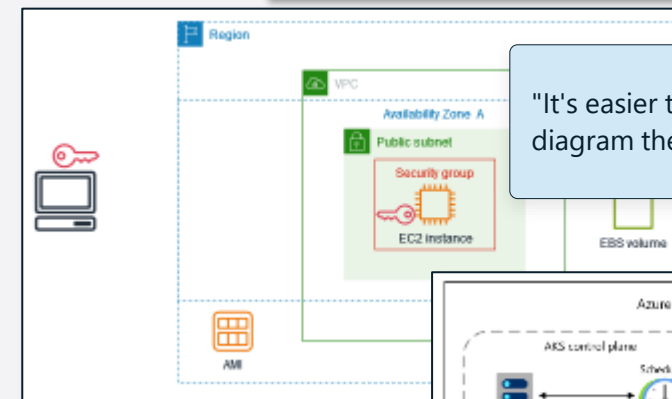
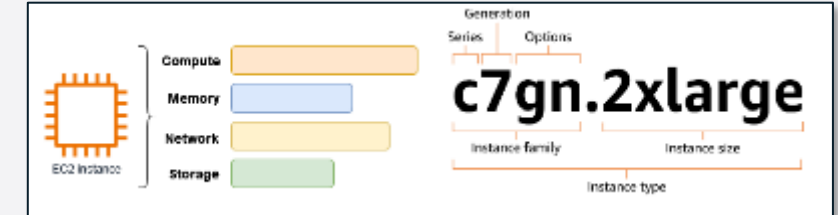
Diagrams distilled dense information into digestible insights—bridging the gap between technical depth and user clarity

- Diagrams emerged as a decisive factor in user content preference. In the MaxDiff survey, they ranked near the top, signaling their vital role in how users absorb information.
- When diagrams were present, users gravitated toward that content, finding it clearer and easier to understand.
- Their absence—particularly in Google’s documentation—was frequently mentioned as a shortfall, underscoring the expectation for visual aids in technical communication. In short, diagrams weren’t just helpful—they were a dealbreaker.

Recommendation:

- Consider surfacing visual aids early in documentation (where/when appropriate), ensuring that users readily and easily locate them when skimming.

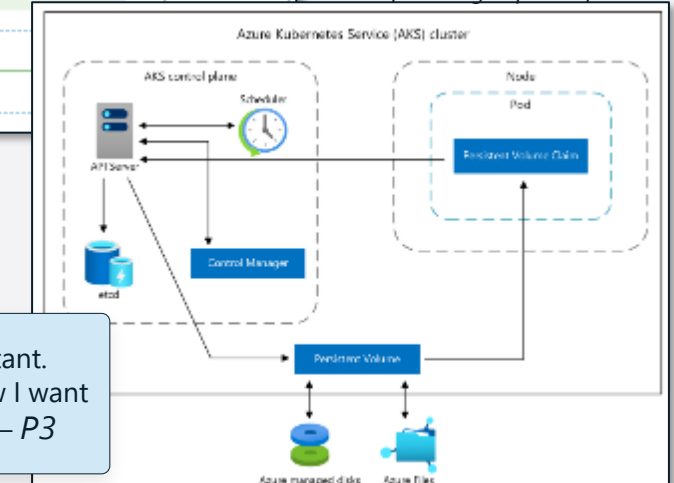
Amazon: What is Amazon EC2



Amazon: Get Started with EC2

"It's easier to explain something in a diagram than it is to write it all down." – P8

