

Canonical Technical Author — Written Interview

1. Documentation Experience and Insight

1.1. Outline your experience working on software documentation (type, audience, complexity, industry, impact).

Let's start from the beginning of my TW work.

First TW Steps (Career)

I came from a SMM/Content Lead background and started in technical writing in 2017 at the social casino gaming company "Murka". At that time their main product was mobile slot games. My main job was to write Technical Specifications and control the full development cycle of different slots, reskins, live-ops events, game mechanics, etc. I quickly improved my skills, and in a year was able to maintain 3 different projects simultaneously and mentor one more technical writer.

Also, I mentored new Technical Writers Team Lead (she came from Game Designers and had no TW experience), developed internal style guides, and wrote different writing templates for different features. So my main audience was the development team: product owners, game designers, artists, animators, developers (mostly Unity), QAs. I worked in Jira and Confluence, used internal style guide and Google Developer Documentation Style Guide.

My typical workflow looked like this:

1. Game Designer (GD) of Product Owner (PO) comes to me with a plan for a new slot, reskin, live-ops event, or new game mechanics.
2. We schedule a 1-on-1 meeting to explain a feature, to draw and write down some drafts.
3. I write the complete technical specification (some kind of HLD docs, but we didn't know that name yet:)) for the development team:
 1. References, examples, and guidelines of the graphic style for the artists.
 2. A concise description of the feature/functionality.
 3. A detailed description of the functionality (backend, frontend) of the entire feature, including dependencies on other product functionalities: how the feature should work, all dependencies, admin settings, edge cases, etc.
 4. References, examples, and guidelines of the graphic style for the animators.
 5. All transitions of the feature.
 6. QA checklist.
 7. Pay Tables (if this is a slot game) with the UX texts.
 8. Release Notes.
4. GD/PO reviews my document and leaves comments, if needed.
5. I improve documentation based on the comments, ask GD, PO, Devs, QA, or some other people if needed.
6. GD/PO reviews once more, approves the document, adds some hard technical stuff such as feature balance.
7. Final stage: I oversee the entire development cycle and answer questions that arise among developers during the process. During testing, GD/PO responsible for the feature also joins the product.

I have some cool achievement from this job, but I'll share it in the next answers.

Moving to Kyiv

The company was acquired by the Blackstone Inc., and I decided to move forward. I moved from Dnipro to Kyiv (the capital of Ukraine) and found the TW job in small startup (~14 people) inside the big company. Our product was the municipal Resource Management system that routed citizen requests (e.g. utility outages, repairs) to the appropriate city services for rapid resolution. It was complex web admin panel, and mobile applications (iOS, Android) with different roles (admins, operators, executor, dispatch, client, etc.).

There I started learning Markdown/VScode and Docs-as-a-Code approach. I deployed Netlify docs site with the versioning, wrote User Manuals for all kind of roles, Release Notes, and deployed everything via Git. Also, I actively participated in product development: together with the BA and PO, we decided how best to implement a particular feature, and I helped the UX-designer write UX microcopy.

Covid happened, and the startup was shut down shortly after. Then I worked 2 months on the Parking App, but it was shut down too.

Next Steps

Then I became fully remote and started the career at the biggest Ukrainian VoD Service MEGOGO.NET in the Release Department. We have over 200 people in department and nearly 900 overall. It's very complex VoD SaaS platform with a microservices architecture. I write user manuals for the platform operators based on HLD docs, architecture diagrams, API (based on REST) methods, vendor and audit docs and presentations, different complex BPMN flow diagrams, high- and low-level services diagrams, etc. Also, I mentor one TW continuously for over 3 years, improving both our skills. I work in Jira and Confluence here.

