The Dynamic Technical Documentation Landscape in the Software Industry and Impact on the Stakeholders: A Look Ahead

Bishnu Shankar Satapathy [0000-0003-0955-2680][1*], Siddhartha Sankar Satapathy[2], S. Ibotombi Singh[2], and Joya Chakraborty[3]

[1]Center for Multidisciplinary Research, [2]Department of Computer Science and Engineering, [3]Department of Mass Communication and Journalism, Tezpur University, Napaam-784028, Assam, India.

Abstract 1

A comprehensive set of software documentation facilitates end users and customers understand a software solution appropriately, ensuring the solution provider's market credibility. However, for the stakeholders like developers and technical writers, delivering complete technical documentation presents a considerable challenge. Now, stakeholders' responsibilities are not limited to planning and organizing content but also to aligning with emerging trends and technologies. In addition to embracing the latest technology, stakeholders prioritize accuracy, ensure universal accessibility of documents, refine their writing practices, and contemplate ethical considerations. This article investigates technical documentation professionals' challenges when adapting new tools, technologies, and delivery methods. It also examines how the technical writing space has changed with advancements in artificial intelligence (AI) and natural language processing (NLP) and how these technologies may help with tasks like proofreading, drafting content, personalizing information, and optimizing content in real-time. It emphasizes the need for stakeholders to alter while preserving high standards and being thoughtful of the possible impact of their work on the larger community.

Keywords:

Software Documentation; Doc-as-a-Code; Artificial Intelligence; NLP; Software-as-a-Service

1. Introduction

Following the footprint of rapid technical evolution, the technical communication landscape is also evolving to meet changing end user expectations and customers' demand. Version control and documentation management now heavily rely on collaboration tools like Git and automation tools such as Sphinx, MkDocs, and Jekyll. In addition, continuous integration and continuous development (CI/CD), agile processes, and frequent small-scale updates to documentation in line with software releases have become integral part of the development process (Figure 1). Agile development has incorporated feedback loops, resulting in documentation that is more user centric [22]. The formation of cross-functional teams has promoted collaboration and a culture of continuous learning, motivating technical writers to remain abreast of evolving technologies, tools, and best practices [2,3].

In this manuscript, we explored the current trends in technical documentation and writing practices within the software industry. Our primary focus is to understand the processes, challenges, and difficulties associated with technical documentation. The overarching goal is to identify the significant challenges faced by stakeholders in delivering high-quality documentation.

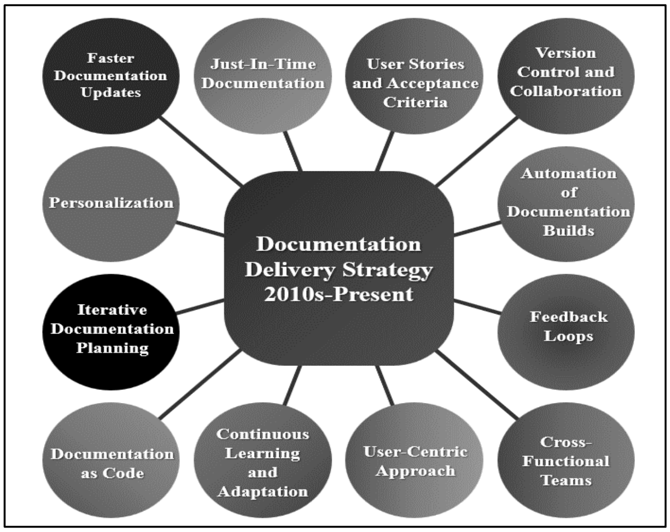
Faster Documentation Updates
Just-In-Time Documentation
User Stories and Acceptance Criteria
Version Control and Collaboration
Personalization
Automation of Documentation Builds
Documentation Delivery Strategy 2010s-Present
Iterative Documentation Planning
Feedback Loops
Documentation as Code
Continuous Learning and Adaptation
User-Centric Approach
Cross-Functional Teams

Figure 1. The figure depicts the significance of the technical documentation sectors' current practices.

The study addresses the following research questions:

- What challenges do software documentation stakeholders of providers encounter when creating and managing technical documentation that aligns with the SaaSification of software solutions?
- How does the study assess the integration of Doc-as-a-Code principle into technical writing and documentation as part of the software release management process?
- What challenges do stakeholders face when understanding and adapting to new technology through technical documentation practices?

The following sections of the paper are structured as follows: Section 2 provides details from the research methodology on the effects of industry revolution technology on the release management of the software industry, Section 3 outlines the findings from the questionnaire-based survey among technical documentation stakeholders and the associated gaps during the transition phase aligning with technological advancements. Thes section 4 and 5 summarizes the research observations and concludes the paper.