Microsoft: Storage options for AKS



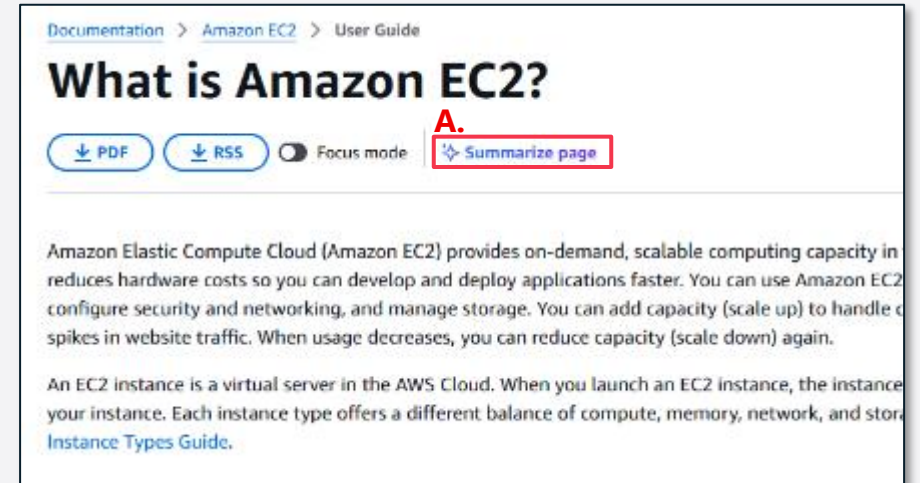
"Diagrams are the most important. I'm a visual learner and it's how I want information presented to me." – P3

Without prompting, users questioned what actions 'Ask Learn' performed and if it was tied to the document or Learn

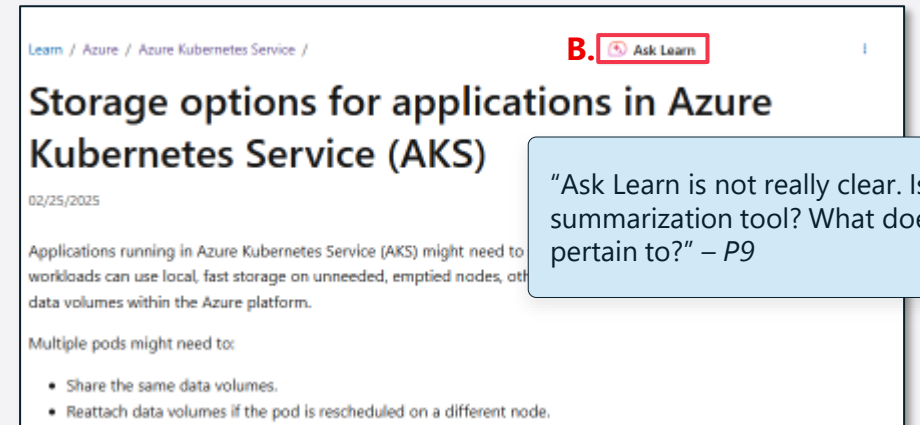
- When skimming the concept pages, users quickly identify the AI experiences for both Amazon and Microsoft stating it was obvious and needed.
- However, users preferred Amazon because it was clear what the AI did—summarize page [A]. This need for summarization surfaced in previous research ([Ask Learn Experience on Docs research](#)) indicating a need for a clear CTA for this feature.
- When viewing the 'Ask Learn' entry point, users identified it as AI but were unsure of what it did [B]. Upon click, some users said its just too much to do (compared to AWS) and they wanted to know its function upfront, without investment.

Recommendation:

- Explore design solutions that provide users with clear functionality of 'Ask Learn' (value propositions).
- Explore design solutions that let users generate an AI-powered summary directly from a tech doc page—eliminating the need to open 'Ask Learn' and manually select the 'Summarize page' prompt.



