APPENDIX A

When to hire an expert

As your documentation grows and scales, you might struggle to keep up with changes or need answers to documentation questions beyond the scope of your knowledge. A rapidly growing set of documentation requires you to manage many moving parts and demands multiple kinds of expertise. In these situations, it's better to hire a professional.

Documentation specialists, also called documentarians[1], can help you with advanced documentation issues. Documentarians go by various job titles, including technical writers, UX designers, project managers, and software engineers who focus on content. If you're looking to hire a documentarian, you can find them in the professional communities listed in the Resources appendix.

Regardless of their job title, documentarians can help you with critical inflection points for your documentation, such as the ones listed below.

[1] Eric Holscher, "Documentarians", Write the Docs. Retrieved June 22, 2021, `https://www.writethedocs.org/documentarians/`.

J. Bhatti et al., *Docs for Developers*, https://doi.org/10.1007/978-1-4842-7217-6

Meeting a new set of user needs

If you find yourself working with a new kind of user whom you don't fully understand, a documentarian can help you describe their use cases, define user journeys, and perform end to end testing of your documentation.

Increasing support deflections

If your support team is overwhelmed with solving support cases on a one-to-one basis, a documentarian can assess these issues and create docs that provide scalable support.

Managing large documentation releases

If the number or size of your launches make it difficult to keep your documentation up to date, or if you find that documentation consumes an increasing amount of your engineering and development time, a documentarian can help manage and write the documentation for large-scale software releases.

Refactoring an information architecture

If you find yourself trying to refactor an information architecture for large numbers of documents, a documentarian can help you plan and manage that process. Organizing documentation for searchability and scalability is difficult. A documentarian can guide you through planning a new information architecture and migrating content over.

Internationalization and localization

If you're struggling to localize your documentation for an international customer base, a documentarian can help you build and manage this content pipeline.

Versioning documentation with software

If you're creating a new version of your documentation with each software release and worried about scalability and SEO, a documentarian can help create a versioning process for your site.

Accepting user contributions to documentation

If you're considering accepting community feedback to your documentation and publishing articles or technical documents submitted by your users, a documentarian can provide a path for user-contributed content and respond to community feedback.

Open-sourcing documentation

If you're open-sourcing your documentation, a documentarian can assist with creating templates, standards, processes, and reviews for open source contributors.

APPENDIX B

Resources

This appendix offers a small selection of resources you'll find useful as you continue to work on documenting your projects. Resources are listed in no particular order.

We wrote this book as a field guide, a way to get your hands dirty with the work of documentation. We hope this book sets a path for your future adventures in technical writing.

We don't want the book to end here either, so don't think of this as the end. Consider this the starting point for future conversations. If you'd like to reach out to us directly, find us at docsfordevelopers.com.

Courses

- **Technical Writing Courses from Google**

  Google's technical writing team offers two self-guided courses in beginner and intermediate technical writing, focused on developers.

  Available at: `developers.google.com/tech-writing`

J. Bhatti et al., *Docs for Developers*, https://doi.org/10.1007/978-1-4842-7217-6

- **Documenting APIs: A Guide for Technical Writers and Engineers**

  Tom Johnson's API documentation course is an extensive set of self-guided tutorials full of practical tasks. Read Tom's blog for even more resources.

  Available at: `www.idratherbewriting.com/learnapidoc`

Templates

- **The Good Docs Project**

  The Good Docs Project is an open source set of processes, doc templates, and guides for creating great documentation.

  Available at: `www.thegooddocsproject.dev`

- **Diataxis Framework**

  The Diataxis Framework provides a guide to templating and structuring your documentation to meet different user needs.

  Available at: `www.diataxis.fr`

- **README checklist**

  There are many README checklists available, but Daniel Beck's is one of the best. It's a useful accompaniment to Daniel's talk "Write the readable README" which is available on YouTube.

  Available at: `www.github.com/ddbeck/readme-checklist`

Style guides

- **Google Developers Style Guide**

  This guide is widely used as a default for writing about API components and interactions, especially in open source projects.