2. Research Materials and Methodology

2.1 Systematic Literature Review

A systematic literature review is an effective method for delving into new research areas and is essential for developing the fundamental concepts that form the basis of empirical studies. It consists of three key stages: planning, conducting, and reporting. During the planning phase, the groundwork is laid by identifying search terms, selecting databases, and defining timeframes. The next step involves searching, screening initial papers, and applying inclusion and exclusion criteria. Finally, the data is analyzed and synthesized, leading to the reporting stage, where the findings are presented based on the analysis of the selected papers.

2.1.1 Explore the keywords on industry revolutions and software release management

Considering keywords used, we have systematically reviewed available research literature to explore unfamiliar research issues competently. We aimed to gain insight into the latest technology adoption trends in the technical writing industry, focusing on software solution organization. This article explores the current trends that affect technical writing. It addresses the typical challenges faced by professionals in this field and looks ahead to the future.

2.1.2 Inclusion Criteria

We have selected papers from peer-reviewed journals and international conference proceedings to ensure the reliability of the findings. These papers have been sourced from reputable libraries to prevent any bias in the sample collection. They specifically discuss the impact of recent AI technology on decision-making in release management workflow. Our focus was on studies that offered new data through experiments or observations, as well as a qualitative study examining the technical documentation process and recent technologies. To ensure that our review remains relevant, we only considered papers published within the last two decades, from 2010 to 2024.

2.1.3 Exclusion Criteria

We excluded papers not having full-text and non-peer-reviewed papers to maintain standard. We also excluded papers not relevant to the software industry. For instance, papers that discussed AI in healthcare contexts were excluded. As our focus was on primary research papers, secondary literature reviews, tutorials were excluded to avoid redundancy. We excluded papers published in other than English languages.

We conducted a keyword assessment of available research articles, a significant step in our research process, to analyze the scope of issues addressed in research papers on software release and research article metadata, including titles, abstracts, and author keywords. We utilized relevant keywords on software documentation; Doc-as-a-Code; artificial intelligence technology; NLP; Software-as-a-Service and search for articles published since 2010, as this period marked the maturation of Industry 4.0 and the conception of the Industry Revolution [1], [25]. After screening 6059 downloaded articles based on inclusion and exclusion criteria, we identified 5048 unique articles (Supplementary Table 1) for frequency analysis of key terms using text mining techniques. Our study focused on keywords related to technical documentation in software development practices, and the frequency distribution is presented in Table 1.

Table 1: *Software Documentation and related keywords aligning with the technological transformations*

Software doc	Doc as a code	NLP	AI Open-source Tools
User Manual	DocOps	Natural Language Processing	Conversational AI
API Documentation	Documentation Automation	Text Analysis	Language Model
Technical Documentation	Markdown Documentation	Machine Learning	OpenAI
Release Notes	Versioned Documentation	Language Understanding	Text Generation
Installation Guide	Continuous Documentation	Sentiment Analysis	Dialogue System
Configuration Guide	Collaborative Documentation	Named Entity Recognition	Natural Language
Troubleshooting Guide	Documentation as Code	Text Classification	Understanding
Best Practices	DevOps Documentation	Text Generation	Human-Machine Interaction
Glossary	Automated Documentation	Language Modeling	AI Chatbot
Example Code	Git-based Documentation	Information Retrieval	
Use Case Examples	Document Automation		
FAQs			
Known Issues			
Change Log			
Compliance Documentation			
Data Dictionary			
User Feedback			
Security Documentation			
Backup and Recovery Guide			
License Information			

2.2 Questionnaires based survey among the stakeholders

In technical communication, organizations grapple with integrating innovative technologies into their systems. To explore this dynamic, we designed a comprehensive survey using Google Forms. Our target audience consisted of stakeholders from reputable product-based companies who play pivotal roles in strategic decision-making regarding technological adoption. By analyzing their responses, we aimed to uncover insights into effective technical writing strategies that align with the ever-evolving landscape of software products.

Throughout the planning phase, we meticulously curated our research sources. While Google Scholar remained a valuable repository, we broadened our scope to encompass other authoritative research libraries. Science Direct, Web of Science, and Scopus emerged as crucial platforms for accessing mainstream scientific journals. Our search focused on topics such as "Doc as Code", "Software Documentation", "AI Technology", "NLP" and "Software as a Service". By concentrating on reputable sites like Gartner and conference papers, we ensured the relevance and reliability of our findings. We also acknowledged the rapid shifts in technical documentation practices by extending our temporal lens to 2005 and beyond.

3. Results

3.1 Literature survey on documentation-as-a-code (DocOps) and automation

The importance of research in the software industry related to technical documentation can be observed from the result (**Figure** 2) as the key terms such as AI, & NLP related key words were seen with higher frequencies. The data analysis reveals that AI technology has the highest relevance across all sections (abstract, key word, and title), with 26.40%, 20.50%, and 9.40%, respectively. This indicates a significant focus on AI Technology in current discussions and research. NLP follows with scores of 12.00% in the Abstract, 9.10% in key word, and 4.40% in the title, highlighting its importance as well. In contrast, Software Documentation and Documentation as Code have much lower relevance scores, suggesting they are less emphasized in the current context.