Amazon: Summarize page AI feature



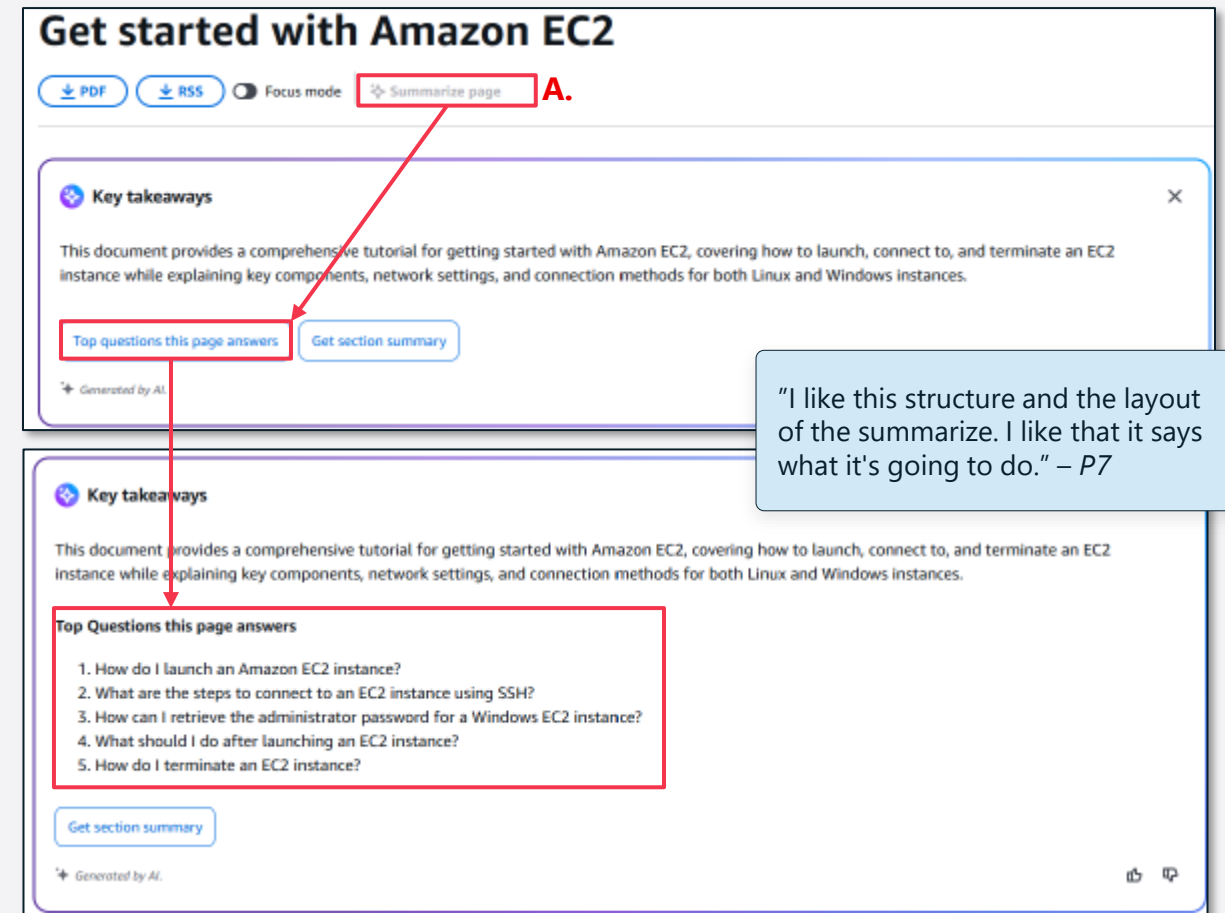
Microsoft: Ask Learn AI feature

Amazon's 'Summarize page' feature—delivering instant document overviews and contextual follow-up prompts—was positioned as a standout capability that set it apart

- Users consistently favored Amazon's 'Summarize page' AI feature [A] for its clarity, placement, and focused functionality.
- Unlike Microsoft's 'Ask Learn' experience, which appeared as a side car (upon click), Amazon's summarization feature was centrally embedded and clearly labeled—making its purpose immediately obvious.
- The direct call-to-action, paired with relevant follow-up prompts, created a streamlined and intuitive experience that users found easier to engage with and more effective for digesting documentation.

Recommendation:

- Explore design solutions that let users generate an AI-powered summary directly from a tech doc page—eliminating the need to open 'Ask Learn' and manually select the 'Summarize page' prompt [repeat].
- Explore user sentiment regarding additional AI features such as page summarization while maintaining 'Ask Learn' side car experiences, offering users the best of both worlds.