  Available at: `developers.google.com/style`

- **Microsoft Style Guide**

  Microsoft's guide historically served as a common standard for interacting with UI components.

  Available at: `docs.microsoft.com/style-guide`

- **Mediawiki Style Guide**

  Mediawiki maintains a comprehensive style guide with example documentation templates for a wide variety of documents.

  Available at: `mediawiki.org/wiki/Documentation`

Automation tools

- **API reference generation**

  OpenAPI, Redoc, and Swagger are flavors of one of the most common API specifications for integrating documentation directly into an API.

  Available at:

  - `openapis.org`
  - `Redoc.ly`
  - `swagger.io`

- **Vale linter**

  Vale is one of the most common prose linters and allows you to write your own style rules and use codified style guides from Google, Microsoft, and others.

  Available at: `github.com/errata-ai/vale`

- **htmltest**

  htmltest lets you detect broken links in generated HTML.

  Available at: `github.com/wjdp/htmltest`

- **Read the Docs**

  Read the Docs is a site that automates building, versioning, and hosting documentation.

  Available at: `readthedocs.org`

- **Docsy**

  Docsy is a Hugo theme for technical documentation. Hugo (`gohugo.io`) is a Golang-based static site generator.

  Available at: `docsy.dev`

- **Netlify**

  Netlify is a content delivery network (CDN) with well-integrated continuous integration and delivery (CI/CD). It's a powerful and easy way to automatically publish content to the Web from a Git repository.

  Available at: `netlify.com`

- **Prow**

  Prow is a heavyweight CI/CD tool based on Kubernetes. Its features are powerful and almost certainly overkill for all but the largest projects—but it's invaluable for wrangling toil at increasingly massive scale.

  Available at: `github.com/kubernetes/test-infra/tree/master/prow`

Visual content tools and frameworks

- **Excalidraw**

  An open source whiteboarding tool to sketch diagrams.

  Available at: `excalidraw.com`

- **Snagit**

  One of the most widely used tools for screenshots and animated screen GIFs.

  Available at: `snagit.com`

- **C4 Model**

  A standardized, developer friendly approach to software architecture diagramming.

  Available at: `c4model.com`

Blogs and research

- **Tom Johnson, I'd Rather Be Writing**

  Comprehensive blog about technical writing, especially API documentation and the business value of technical writing.

  Available at: `idratherbewriting.com`

- **Bob Watson, Docs by Design**

  Great for academic articles about technical writing and measuring the quality of documentation.

  Available at: `docsbydesign.com`

- **Sarah Maddox, Ffeathers**

  Practical technical writing advice from a seasoned professional. Sarah also gives classes on technical writing and API documentation.

  Available at: `ffeathers.wordpress.com`

- **Daniel Beck**

  Practical technical writing advice from a freelance technical writer for GitHub, ARM, Mozilla, and others.

  Available at: `ddbeck.com/writing`

- **Stephanie Morillo**

  Advice on creating content with a focus on developer marketing, technical writing, and content strategy.

  Available at: `stephaniemorillo.co/blog`

- **Nielsen Norman Group**

  Well-researched insight into user experience (UX) data and best practices.

  Available at: `nngroup.com/articles`

Books

- ***Docs Like Code*, Anne Gentle**

  One of the most widely adopted models for developer documentation in current professional practice.

- ***Every Page is Page One*, Mark Baker**

  A guide to topic-based writing and helping users orient themselves in your documentation no matter where they land.

- ***How to Make Sense of Any Mess*, Abby Covert**

  A broad overview of information architecture, including a seven-step process for approaching information architecture challenges.

- ***The Content Design Book*, Sarah Richards**

  A tour through content design and meeting the needs of your users, using data to determine when, where, and how users want to digest information.

- ***User Research: A Practical Guide to Designing Better Products and Services*, Stephanie Marsh**

  A practical guide to user research methods, including face-to-face user testing, card sorting, surveys, A/B testing, and more.