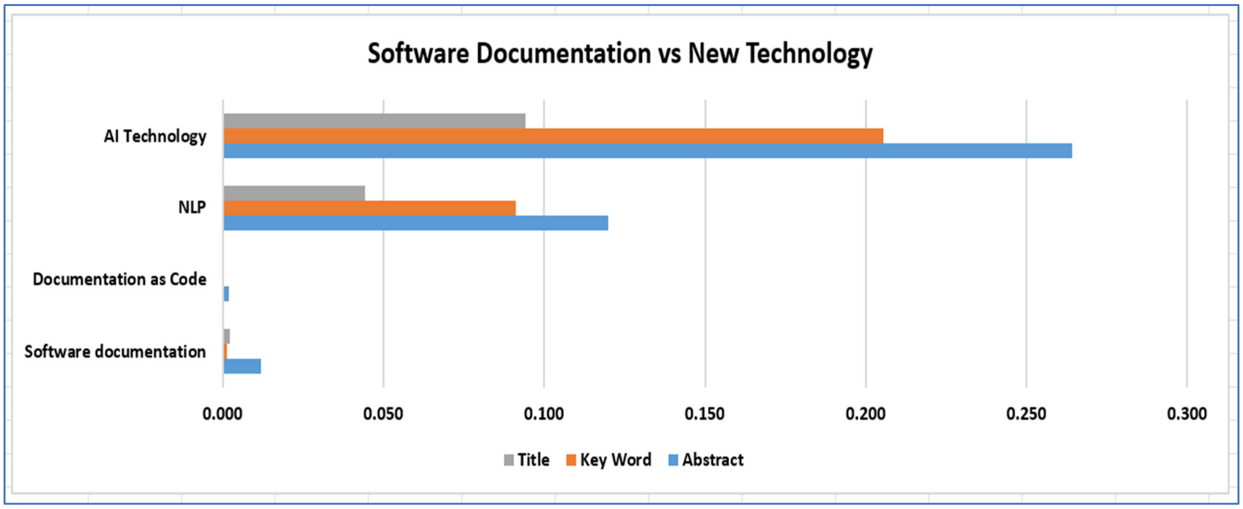

Figure 2. Figure depicts the frequency of the keywords in the research articles analysed in this study. The X axis represents in the figure stands for frequencies of keywords of various categories in the title, author's keywords, and abstract of the research articles published in different journals.

This analysis underscores the prioritization of AI Technology and NLP in technical documentation and research. The formula used to determine relevance involved extracting values from the provided data for each category (Abstract, Key Word, and Title) and converting them to percentages. This approach helps identify the most prominent topics, guiding future research and development efforts in these areas. Frequencies of the terms such as documentation as a code and software documentation were found to be among the lowest values. Observations from this study suggest that although research manuscripts related to the emerging trend of modern technologies aligning with Industry Revolution principles are available, however, on the other hand, the manuscript number about the technical documentation and related delivery strategy volume are on the lower side.

Researchers are also talking about using robots and computers to help write documents. It includes things like chatbots, understanding human language, and programs that learn. Another idea is to treat documents like they are part of the software they are about. It means they can be tracked, checked, and updated like software. In the future, experts think that AI and chatbots programs will be able to create documents that are just right for each end user [23]. Stakeholders also think that leveraging on the new technologies will make it easy to measure how good the software product documents are and use the inputs to improve them.

3.2 Empirical challenges in the field of technical documentation

3.2.1 Continuous Integration and Continuous Deployment (CI/CD)

The rise of continuous integration/continuous deployment (CI/CD) pipelines in software development has drastically shortened the development lifecycle [6]. As software products evolve almost in real-time, technical documentation teams face immense pressure to keep pace with these rapid changes. Updating the software documentation during every release is a tedious task, especially when releases happen multiple times a month. Also, more appropriate documents are needed to support end user satisfaction and reduce support costs.

3.2.2 Increasing complexity of software architectures

In recent times, software solutions are built using complex cloud architecture such as microservices, cross-platform functionality and hybrid cloud deployments. To support technical documentation for such architectures requires inherent technical understanding and often involves collaborating with teams spread across different geographic locations and time zones [20, 21].

3.2.3 AI-powered software and black-box systems

As AI-powered applications and machine learning models become more established, documenting systems with opaque decision-making processes mounts new challenges as explaining how AI models reach decisions, especially in high-stakes environments is very critical. Many technical writers need help to describe AI models' functionality without overwhelming end users with too much technical jargon.

3.2.4 Customers expectations for interactive documentation

The customers/end users are now looking for interactive, dynamic, and personalized content that adapts to their needs and roles instead of static text form of technical documentation [4]. This includes contextual help, video tutorials, embedded troubleshooting flows, and chatbots capable of answering specific queries in no time [7]. Matching these high expectations from customers often requires integrating multiple media types and creating complex decision trees to personalize content.