Amazon: Summarize page AI feature

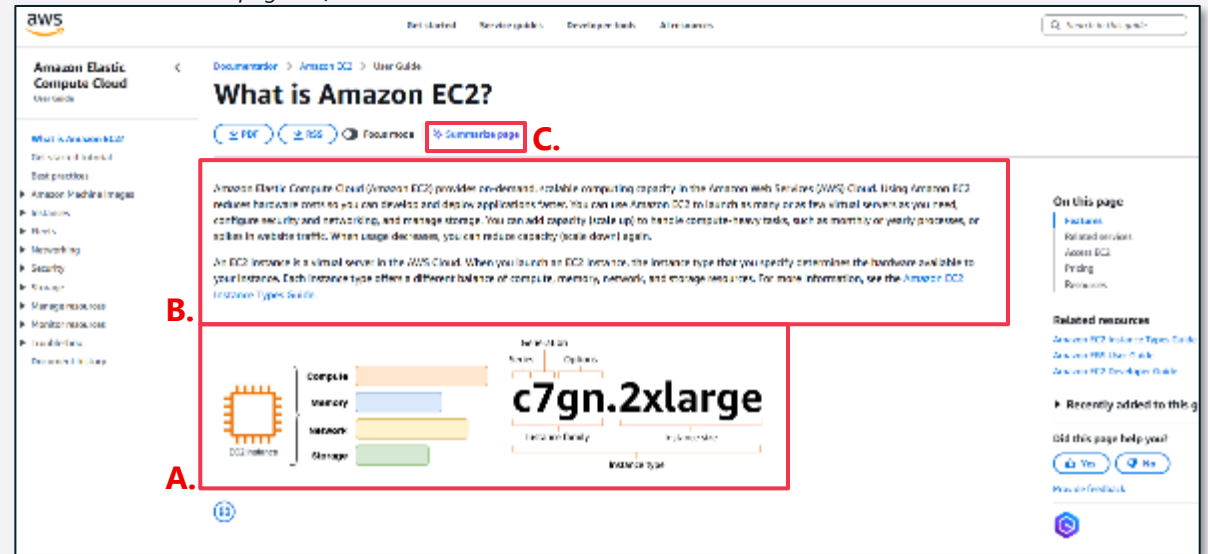
Comparative Portion

- a) Methodology
- b) General Findings
- ▶ c) Concept-Article Findings
- d) How-to Findings
- e) Troubleshooting Findings

Diagrams, 'Focus mode', and clear CTA to AI made Amazon's conceptual-article doc the clear favorite

- Amazon's conceptual documentation stood out by prioritizing user clarity from the start. A bold, colorful diagram led the page, offering immediate visual context that made complex ideas feel approachable [A].
- Right above that was a to-the-point summary [B]. No fluff, no scrolling required. Users could absorb the core message within seconds, reinforcing speed and confidence in the content.
- The experience was wrapped in functionality: a clear 'Summarize page' CTA gave the AI obvious utility, guiding users toward fast content digestion [C].
- 'Focus mode' eliminated distractions, and a nested top-of-page download button made it easy to save and revisit. Together, these features created a seamless, highly preferred documentation flow.

Amazon: Summarize page AI feature



"Everything else is just more cluttered. I just want to know the answer and go. Amazon gives a more clear UI and the diagram is nice." – P1

"Microsoft just feels a bit clumsy to me. You can differentiate easily between the sections [AWS]. It's just very easy to read." – P4

Users gravitated toward brief, bulleted summaries paired with concise text—favoring Microsoft and Amazon for combining clarity with speed

- Top summaries at the start of docs help users immediately orient themselves—saving time and cognitive load. When well-crafted, they serve as a quick scan point that validates whether the content is relevant or worth deeper reading.
- Users consistently responded better to formats that were short, bulleted, and plainspoken. Microsoft and Amazon approach struck the right balance—providing high-level context and clarity upfront without overwhelming the reader.
- The best experiences are seamless, offering just enough signal to keep users moving efficiently.

Microsoft: Top doc summary

The screenshot shows the top of a Microsoft documentation page. The title is 'Storage options for applications in Azure Kubernetes Service (AKS)' in a large, bold font. Below the title is a date '02/25/2025'. The main text block starts with a paragraph: 'Applications running in Azure Kubernetes Service (AKS) might need to store and retrieve data. While some application workloads can use local, fast storage on unneeded, emptied nodes, others require storage that persists on more regular data volumes within the Azure platform.' This is followed by a section 'Multiple pods might need to:' with a bulleted list: '• Share the same data volumes.' and '• Reattach data volumes if the pod is rescheduled on a different node.' Below this is another paragraph: 'You also might need to collect and store sensitive data or application configuration information into pod...'. At the bottom of the visible section, it says 'This article introduces the core concepts that provide storage to your applications in AKS:' followed by a bulleted list: '• Volumes', '• Persistent volumes', '• Storage classes', and '• Persistent volume claims'.