- ***The Elements of Style,* William Strunk, Jr. & E.B. White**

  A classic and timeless guide to effective prose in English.

Communities

- **Write the Docs**

  Write the Docs is a global community of people who care about documentation, including programmers, tech writers, developer advocates, customer support, marketers, and anyone else who wants people to have great experiences with software. Write the Docs maintains an active online and in-person community through its Slack network, conferences, and local meetups.

  Available at: `www.writethedocs.org`

- **Society for Technical Communication**

  The Society for Technical Communication (STC) is a professional association dedicated to the advancement of technical communication. The STC supports a growing community of technical communicators through its publications, certifications, and conferences.

  Available at: www.stc.org

Bibliography

Abdelhafith, Omar, “README.md: History and components,” Medium, published August 15, 2015, https://medium.com/@NSomar/readme-md-history-and-components-a365aff07f10.

Baker, Mark, “Findability is a Content Problem, not a Search Problem,” published May 2013, https://everypageispageone.com/2013/05/28/findability-is-a-content-problem-not-a-search-problem/.

Beck, Daniel, “README checklist,” GitHub, https://github.com/ddbeck/readme-checklist/blob/main/checklist.md.

Beyer, Betsy, Chris Jones, Jennifer Petoff, and Niall Richard Murphy, *Site Reliability Engineering: How Google Runs Production Systems 1st. ed.*, (O'Reilly, 2016).

Calhoun, Ragowsky and Tallal, “Matching learning style to instructional method: Effects on comprehension,” *Journal of Educational Psychology,* Vol. 107 (2015).

Camerer, Colin, George Loewenstein, Martin Weber, “The Curse of Knowledge in Economic Settings: An Experimental Analysis,” *Journal of Political Economy,* Vol. 97 no. 5.

Casali, Erin ‘Folleto’ “Pixar’s plussing technique of giving feedback,” Intense Minimalism, https://intenseminimalism.com/2015/pixars-plussing-technique-of-giving-feedback/.

Chromium Projects, “Triage Best Practices,” accessed May 14, 2021, www.chromium.org/for-testers/bug-reporting-guidelines/triage-best-practices.

J. Bhatti et al., *Docs for Developers*, https://doi.org/10.1007/978-1-4842-7217-6

DAXX, “How Many Software Developers Are in the US and the World?”, published February 9, 2020, www.daxx.com/blog/development-trends/number-software-developers-world.

Fessenden, Therese “First Impressions Matter: How Designers Can Support Humans’ Automatic Cognitive Processing,” Nielsen Norman Group, accessed June 27, 2021, www.nngroup.com/articles/first-impressions-human-automaticity/.

Gaffney Gerry and Caroline Jarrett, *Forms that work: Designing web forms for usability* (Oxford: Morgan Kaufmann, 2008), 11–29.

GitHub, “About Code Owners,” accessed December 29, 2020, https://docs.github.com/en/free-pro-team@latest/github/creating-cloning-and-archiving-repositories/about-code-owners.

GitHub, “Open Source Survey,” accessed June 2021, https://opensourcesurvey.org/2017/.

Google, “Creating Great Sample Code,” Google Technical Writing One, https://developers.google.com/tech-writing/two/sample-code.

Jin, Brenda, Saurabh Sahni, Amir Shevat, *Designing Web APIs: Building APIs That Developers Love* (O'Reilly Media, 2018).

Johnson, Tom “Code Samples,” I’d Rather Be Writing, accessed June 29, 2021, https://idratherbewriting.com/learnapidoc/docapis_codesamples_bestpractices.html.

Keeton, B.J., “How to comment your code like a pro,” Elegant Themes, accessed June 29, 2021, www.elegantthemes.com/blog/wordpress/how-to-comment-your-code-like-a-pro-best-practices-and-good-habits.

Kubernetes documentation, “Documenting a feature for a release,” last modified February 11, 2021, https://kubernetes.io/docs/contribute/new-content/new-features/.