3.2.5 Security and compliance documentation

The rise of stringent global data privacy regulations (such as GDPR, CCPA, and other emerging policies) forced the technical documents to be accurate and up-to-date security and compliance [15]. Information on data handling procedures, encryption standards, and privacy policies must go through the legal validation apart from the functional validation by the engineering teams, which further complicates the process. Software solution providers may face severe legal and financial implications if there are any errors in the provided information [16].

3.2.6 The role of AI in transforming technical documentation

AI is emerging as a pivotal force in addressing these challenges, providing tools to help automate, streamline, and enhance the process of creating and maintaining technical documentation. Writing tools leveraged on AI technology, like natural language generation (NLG) systems [1], can create initial draft automatically using structured data [10], API details, or codebases [5], [9]. This helps technical writers save time by handling routine or template-based content, allowing them to focus on more detailed and complex parts of the documentation. Further integration of CI/CD pipelines can detect software changes automatically and update the relevant information in the software documentation in real time [17]. This keeps the documentation in sync with the latest software releases, ensuring its accuracy. Leveraging on the advanced AI search algorithms, the knowledge bases can also organize, and present content based on user activities, ensuring user see the most relevant information when they need it [8] [18, 19].

The chatbots and virtual assistants can now be embedded into software documentation sites, offering users interactive, conversational access to documentation. This means users can ask specific questions and get tailored answers, engaging with the content more effectively and avoiding unnecessary information overload. AI-driven translation tools are improving the accessibility of technical documentation across languages. Machine translation powered by AI can now accurately translate documentation into multiple languages and adjust for regional preferences, technical standards, and idiomatic expressions, creating a better experience for users in different markets [11, 12].

3 2.7 Future of Technical Documentation with AI

In future, leveraging on AI, both text and multimedia documentation can be created automatically which will include video tutorials, interactive diagrams, and voice-guided walkthroughs. This approach could make technical content more engaging and meet the demand for interactive learning resources without requiring manual updates. AI will allow for personalized documentation based on user profiles, roles, or past interactions. By 2030, AI could be involved throughout the entire documentation process, from initial content creation to updates. This would automate much of the repetitive, error-prone work, allowing writers to focus on reviewing AI-generated content for accuracy and addressing complex topics that require human expertise.

3.2 Questionnaires based survey among the stakeholders

To understand the current trend and challenges in the software documentation industry and the real-world challenges faced by stakeholders due to rapid technological advancements, we conducted survey considering stakeholders involved in various phases of the software documentation life cycle. The survey was conducted over five months, from January 2024 to May 2024. Skilled software documentation stakeholders from various organization were interviewed for this study, gathering pertinent information covering simple to complex multi-modal solutions. The project type includes traditional documentation delivery model, also the organization who follows automation and latest AI technologies for software documentation delivery, support, and maintenance. In addition, we have gathered data points on the challenges the organization faces. Keeping the confidentiality of the individuals and organizations in view, we present a summary of the answers received in the later sections focusing on the software industry's process in the direction of SaaSification.

The survey results underscore a significant shift in the industry, with traditional delivery methods, while still prevalent and highly effective, gradually making way for next-gen approaches. This transition is particularly evident in the software solution providers related to the organization adopting more modern methods. The structured authoring process stand out with the highest delivery approach score and a strong focus on future improvements, demonstrating a clear commitment to modernization despite the uncertainty in their roadmap.