"I want bullets so I can scan and just go." – P5

Get started with Amazon EC2

The screenshot shows the top of an Amazon documentation page. The title is 'Get started with Amazon EC2' in a large, bold font. Below the title are links for 'PDF', 'RSS', 'Focus mode', and 'Summary page'. The main text block starts with a paragraph: 'Use this tutorial to get started with Amazon Elastic Compute Cloud (Amazon EC2). Your virtual server in the AWS Cloud. With Amazon EC2, you can set up and configure the...'. This is followed by an 'Overview' section with the text: 'The following diagram shows the key components that you'll use in this tutorial:'. Below this is a bulleted list: '• An image – A template that contains the software to run on your instance, such as...', '• A key pair – A set of security credentials that you use to prove your identity when a private key is on your computer.', '• A network – A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. To help you get started quickly, your account comes with a default VPC in each AWS Region, and each default VPC has a default subnet in each Availability Zone.', '• A security group – Acts as a virtual firewall to control inbound and outbound traffic.', and '• An EBS volume – We require a root volume for the image. You can optionally add data volumes.'

"Microsoft is the preferable experience. I like this up front; it's almost like a case summary. AWS and Microsoft definitely lead here overall." – P7

Amazon: Top doc summary

Comparative Portion

- a) Methodology
- b) General Findings
- c) Concept-Article Findings
- ▶ d) How-to Findings
- e) Troubleshooting Findings

Google's docs were designed for flow—developers could jump in, grab what they needed, and move fast

- From the moment users opened a page, they found clean entry points like 'Before You Begin' and copy-paste-ready code blocks.
- Clickable blue buttons, editable code snippets, and icon-based cues helped users act without second-guessing. These features weren't just aesthetic—they accelerated deployment and made even complex steps feel intuitive.
- Google's use of spacing, color, and consistent formatting made scanning fast and execution easy.
- Compared to boxy layouts or buried steps on other platforms, Google's docs felt lighter, more helpful, and purpose-built for real developers under time pressure.

Recommendation:

- Review current spacing, hierarchy, and formatting to ensure users can visually scan content and pinpoint solutions without minimal friction.
- Consider adding clickable buttons, collapsible sections, and inline menus to streamline navigation and keep content concise without sacrificing depth.

Google: Build and deploy a Python web app to Cloud Run

Before you begin

1. Start by creating a Google Cloud account. With this account, you can create Google Cloud products, up to monthly limits.

Create an account

2. In the Google Cloud console, on the project selector page, select or create a Google Cloud project.

★ **Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

Go to project selector

3. Make sure that billing is enabled for your Google Cloud project.

4. Make sure that you have the following role or roles on the project: Cloud Run Admin, Cloud Run Source Developer, Logs Viewer

Check for the roles

- a. In the Google Cloud console, go to the IAM page.

Go to IAM

- b. Select the project.

- c. In the **Principal** column, find all rows that identify you or a group that you're included in. To learn which groups

"I just want to know the link and what I need—Google gives me that." – P1

"The steps, the color, and the layout—it's all just more usable." – P10

"The blue button? That's the thing Google does that's actually helpful." – P8

column to see whether the list of roles includes the

Real-time code editing within documentation emerged as a key capability—highly relevant and directly aligned with developer needs and expectations