Kubernetes, "Issue Triage Guidelines," 2021, accessed June 27, 2021, www.kubernetes.dev/docs/guide/issue-triage/.

LePage, Pete, "Widgets," Google Web Fundamentals, accessed January 28, 2021, https://developers.google.com/web/resources/widgets.

Levie, W. Howard, Richard Lentz, "Effects of text illustrations: A review of research," *Educational Technology Research and Development,* 30, 195–232 (1982).

Lieby, Violet, "Worldwide Professional Developer Population of 24 Million Projected to Grow amid Shifting Geographical Concentrations," Evans Data Corporation, accessed June 29, 2021, https://evansdata.com/press/viewRelease.php?pressID=278.

Macnamara, Riona et al. "Do Docs Better: Integrating Documentation into the Engineering Workflow" in *Seeking SRE*, ed. David Blank-Edleman (O'Reilly Press, 2018).

Medina, John, *Brain rules: 12 principles for surviving and thriving at work, home and schoo*l (Seattle:Pear Press, 2008).

McCloud, Scott, *Understanding Comics: The Invisible Art* (New York: William Morrow Paperbacks, 1994).

Nasehi, Seyed Mehdi "What makes a good code sample? A study of programming Q&A in Stack Overflow," *2013 IEEE International Conference on Software Maintenance*, 2012.

Nazr, Shariq "Say goodbye to manual documentation with these 6 tools," Medium, accessed June 21, 2021, https://medium.com/@shariq.nazr/say-goodbye-to-manual-documentation-with-these-6-tools-9e3e2b8e62fa.

Nielsen Jakob, "F-shaped pattern for reading web content (original study)," Nielsen Norman Group, accessed June 29, 2021, www.nngroup.com/articles/f-shaped-pattern-reading-web-content-discovered/.

Nielsen, Jakob, “Keep online surveys short,” Nielsen Norman Group, accessed June 29, 2021, www.nngroup.com/articles/keep-online-surveys-short/.

Nielsen, Jakob “Photos as Web Content,” Nielsen Norman Group, accessed June 29, 2021, www.nngroup.com/articles/photos-as-web-content/.

Nielsen, Jakob, “Why you only need to test with 5 users,” Nielsen Norman Group, accessed June 26, 2021, www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/.

Newton, Elizabeth Louise Ph.D., “*The Rocky Road From Actions to Intentions*,” Stanford University, 1990, Ofcom, Connected Nations Report (2017), accessed June 2021, www.ofcom.org.uk/research-and-data/multi-sector-research/infrastructure-research/connected-nations-2017/.

Pelli, Denis G., Bart Farell, Deborah C. Moore, “The remarkable inefficiency of word recognition,” *Nature* (June: 2003), 423, 752–756.

Pittman, Paris, “How Kubernetes contributors are building a better communication process,” Kubernetes Blog, published April 21, 2020, https://kubernetes.io/blog/2020/04/21/contributor-communication.

Podmajersky, Torrey, *Strategic writing for UX: Drive Engagement, Conversion, and Retention with Every Word* (O'Reilly, 2019).

Potter M.C, Wyble B., Hagmann C.E, McCourt E.S, “Detecting meaning in RSVP at 13 ms per picture,” *Attention, Perception and Psychophysics* (December 2013).

Reyes, Jarod, “How Twilio writes documentation,” Signal 2016, YouTube, www.youtube.com/watch?v=hTMuAPaKMI4.

Roberts, David, “The power of images in teaching dyslexic students,” Loughborough University, accessed June 26, 2021, https://blog.lboro.ac.uk/sbe/2017/06/30/teaching-dyslexic-students/.

Rosala, Maria, “Ethical maturity in user research,” Nielsen Norman Group, www.nngroup.com/articles/user-research-ethics/.

Steele, Julie, *The art of data visualisation*, PBS film, 2013, accessed June 29, 2021, www.pbs.org/video/book-art-data-visualization.

Strunk, William, *The Elements of Style*. 4th ed. (Pearson, 1999).