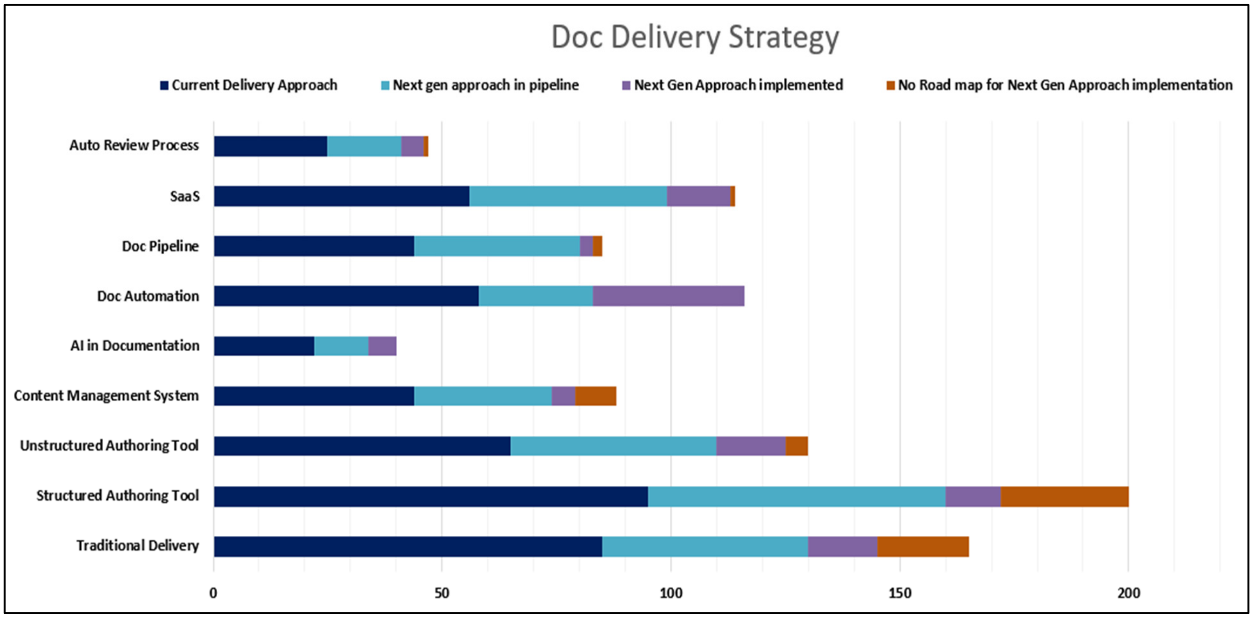

Figure 3: Technical Documentation Stakeholder's Key Concern on Doc Delivery Strategy. The length of the horizontal bars represents the proportion of responses aligning with the legends specified in the chart with respect to the survey questionnaire.

Analysing various documentation and content management delivery approaches reveals distinct trends and future potentials (Figure 3). Traditional delivery methods remain prevalent, accounting for approximately 17.21% of the delivery approach. Despite their high usage, the next-gen approach in the pipeline is moderate at 14.20%, indicating some movement towards modernization. The average team size is large at 20.00%, with a balanced mix of defined (13.89%) and uncertain (30.30%) roadmaps.

Structured authoring tools, with their high effectiveness and significant focus on future improvements, are a clear indicator of the industry's commitment to modernization. Their delivery approach score of 19.23% and 20.50% in the next-gen approach pipeline, despite an average team size of 17.50% and a higher uncertainty in the roadmap, show a strong dedication to staying at the forefront of the field.

Emerging technologies such as AI in documentation and auto-review processes have lower delivery approach scores of 4.45% and 5.06%, respectively. These technologies are still nascent, reflected by their smaller team sizes (5.00% and 7.50%) and minimal uncertainty in their roadmaps (0.00% and 1.52%). Document automation, with a delivery approach score of 11.74%, shows a balanced approach with a strongly defined roadmap (30.56%) and no uncertainty, indicating a clear path forward.

Overall, each approach has its strengths and areas for improvement, with varying levels of team size, defined roadmaps, and uncertainties. This comprehensive view provides valuable insights into the current state and future potential of different delivery approaches in the field of documentation and content management.

Analyzing various documentation and content management delivery approaches reveals distinct trends and future potentials (Figure 4). Traditional delivery methods remain prevalent, accounting for approximately 30.30% of the uncertain road map category. Despite their high usage, the development task is a priority for 28.57% of the cases, with budget constraints affecting 20.00% and unknown reasons accounting for 19.05%.

Structured authoring tools, despite exhibiting the highest uncertainty in their road map at 42.42%, show a significant commitment to development tasks, with 45.24% prioritizing these tasks. Budget constraints concern 30.00%, while unknown reasons account for 9.52%. This high level of focus on development tasks and the relatively low percentage of unknown reasons suggests a strong commitment to modernization, instilling optimism about their potential despite the uncertainty.