I quickly realized that I could sustain good work-life balance, and decided to take one more career. So in 2021 I became a part of European iGaming company SkyWind and started writing in English as well as Ukrainian. There was no documentation for the complex back office platform, also microservices-based. I raised Docusaurus portal with the full Documentation site, including integrations, API, user manuals, release notes, all kinds of diagrams, etc. My documentation guided engineers through platform deployment and services configuration, payments services and integrations, betting and casino control panels, analytics dashboards, etc.

In a couple of years the Playtika's recruiter reached me in LinkedIn, and I moved there. It's a leading developer of social casino games, powered by a microservices-based platform. Their platform already had some docusaurus portal, but the docs were outdated. The product was also complex back office microservices platform with over 30 products (over 200 microservices). I worked on the platform docs and fully maintained versioned Docusaurus portal (so for the complex react components we worked together with the developer to integrate) via CI/CD pipelines in Git. I described integration docs, APIs, release notes for the platform, all the product manuals and dependencies, and released new documentation versions via Github Actions. I also tried to mentor another TW, but we decided to say goodbye to him, it was very unproductive and not learning guy. Unfortunately, during my time at the company, there were several restructuring phases — and I was impacted by the most recent round of layoffs.

I also have some cool achievements from Playtika to share later on.

Unfortunately, almost every piece of my work is under NDA, so I can't share direct portfolio.

1.2. What is the most significant body of documentation that you have had responsibility for?

There are several great examples in my work:

1. Complex Web Store manual: a store such as Google Store / AppStore but for the in-game purchases for the different studio games. This store included different buying options, loyalty program, sales section, etc. The PO reached out to me during development stage me with the HLD drafts, API and integration docs from the developers (written in "their" complex-to-understand language), and Figma mockups. In 1 month I created and deployed a fully comprehensive manual for the product, in audience-friendly language (studio managers and operators) based on Google Developer Documentation Style Guide and internal guidelines. It included integration docs, HLD diagram, API methods, and user manuals.
2. Recently I have asked PO/PMs of 12 development teams of their CI/CD processes, and DevOps for the architecture nuances. The result was the complex Confluence document with every process documented and explained for the company newcomers. The Head of Release Department reviewed it and approved without a hesitation.
3. In Skywind, I built the documentation portal from scratch, implemented a docs-as-code workflow (Docusaurus + Markdown + Git), introduced structure based on the Diátaxis framework, and migrated Confluence materials into a modern, searchable system. This portal became the single source of truth for 150+ developers across several teams. It reduced repeated questions in Slack, improved onboarding for new engineers, and enabled cross-team collaboration. I was solely responsible for architecture, tooling, versioning, content structure, and ongoing maintenance.

1.3. What is your proudest success or accomplishment as a technical author?

I think is was one my recent accomplishments, the manual for the internal AI tool. This tool helps the artists develop different assets for the games. It's like complex AI image generator trained on internal game assets. In one week based on the meeting with the PO and stage environment I developed user manual for this product, which was approved without comments, and it helps artists every day. It shortens the work on assets from 5-10 days to 1-2, and my manual helps to understand every product feature such as "stable diffusion", for not-so-technical audience.

Also the answer for the [1.6](#) fits here 😊

1.4. What open-source software documentation do you use the most? How would you suggest improving its content?

I use Docusaurus docs the most when trying new tools or adding components, and often check Stack Overflow when working with docs-as-code setups.

While Docusaurus docs are generally great, some advanced topics (like plugins or theming) would benefit from more real-life examples and a clearer structure. I'd also split tutorials, how-to guides, and reference pages more explicitly — sometimes they feel a bit mixed.

1.5. Consider one of the software documentation projects you have worked on. How well did its processes work, and what would you like to improve?

There was a time in one project when I realized that many important parts of documentation work — like setting up linting, broken link checks, advanced search plugins, and Git workflow automation — weren't

visible to stakeholders, even though they made a big difference in quality and scalability.

So I started thinking about how to improve DocOps contributions. I added internal changelogs (release notes), gave short demos, and posted regular chat and email updates to highlight non-content work. I also made sure to align with the team early on to explain what "documentation" includes — not just writing pages, but also improving the tools and systems behind them.