Tufte, Edward R, *The art of data visualisation*, PBS film, 2013, accessed June 29, 2021, www.pbs.org/video/book-art-data-visualization.

Tufte, Edward R. *The visual display of quantitative information* (2001, 2nd ed.). Web Content Accessibility Guidelines, accessed June 2021, www.w3.org/WAI/.

Watson, Bob, “Measuring your technical content – Part 1” Docs by Design, published August 24, 2017, https://docsbydesign.com/2017/08/24/measuring-your-technical-content-part-1/.

Whicher, Charlie, “What we learnt from building a User Council,” Repositive.io, published November 13, 2017, https://repositive.io/resources/what-we-learnt-from-building-a-user-council.

Winters, Titus, Tom Manshreck, Hyrum Wright, “Documentation” in *Software Engineering at Google: Lessons Learned from Programming over Time*, (O'Reilly, 2020).

Zong, Jie, and Jeanne Batalova, “The Limited English Proficient Population in the United States in 2013,” Migration Policy Institute, published July 8, 2015, www.migrationpolicy.org/article/limited-english-proficient-population-united-states-2013.

Index

A

Accessibility, 106, 152, 153
Accuracy, 156, 157, 159
Analytics strategy, 160
- documentation metrics, 164
- organizational goals, 161
- questions of
  - quality, 165, 166
- readers' goals, 163

API documentation, 35, 36, 39, 95
Architects, 189
Audio file, 48–51, 55
Automating documentation, 177
- freshness checks, 178, 179
- link checkers, 179
- linters, 179, 180
- reference, 180

B

Business problem, 25, 33

C

Callouts, 46, 52, 56, 57
Changelogs, 27, 28, 39
Clarity, 60, 73, 157, 158
Code comments, 26, 87
Code samples, 84, 85
- autogenerating, 98
- design
  - choose language, 94, 95
  - complexity, 95
  - presentation, 96
- principles, 87
  - clear, 92, 93
  - concise, 90, 91
  - explanations, 87, 88, 90
  - trust, 94
  - usable, 93, 94
- sandbox code, 97
- testing, 97
- tooling, 96, 97
- types, 85

Cognitive processing, 102
Completeness, 71, 72, 156
Conceptual documentation, 30, 31, 35, 115
Content gap, 199, 205
Content release process, 123, 124
Content types, 24, 25
- code comments, 26
- READMEs, 27, 28

Corg.ly
- collecting new data, 10, 11
- developer surveys, 13, 14

J. Bhatti et al., *Docs for Developers*, https://doi.org/10.1007/978-1-4842-7217-6

Corg.ly (*cont.*)
direct interviews, 11–13
existing data sources, 9
friction log, 19, 20
identify user, 6
research findings, 14
support tickets, 9, 10
user goals, 4, 5
user journey map, 17, 18
user needs, 7
user stories, 16
user understanding, 8
Customer satisfaction score (CSAT), 140
Customer surveys, 135, 139

D

Deprecating content, delete, 180–183
Deprecation, 181, 182, 184
Developers, 6, 24, 129
Diagrams, 108
box and arrow, 109, 110
flowcharts, 110, 111
swinlane, 112
Direct interviews, 11–13
Documentation, 24
concept, 30
content outline, 42
planning, 41
procedural, 31, 32
reference, 35
starting, 29
Documentation metrics, 164, 166
baseline, 167
clusters, 166
context, 167
plan, 166
Documentation quality
definition, 151
functional, 151
accessibility, 152, 153
accuracy, 156
completeness, 156
findability, 155, 156
purposeful, 154
fundamental categories, 151
Document consistency, 159
Document sentiment, 138, 139
Drawing diagram, 113
complicated flowchart, 115
help, 117
place, 117
publish, 117
simplified flowchart, 114
starting point, 116
use colors, 117
use labels, 116

E

Editing, 68
approaches, 69
clarity and brevity, 73, 74
completeness, 71, 72
structure, 72, 73

technical accuracy, 70, 71
create, 75
peer review, 76, 77
review document, 75, 76
technical review, 77, 78
good feedback, 79, 80
process flow, 75
receive feedback, 78
Executable code, 85
Explanatory code, 86
External developer, 188