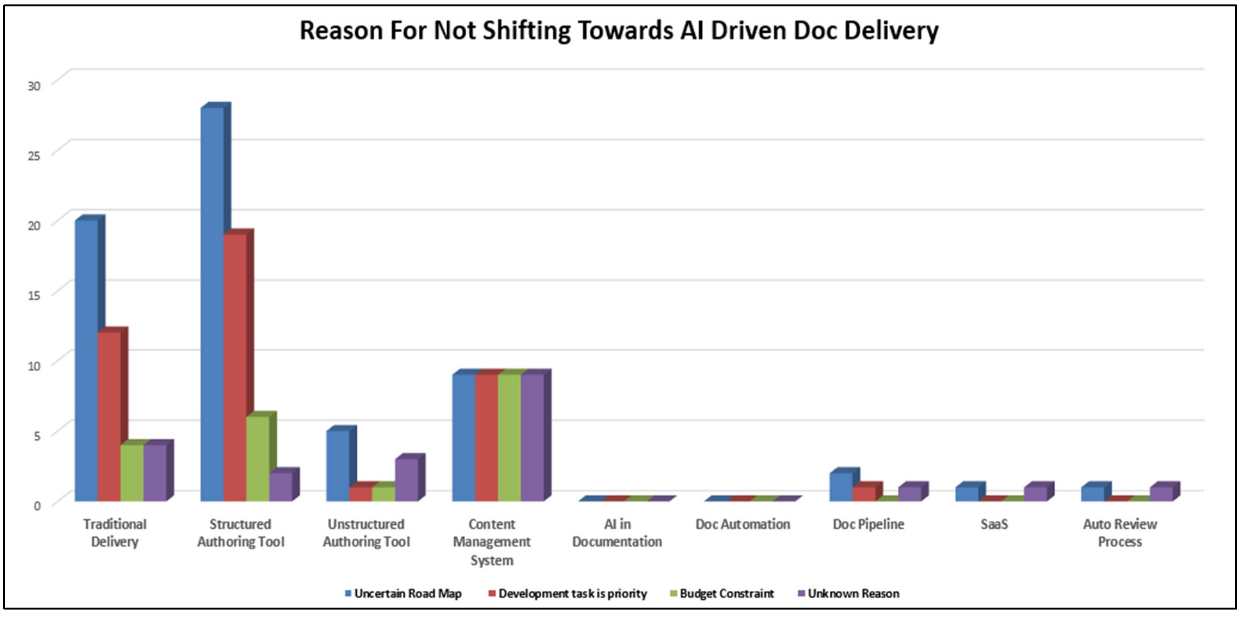

Figure 4: Depicts the Technical Documentation distinct trends and future potentials. The height of the vertical bars represents the proportion of responses aligning with the legends specified in the chart with respect to the survey questionnaire.

The use of AI in documentation and auto-review processes, with road maps showing uncertainty, indicates a possible path of vast improvement. As it’s still in their early stages, these technologies are reflected by their minimal impact on development tasks, budget constraints, and unknown reasons. Overall, each approach has its strengths and areas for improvement, with varying team sizes, defined roadmaps, and uncertainties.

4. Discussion

The "Docs-as-code" approach offers a promising method to standardize the software documentation delivery and encourages the stakeholders for collaborative authoring. It holds the potential to revolutionize workflows, making documentation generation, publication, and maintenance more straightforward and efficient. The documentation CI/CD pipeline improves doc deliverables integration with code and application deployment by streamlining software development processes and reducing repetitive doc publication tasks [24]. Open-source tools such as Confluence, MediaWiki, and GitHub/docs facilitate collaborative authoring. Integrating CI/CD with documentation can be challenging for the stakeholders due to the widespread use of desktop-based tools. Often, it's managed by technical writers with limited exposure to the solution coating aspects. Integrating documentation with the CI/CD pipeline can drive innovation in software development.

Collaboration between technical and UX writers is crucial for creating high-quality product documentation. Your expertise and input are not just important, they are integral as you work closely with subject matter experts to meet industry standards. However, the shift from traditional on-premises distribution to a SaaS model has led to gaps in documentation delivery. The rapid pace of development often places more emphasis on coding than documentation, resulting in incomplete product documentation that may be insufficient in critical scenarios.

4.1 Key takeaways for practitioners (developers, technical writers, product teams)

Treating documentation as an integral part of the development process (e.g., via DocOps) ensures it stays up to date with rapid software changes. By integrating documentation into CI/CD pipelines, teams can reduce the risk of documentation drift. Documentation should not be viewed as a secondary task or solely the responsibility of technical writers. Encourage cross-team collaboration, where developers, testers, and writers collectively own and contribute to the documentation [13]. Collaborative platforms and tools can streamline this process. Explore AI-driven documentation tools that can auto-generate content from code or track user behavior to make updating documentation easier. Practitioners should explore the potential of AI-driven documentation tools and conduct experimental research on their effectiveness in maintaining accuracy, reducing manual effort, and improving the user experience. Insights from this research could drive broader adoption of such tools. To develop a clear structure that addresses the needs of both novice and expert users, interactive tutorials, videos, and contextual help can be planned by developers and administrators.

4.2 Key takeaways for researchers

The researchers should focus on the empirical analysis on documentation practices. Conduct studies that provide quantitative data on how different industries and software development practices impact content quality. Researchers need to focus on how teams manage documentation under Agile, DevOps, or hybrid models would offer valuable insights [14]. In addition, there is a need for researchers to investigate the impact of user behavior, preferences, and engagement with documentation on their learning and adoption processes. Researching different types of documentation (static vs. interactive) to understand the influence on user satisfaction and efficiency would help refine best practices. Investigate how documentation needs and practices evolve in emerging fields such as AI, blockchain, and quantum computing. As these technologies grow in complexity, understanding the unique documentation challenges they present could yield new methods and tools for practitioners. The practitioners focus should be on improving documentation workflows through collaboration, automation, and user-centric design. For researchers, there are numerous opportunities to deepen the understanding of documentation practices, with empirical research needed to address gaps in knowledge around user behavior, AI applications, and organizational influences.

4.3 Limitations of the Study

This article highlights several challenges and prospects in software product documentation, but it is important to acknowledge certain limitations. This article is largely conceptual and does not incorporate empirical studies or quantitative data on how prevalent these challenges are across different industries or software types. A more thorough investigation based on real-world case studies would provide a stronger foundation for the observations presented. The study believes broad applicability across software development environments, but different industries (e.g., healthcare vs. finance vs. open source) may experience varying degrees of difficulty with documentation. Further industry-specific insights could enrich the analysis. While emerging trends like AI-driven documentation are mentioned, the study does not delve into experimental or cutting-edge practices that may soon reshape the landscape. The field of software documentation is rapidly evolving, and new methodologies could emerge that are not covered here.