I was shifting careers at the time and applied these improvements in my next job — and they worked really well.

1.6. Describe a significant learning experience you had working on documentation — what happened, what did you learn, and what did you change as a result?

During my time in Murka, there was a great case, let me share it with you.

GD reached me with the new complex live-ops feature idea: we have a gaming hexagonal map (such as HoMM3 map) with different chests and armies (mobs) on the hexes. You start with your own army, and the goal is to gather all the chests and reach the end of the map (then the next harder map begins). You walk on the map using event currency, which you gain spinning main game slots. You walk the hex field, fight the armies, gather chests with rewards (which includes in-game currencies and other live-ops events rewards), and upgrade your army to fight better.

GD explained this feature to me during a 30-minutes 1-on-1 meeting and gave me two A4 sheets with hand-drawn sketches of the main user interface screens.

In 1 week I transformed this knowledge into full Technical Specifications for the dev team, from the artists to the devs and QAs. The Game Designer approved it without comments, and I supervised and coordinated the entire feature development without further involvement from his side. So basically there was only an idea from the GD, and full my realization (except feature balance).

This experience taught me active listening, real-time note-taking, structuring raw ideas into clear documentation, and identifying edge cases early in the process.

As a result, I became more proactive during early feature discussions, started using quick wireframes in my docs, and added regular checkpoints with stakeholders when working on complex specs.

1.7. In general, what aspect of your documentation work do you find most challenging? Give an example or two to illustrate the nature of this challenge.

I found it most challenging to understand developers' specific "language", to decompose these the tool into smaller parts, and to explain it in audience-friendly language.

For example, at one point a developer gave me one or two brief paragraphs about a Kubernetes tool I had no experience with. He was too busy to help, so I researched it myself, broke it down into smaller parts, and wrote a clear architecture overview. Later, the same developer shared it with others as a reference.

2. Software Experience

2.1. Describe your programming skills and experience and what you have enjoyed.

I had a ZX-Spectrum clone back in childhood, and then I wrote simple apps/games like pong using basic and command line. In high school/college, I learned Pascal/Delphi and Visual Basic. I wrote my graduate diploma in Delphi: It was "Russian-Ukrainian dictionary-reference of transport terms". The entire program was made in Delphi, including the vocabulary with the bubble sort algorithm and additional materials: illustrated traffic rules (built-in PDF), some cartoons made in Flash, and additional materials. It was a challenging project at age 15, and I really enjoyed it.

After that I had no direct experience in writing and hard learning complex programming languages, but I know HTML/CSS/XML, learning React so it's main Docusaurus language, and can read and understand most modern languages, such as JS/JavaScript/Python/etc, cause it helps me with my documentation work.

2.2. Are there any areas of software technology (e.g. cloud services, Linux, infrastructure, deployment, robotics, databases) in which you have particular experience or interest?

I have experience with infrastructure (deployment of Docusaurus/Netlify portals and WordPress sites), deployment processes (CI/CD pipelines, GitHub Actions), and cloud-based services used in video streaming and iGaming platforms.

I'm also interested in AI-assisted tooling. At one company, I documented an internal AI image generation tool used by artists — it was based on Stable Diffusion and trained on proprietary assets. This sparked my interest in the broader application of AI in development and DevOps workflows.

2.3. What does open-source software mean to you personally?

It's the software with the free access to all the audiences and the free-to-use and modify code. It's very cool feature so anyone can deploy it, use it, and taylor to ones needs.

2.4. Describe any experience you have working in open-source contexts, including development of software, documentation, or community.

I've worked only in large commercial B2B projects, some NDA products, and startups, so I don't have direct open-source experience. But I work daily with this kind of software such as Docusaurus and Git and I'm interested in contributing there.