F

Feedback channels, 135, 136
customer surveys, 139
documentation page, 137, 138
document sentiment, 138, 139
support team, 138
user council, 141
Feedback, 141
triage, 142, 143
actionable issue, 143, 144
priorities for issues, 144
valid issue, 143
users, 145, 146
Findability, 155
Friction log, 4, 19, 20
Functional quality, 160

G

Gantt chart, 124, 125

H

Headers, 52, 53
Hierarchical structure, 190, 191
How-to guides, 33–35

I

Ineffective visual content, 104, 106
Information architecture, 187, 193
Irrelevant code, 91

J, K

Junior developers, 6, 15

L

Landing pages, 194–196
Links, 27, 34, 35, 37, 42, 179
Linters, 179, 180
Lists, 55
Low-accuracy documentation, 156

M, N

Maintaining, documentation, 173, 174
assign owners, 176
release process, 175
reward, 177

Markdown-based issue template, 137
Metrics, 135, 150, 162

N

Navigation cues, 178, 179

O

Organizational goals, 161, 162
Organizing documentation, 187, 198
  existing content, 198, 199
  hire expert, 203–205
  maintain information, 202, 203
  new information, 199–201
  validation checklist, 201, 202

P

Page-level elements, 189
Page-level feedback mechanisms, 138
Paragraphs, 46, 52–54
Peer reviews, 76, 77, 79
Pixar's method, 79
Plussing, 79
Procedural documentation, 31, 32
  how-to guides, 33–35
  tutorials, 32
Publishing content, 122, 123
Publishing timeline, 124, 125
  announcement, 129
  code releases, 126
  deliver content, 128, 129
  final approver, 126, 127

Q

Qualitative feedback, 166, 169
Quantitative feedback, 166
Quantitative metrics, 167

R

README, 27, 28, 44
Reference documentation, 35
  API, 35, 36
  changelogs, 39
  glossary, 37
  troubleshooting, 37–39
Release notes, 40, 65, 126
Release processes, 174–176

S

Sandboxes, 97
Screen readers, 108, 152, 153
Screenshots, 107, 108
Search engine optimization (SEO), 53
Sequential structures, 190
Site navigation, 190
  hierarchical structures, 190, 191
  sample architecture, 193
  sequential structures, 190
  users step, 192
  webs, 191, 192
Site reliability engineer (SRE), 6
Skimming titles, 57, 58

Structural quality, 157
- clear, 157, 158
- concise, 158
- consistent, 159

Support tickets, 9, 10
Swimlane diagrams, 112

T

Technical documentation, 57, 126
Technical reviews, 77, 78
Templates, 63, 64
Time to Hello World (TTHW), 154
Troubleshooting documentation, 37–39
Twilio's documentation team, 84

U

Use case, 24, 25, 30, 31
User interface (UI), 49, 50
User journey map, 14, 17, 18
User story, 16

V

Validate code, 68
Video content, 118, 119
- maintain, 120
- review, 119

Visual content, 103, 104
- accessibility, 106
- comprehension, 105
- performance, 106

W, X, Y, Z

Web Content Accessibility Guidelines (WCAG), 118
Writing, 46
- blank page, 47, 48
- choosing tools, 47
- draft, 52
  - callouts, 56, 57
  - headers, 52, 53
  - lists, 55
  - paragraphs, 53
  - procedure, 54, 55
- first draft, 65, 66
- goals, 49
- outline, 49, 50
  - complete, 51, 52
  - readers expectations, 50
- skimming, 57, 58
  - breaking large texts, 58, 59
  - breaking long documents, 59, 60
  - critical information, 58
  - simplicity/clarity, 60
- titles, 48
- unstuck, 61
  - ask for help, 61
  - change medium, 63
  - highlight missing content, 62
  - perfect, 61
  - sequence, 62
  - templates, 63, 64