The Article touches on personalization of documentation but lacks a detailed analysis of how varying user behavior or experience levels affect the effectiveness of documentation. More research on user feedback and interaction with documentation could provide deeper insights into user-centric designs. While the challenges are discussed from a documentation standpoint, the study does not consider how organizational culture, budget constraints, or team structure influence documentation quality. These internal factors play a crucial role in the success of documentation efforts and should be explored further.

These limitations suggest the need for future research that includes empirical data, industry-specific case studies, and a more thorough exploration of user behavior and organizational influences. Further analysis can provide a more comprehensive view of the challenges and opportunities in software product documentation.

5. Conclusions

The field of technical documentation is constantly progressing aligning with new approach emerging in the landscape. The rapidly evolving landscape of software development, sophisticated architectures, and increasing user expectations pose significant pressure on the software product documentation delivery practices. Space is evolving more with more niche AI based fascinating solutions, such as automating content creation, enhancing search capabilities, and transforming documentation creation, maintenance, and consumption. Despite persistent challenges, embracing automation, encouraging collaboration, and evolving documentation processes can help surmount these obstacles. AI and NLP have revolutionized technical writing, with AI-powered chatbots and NLP algorithms elevating content creation and user support. It's worth emphasizing that technical writers play a vital role in elucidating complex concepts, and their significance will endure in the future of human-AI collaboration. Future research scope in this field of research can be summarized as follows:

- **Enhanced AI Algorithms:** Developing AI that better understands context and nuance to reduce inaccuracies in documentation.
- **Privacy and Security:** Ensuring AI tools comply with data privacy regulations and secure user data.
- **Human-AI Collaboration Models:** Exploring optimal ways for humans and AI to work together in technical writing.
- **User-Centric Documentation:** Creating more personalized and interactive documentation experiences.
- **Integration Strategies:** Developing seamless methods to integrate AI tools into documentation workflows without disruption.

Combining these technologies can accelerate content development, enhance translation accuracy, and improve user experience. It is crucial to seamlessly integrate AI into workflows and tackle these concerns, highlighting the significance of understanding and valuing the audience.

Supplementary Materials: Supplementary Material has been provided in the form of Supplementary Table and File.

Author Contributions: Bishnu Shankar Satapathy: conceived the study, performed research, developed required methodology, analyzed data, and authored the paper; Siddhartha Sankar Satapathy: analyzed data and authored the paper; S. Ibotombi Singh and Joya Chakraborty: authored the paper and supervised the research.

Funding: This research received no external funding.

Data Availability Statement: All the data has been shared as Supplementary Material.

Acknowledgments: Bishnu Shankar Satapathy acknowledges CMDR, Tezpur University for providing a research facility.

Conflicts of Interest: The authors declare no conflicts of interest.

References

[1] Hou X, Zhao Y, Liu Y, Yang Z, Wang K, Li L, Luo X, Lo D, Grundy J, Wang H (2024) Large Language Models for Software Engineering: A Systematic Literature Review. ACM Transactions on Software Engineering and Methodology. https://doi.org/10.1145/3695988

[2] Bhat A, Shrivastava D, Guo J (2024) Do LLMs Meet the Needs of Software Tutorial Writers? Opportunities and Design Implications. Proceedings of the 2024 ACM Designing Interactive Systems Conference. https://doi.org/10.1145/3643834.3660692

[3] Arya D, Guo J, Robillard M, Graziotin D, Nolte A, Penzenstadler B, Stol K, Destefanis G, Soria A, Abrahão S (2024) Why People Contribute Software Documentation. Proceedings of the 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering. https://doi.org/10.1145/3641822.3641881

[4] Boll A, Rani P, Schultheiß A, Kehrer T (2024) Beyond code: Is there a difference between comments in visual and textual languages? Journal of Systems and Software. https://doi.org/10.1016/j.jss.2024.112087

[5] Karlsson S, Jongeling R, Čaušević A, Sundmark D (2024) Exploring behaviors of RESTful APIs in an industrial setting. Software Quality Journal. https://doi.org/10.1007/s11219-024-09686-0

[6] Wu J, He H, Gao K, Xiao W, Li J, Zhou M (2024) A comprehensive analysis of challenges and strategies for software release notes on GitHub. Empirical Software Engineering. https://doi.org/10.1007/s10664-024-10486-0

[7] Silva G, Canedo E (2024) Designing from experience: on the adoption of chatbot conversational design practices by practitioners. Personal and Ubiquitous Computing. https://doi.org/10.1007/s00779-024-01800-0

[8] Messer M, Shi M, Brown N, Kölling M (2024) Grading Documentation with Machine Learning. Artificial Intelligence in Education. https://doi.org/10.1007/978-3-031-64302-6_8