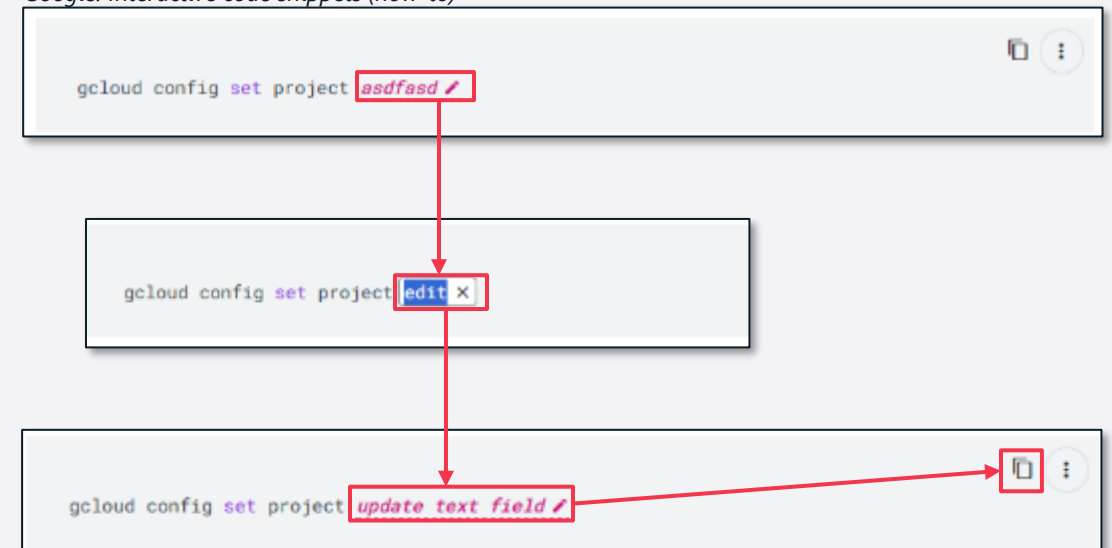
- Users consistently noted Google’s real-time code snippet editing feature, marked by a pink highlight and pencil icon.
- They appreciated how simple it was to edit and copy code—calling it “just this easy.”
- Throughout the study, this functionality was repeatedly praised as developer-focused and thoughtfully designed.

Recommendation:

- Explore engineering solutions that would allow users to edit and copy code directly in the document removing the need to edit later in their workflows.

“I like this. It’s easy for any developer to start. All sample code and commands which allows me to deploy. This is a cakewalk for a developer!” - P10

Google: Interactive code snippets (how-to)



“The Google doc is more for developers. More code. Less text.” – P3

Comparative Portion

- a) Methodology
- b) General Findings
- c) Concept-Article Findings
- d) How-to Findings
- ▶ e) Troubleshooting Findings

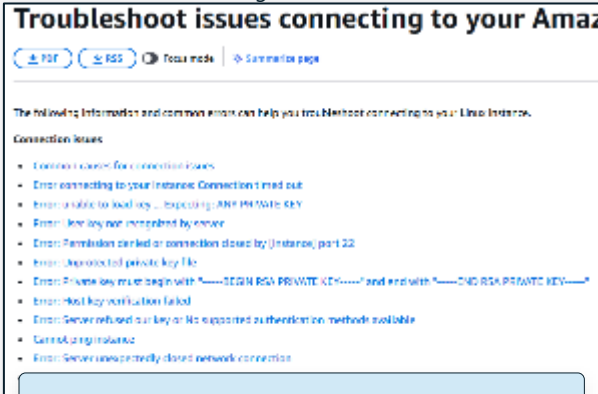
AWS was the preferred troubleshooting doc for its intuitive structure, clear resolution paths, and low-friction navigation that minimized cog load and accelerated problem-solving

- Users consistently preferred Amazon’s troubleshooting docs for their modular design, which surfaced common issues and hyperlinked errors prominently at the top of the page.
- Many users appreciated the well-separated sections, streamlined navigation, and low cognitive load that made resolving errors fast and straightforward.
- In contrast, the Microsoft troubleshooting doc was described as overly dense, noisy and a dump of information. It was considered difficult to scan, with large tables and limited contextual support.
- Google’s approach landed positively for its edit-and-copy features and multilingual code examples, though some found the layout sparse.

Recommendation:

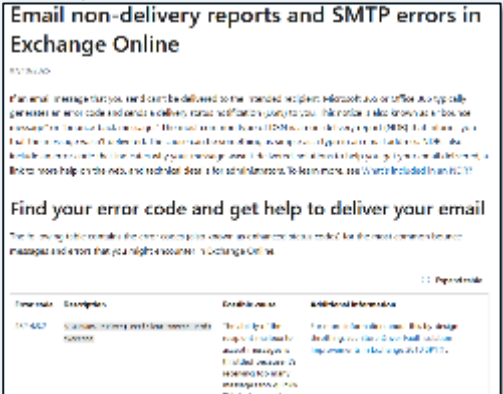
- Consider surfacing common issues at the top through clear hyperlinks and headings to support faster error resolution.
- Explore ways to reduce cognitive load by minimizing unnecessary content and using interactive elements to nest in-depth material that might otherwise overwhelm or bury key information.