2.5. Consider one of those contexts: did you observe any practices that you would like to improve?

For example, as I told above, the Docusaurus manuals could be improved. Tutorials, references, and explanations are sometimes mixed, which can be confusing for new users. I'd improve the structure using the Diátaxis model and add more real-life examples, especially for advanced topics like plugin development and theming.

3. Leadership and Professional Engagement

3.1. Describe any speaking experience at conferences or other events, or public writing about documentation or software.

I have no such experience, only internal presentations for different teams.

3.2. What influence have you had on others in the industry, through your speaking, writing, or other work?

Only internal: I've mentored a Game Designer who transitioned into a Technical Writer role, and I've mentored other technical writers in three different companies. I also developed new documentation templates and introduced new tools — for example, switching from Confluence to Docusaurus in one company to improve documentation.

3.3. Describe your engagement or participation in any professional or industry communities focused on documentation or software.

I have no such experience, but I contributed a few updates and news to a Telegram channel for technical writers run by a friend.

4. Education

4.1. How did you rank amongst your peers in your final year of high school in mathematics and the sciences? What was your strongest subject?

Frankly, I don't remember much about my education, and neither I nor my father could find the score sheets from that time. I did average in everything, but was fond of mathematics and physics. The strongest subject was physics — I got 11 out of 12.

4.2. How did you rank in languages and the arts? What was your strongest subject?

I got 11 out of 12 in both Ukrainian and English, because I had studied English with a tutor since I was 4 years old, and I love my native language.

4.3. What were your results in school-leaving and/or university entrance exams (GPA, A-levels, SAT, etc)?

I don't remember the school-leaving results, but in the university entrance exams I got around 174 out of 200 in physics.

4.4. What sort of high school student were you? Beyond your studies, what were your interests and hobbies? How do you think you are remembered by your peers?

I was a good student. I liked optoelectronics, but I enjoyed practical work more than theoretical subjects. I lost my mother to cancer at the end of my second year at university, and it hit me hard. I took an academic break and later rejoined university on a contract basis, graduating after 4 years with average grades. However, during the entrance exams for the fifth year, I ranked first in a class of 15 students (we had a small group left to graduate).

4.5. What degree and university did you choose, and why?

I chose Applied Physics / Optoelectronics / Radioelectronics because I liked physics, many of my friends went there, and it was more of a "programming" degree and education.

4.6. At university, did you do particularly well in any area of your degree?

I was an average student in most areas except programming — but we had programming only for two years, and it covered basics of Basic and Pascal, same as in high school/college.

4.7. Overall, what was your degree result, and how did that reflect on your ability? Please explain the scoring system.

I did my bachelor's degree work in optics, it was engineering lab work: I studied the properties of crystals under different pressures and temperatures. I received 11 out of 12 for this work.

4.8. In high school and university, what did you achieve that was exceptional?

I think there were no truly exceptional achievements.

4.9. What leadership roles did you take on during your education?

Unfortunately, none.

5. Canonical in Context

5.1. Why do you want to be a technical author at Canonical?

I'm excited about Canonical's engineering culture and commitment to open-source innovation. I want to grow in documenting AI tools and platforms — and Canonical feels like the right place to learn, contribute, and scale that expertise globally.

5.2. Which Canonical products or technology domains are you most interested in working on?

I'm especially interested in Canonical's AI initiatives — like Charmed Kubeflow, the GenAI Infrastructure, and the Data Science Stack. These products combine strong infrastructure with real-world MLOps and LLM use cases, and I'd love to work on them. I work and explore AI-assisted tools daily and want to grow deeper into documenting AI platforms, pipelines, and deployment workflows.

5.3. What is your understanding of Canonical's position in the open-source software market? What strengths and weaknesses do you see?

Canonical leads in making open-source software production-ready. Its strength lies in combining community-driven development with strong enterprise support. I can see an opportunity to enhance the UX and unify documentation across products, especially as Canonical scales its AI efforts.