[9] Bakare A, Ciancarini P, Farina M, Kruglov A, Okonicha O, Smirnova M, Succi G (2023) Learning from West African storytellers. Frontiers in Computer Science. https://doi.org/10.3389/fcomp.2023.1183602

[10] Khan J.Y. and Uddin G. (2022) Automatic code documentation generation using GPT-3. Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering. ACM, Rochester MI USA, 1–6. https://doi.org/10.1145/3551349.3559548

[11] Gao H, Treude C, Zahedi M, Chandra S, Blincoe K, Tonella P (2023) Evaluating Transfer Learning for Simplifying GitHub READMEs. Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. https://doi.org/10.1145/3611643.3616291

[12] Gilmer S, Bhat A, Shah S, Cherry K, Cheng J, Guo J (2023) SUMMIT: Scaffolding Open-Source Software Issue Discussion Through Summarization. Proceedings of the ACM on Human-Computer Interaction. https://doi.org/10.1145/3610088

[13] Nassif M, Robillard M (2023) A Field Study of Developer Documentation Format. Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3544549.3585767

[14] Bhat A, Coursey A, Hu G, Li S, Nahar N, Zhou S, Kästner C, Guo J (2023) Aspirations and Practice of ML Model Documentation: Moving the Needle with Nudging and Traceability. Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3544548.3581518

[15] Kanei F, Hasegawa A, Shioji E, Akiyama M (2023) Analyzing the Use of Public and In-house Secure Development Guidelines in U.S. and Japanese Industries. Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3544548.3580705

[16] Theo Theunissen, Uwe van Heesch, Paris Avgeriou, A mapping study on documentation in Continuous Software Development, Information, and Software Technology, Volume 142,2022,106733, ISSN 0950-5849,https://doi.org/10.1016/j.infsof.2021.106733

[17] Heetae Cho, Seonah Lee, and Sungwon Kang. 2022. Classifying issue reports according to feature descriptions in a user manual based on a deep learning model. Inf. Softw. Technol. 142, C (Feb 2022). https://doi.org/10.1016/j.infsof.2021.106743

[18] Mary Poppendieck, Tom Poppendieck, Lean software development: an agile toolkit, Computer (ISSN: 0018-9162) 36 (8) (2003) 89, http://dx.doi.org/10.1109/MC.2003.1220585.

[19] Christian R. Prause, Zoya Durdik, Architectural design, and documentation: Waste in agile development? in: 2012 International Conference on Software and System Process, ICSSP 2012 - Proceedings, in: ICSSP '12, IEEE Press, Piscataway,NJ, USA, 2012, pp. 130–134, http://dx.doi.org/10.1109/ICSSP.2012.6225956.

[20] V. Jackson, A. van der Hoek and R. Prikladnicki (2022) "Collaboration Tool Choices and Use in Remote Software Teams: Emerging Results from an Ongoing Study," IEEE/ACM 15th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 2022, pp. 76-80, doi: 10.1145/3528579.3529171.

[21] Christian Manteuffel, Dan Tofan, Heiko Koziolek, Thomas Goldschmidt, Paris Avgeriou (2014) Industrial implementation of a documentation framework for architectural decisions, in: Proceedings - Working IEEE/IFIP Conference on Software Architecture, WICSA 2014, IEEE, 2014, pp. 225–234, http://dx.doi.org/10.1109/WICSA.2014.32.

[22] Rashina Hoda, James Noble, Stuart Marshall, How much is just enough? Some documentation patterns on agile projects, in: Proceedings of the 15th European Conference on Pattern Languages of Programs, ACM, 2010, pp. 13:1–13:13, http://dx.doi.org/10.1145/2328909.2328926.

[23] Woubshet Behutiye, Pertti Karhapää, Dolors Costal, Markku Oivo, Xavier Franch, (2017) Non-functional requirements documentation in agile software development: challenges and solution proposal, in: International Conference on Product-Focused Software Process Improvement, Springer, pp. 515–522, http://dx.doi.org/10.1007/978-3-319-69926-4_41.

[24] Satapathy B.S., Satapathy S.S., Singh S.I. and Chakraborty J. (2023) Continuous Integration and Continuous Deployment (CI/CD) Pipeline for the SaaS Documentation Delivery. In: Hasteer, N., McLoone, S., Khari, M., Sharma, P. (eds) Decision Intelligence Solutions. InCITe Lecture Notes in Electrical Engineering, vol 1080. Springer, Singapore. https://doi.org/10.1007/978-981-99-5994-5_5

[25] Su Y., Wan C., Sethi U., Lu S., Musuvathi M. and Nath S. (2023) HotGPT: How to make software documentation more useful with a large language model? Proceedings of the 19th workshop on hot topics in operating systems (HOTOS '23). Association for Computing Machinery, New York, NY, USA, 87–93. https://doi.org/10.1145/3593856.3595910