Amazon: Troubleshooting links



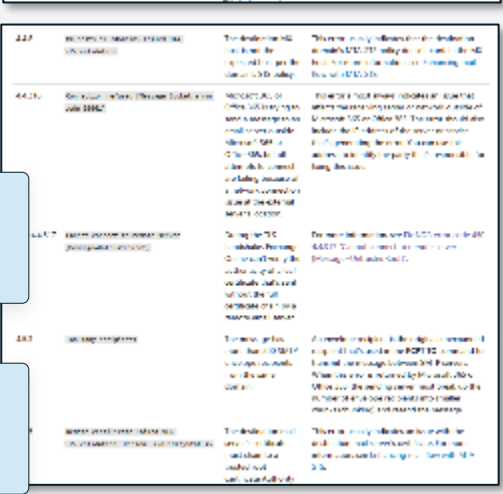
"I can get to the point quickly. Two clicks and I'm at the error description." - P7

Microsoft: Error code table



"It's [Microsoft] a dump of information. It doesn't read well. Its just boring." - P3

"Microsoft was so dense, and the ToC had all those codes, and I didn't really get it. The screenshots at the end felt like way too much." - P8



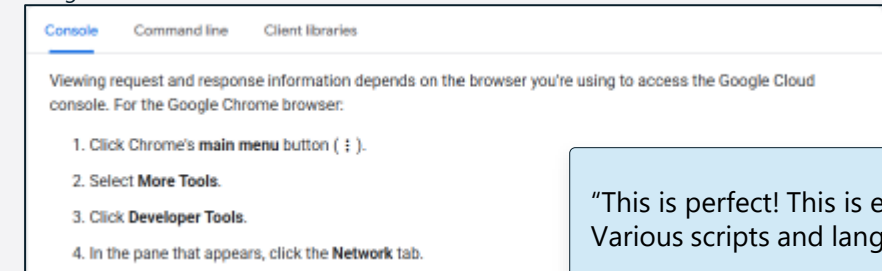
'Console', 'Command line', and 'Client libraries' were identified as developer friendly and critical to solutioning

- Users consistently praised Google's 'Client libraries' for minimizing translation overhead across languages—critical for saving time and enabling quick implementation.
- Removing the need to copy and modify was identified as something developers crave.
- Google continuously offers editable, copyable command line snippets directly in their docs. This supports instant application with minimal friction—an experience developers cited as ideal.

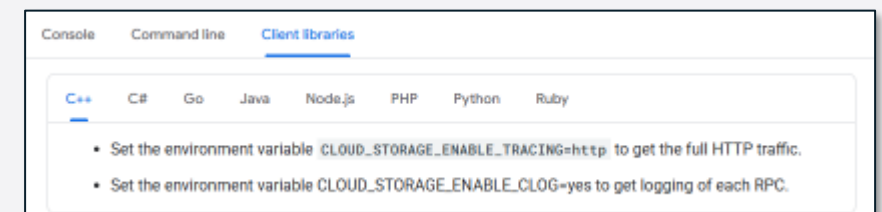
Recommendation:

- Explore design and engineering solutions that incorporate client libraries and command line interactions to reduce implementation friction and accelerate troubleshooting.

Google: Console, Command line, & Client libraries



"This is perfect! This is exactly what I want. Various scripts and languages." - P7

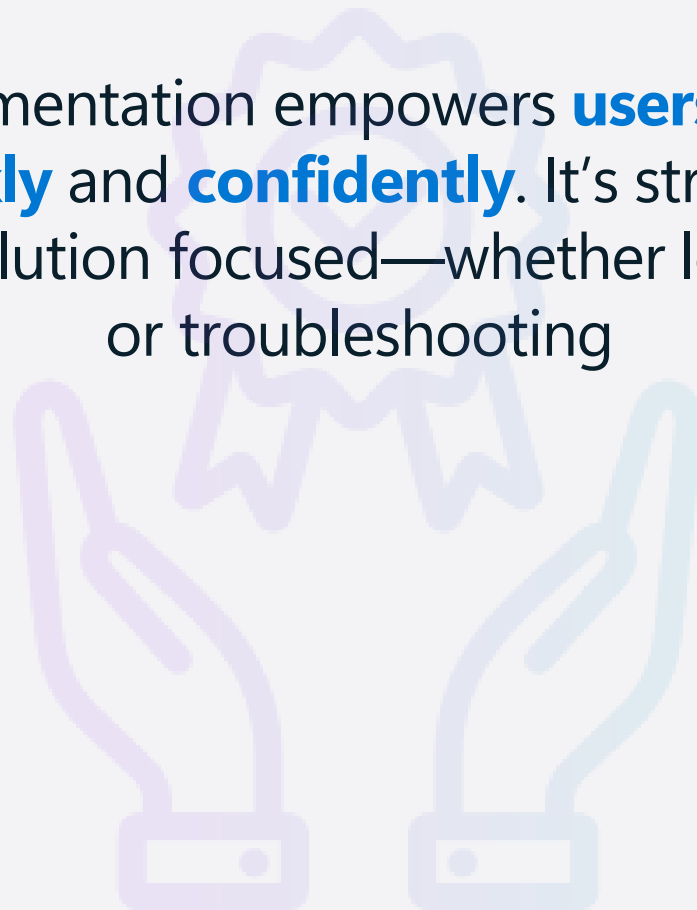


"I love the ability to switch the codes! This super helpful in Google. You don't have to translate or waste time. It just decreases translation time of languages which for me is a huge waste of time." – P5

What is Quality Documentation?

What is Quality Documentation?

Quality documentation empowers **users to solve real problems quickly** and **confidently**. It's structured, easy-to-navigate, and solution focused—whether learning, building, or troubleshooting



Key Recommendations & Next Steps

Key Recommendations

1. Surface Examples and Visual models earlier for faster contextualization

Refine document flow to foreground code snippets, use cases, and diverse visual models—such as workflows, flowcharts, and diagrams—at the top of each page or section. This accelerates user comprehension, anchors expectations, and reduces cognitive load by previewing actionable patterns before deep dives.

2. Enhance navigation and wayfinding by reducing page density

Streamline content presentation through collapsible sections, progressive disclosure, and modular layouts that prioritize scannability. This minimizes cognitive load while preserving depth—letting users skim efficiently, then expand into detail as needed, without compromising Microsoft's standard of thoroughness.

3. Add UX affordances that reflect competitive standards and user expectations

Integrate high-impact UX features—like 'Focus mode', prominent PDF download buttons, and intuitive CTAs for AI-powered actions—to reduce friction and align with user-preferred patterns seen across leading platforms.

4. Enhance semantic structure, readability, and perceived freshness

Ensure headings and subheadings follow a clear, hierarchical semantic structure to reduce information hunting and improve scannability. Introduce subtle cues—like "last updated" timestamps or freshness indicators—that signal content relevance without cluttering the interface or distracting from core tasks.

5. Introduce 'Ask Learn' through a clear, actionable summary entry point

Create a prominent 'Summarize page' pill in the top header that opens the 'Ask Learn' sidecar experience. This lightweight, persistent entry delivers immediate value, meets users in context, and introduces 'Ask Learn's' broader capabilities.

Next Steps

1. **Sync with product, design, and engineering teams to explore Examples and Visual model template updates**

Collaborate across teams to assess the feasibility of standardizing documentation templates that consistently surface examples and visuals—ensuring clarity and usability regardless of author.

2. **Coordinate with design and engineering on modular content components**

Explore reusable patterns—like collapsible sections, visual previews, and summary modules—that can be implemented consistently across documentation surfaces. Align on feasibility and effort.

3. **Operationalize insights through strategic coordination with product and platform team**

Learn PMs should coordinate with Product Groups and the Platform Team to triage and prioritize documentation improvements based on user-driven insights, team bandwidth, and available resources.

4. **Prototype and test enhanced documentation layouts**

Develop and test mockups that integrate examples, visual models, and an AI 'Summarize page' entry point. Validate with users to assess comprehension, navigation ease, and perceived value.

5. **Conduct foundational research on perceived document freshness**

Investigate how users interpret freshness and what signals they trust—informing design decisions around update indicators, timestamps, and content relevance cues.