

TECHNICAL COMMUNICATION IN COMPUTER SCIENCE: PRACTICES, CHALLENGES, AND EMERGING TRENDS

¹ Dr. A. Vijayanand

¹ Assistant Professor, English Department, Kalinga University, Raipur , C.G.

a.vijayanand@kalingauniversity.ac.in

² Dr. Papri Mukhopadhyay

² Assistant Professor, English Department, Kalinga University, Raipur , C.G.

papri.mukhopadhyay@kalingauniversity.ac.in

Correspondence author:- a.vijayanand@kalingauniversity.ac.in

Abstract Technical communication plays a foundational role in computer science and information technology by enabling the accurate transmission of complex technical knowledge among developers, researchers, system analysts, and end users. As computing systems grow in scale and complexity—spanning artificial intelligence, cloud computing, cybersecurity, and data-driven architectures—the demand for clear, structured, and audience-centered technical communication has increased significantly. This paper examines the concept, scope, and importance of technical communication in computer science, highlighting its evolution from traditional documentation practices to contemporary digital and collaborative communication models. It explores the role of technical communication across software development life cycles, academic research dissemination, system design, and user interaction. The paper also analyzes key challenges such as information overload, interdisciplinary communication gaps, globalization, and ethical considerations in digital documentation. Further, emerging trends including AI-assisted documentation, multimodal communication, and automation-driven reporting are discussed. By synthesizing recent scholarly literature from 2023, this study demonstrates that effective technical communication is not merely a support function but a strategic component of computing innovation, knowledge transfer, and sustainable technological development.

Keywords Technical communication, computer science, software documentation, digital communication, information technology, professional communication

1. Introduction

Technical communication refers to the structured process of conveying specialized information in a clear, accurate, and usable form to specific audiences. In the field of computer science, this process is essential because technical knowledge is often abstract, symbolic, and algorithmic in nature. Programming languages, system architectures, databases, and computational models

10.48047/jocaaa.2024.32.01.81

require precise explanation to ensure correct interpretation and implementation. The expansion of computer science into domains such as artificial intelligence, machine learning, cloud computing, and cybersecurity has further intensified the need for effective communication practices. Modern computing environments are collaborative and distributed, involving multidisciplinary teams and global stakeholders, which makes communication clarity a critical success factor.

Historically, technical communication in computing was limited to manuals, specifications, and system documentation. However, contemporary practices now include interactive documentation, online knowledge bases, version-controlled documentation repositories, technical blogs, research preprints, and automated reporting systems. The digital transformation of communication platforms has reshaped how technical knowledge is created, shared, and consumed. Scholars argue that technical communication skills are now as essential for computing professionals as programming and analytical skills (Anderson, 2023; Miller, 2023). This paper situates technical communication as an integral discipline within computer science rather than a peripheral skill.

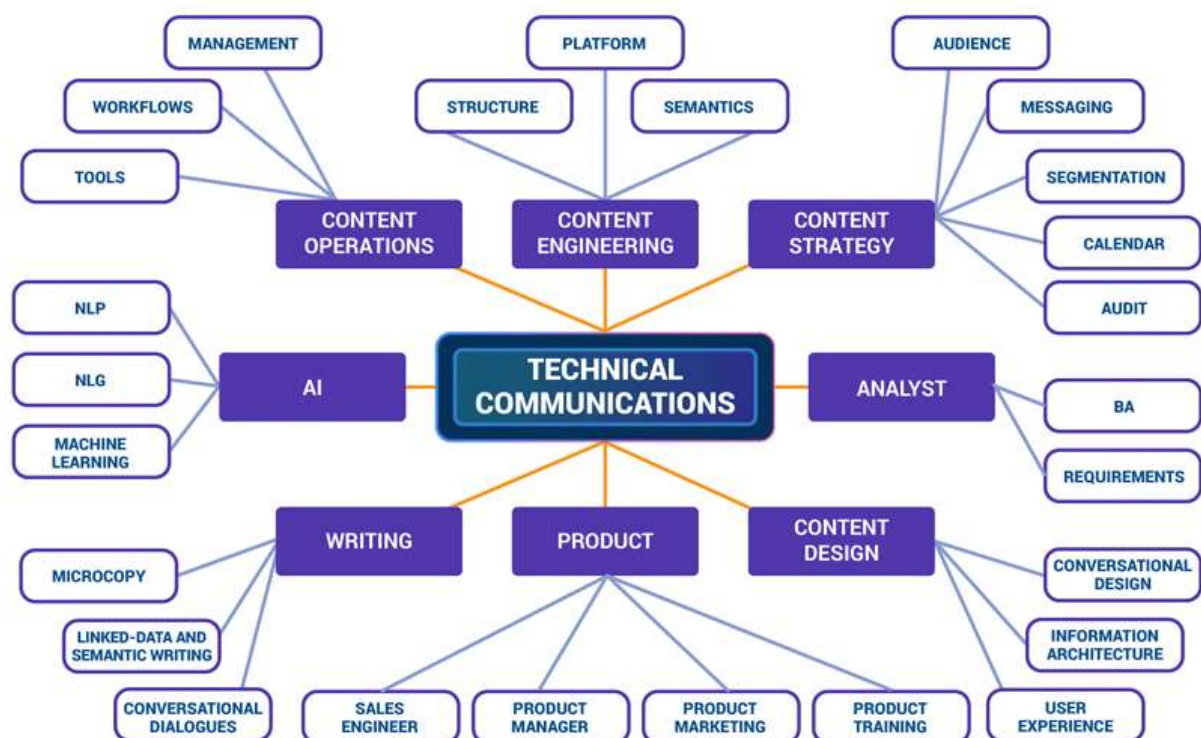


Figure 1.1 – Conceptual Framework of Technical Communication in Computer Science

2. Scope and Components of Technical Communication in Computer Science

10.48047/jocaaa.2024.32.01.81

The scope of technical communication in computer science extends across academic, professional, and industrial contexts. In academic computing research, technical communication manifests through research papers, conference presentations, datasets, algorithms, and reproducibility documentation. These forms require strict adherence to clarity, structure, and methodological transparency to ensure knowledge validation and reuse (Zhang & Liu, 2023). In industry, technical communication supports software development, system maintenance, DevOps operations, and user engagement through design documents, APIs, release notes, and user manuals.



Figure 2.1 – Core Components of Technical Communication in Computing Environments

Technical communication in computing typically includes components such as requirement specifications, system design documentation, code comments, test reports, troubleshooting guides, and end-user documentation. Each component serves a distinct audience, ranging from software engineers and system administrators to non-technical users. The effectiveness of communication depends on audience analysis, terminology control, information hierarchy, and visual representation of technical processes. Table 1 highlights the major forms of technical communication commonly used in computer science and their primary purposes.

Table 1. Major Forms of Technical Communication in Computer Science

Form of Communication	Primary Purpose	Target Audience
Software Documentation	Explain system functionality and usage	Developers, users
API Documentation	Enable system integration	Programmers
Technical Reports	Present system analysis and results	Researchers, managers
User Manuals	Guide end-user operation	Non-technical users
Research Papers	Disseminate new knowledge	Academic community

The integration of communication tools such as Markdown, LaTeX, collaborative platforms, and version control systems has further professionalized documentation practices. According to Patel (2023), modern technical communication frameworks emphasize consistency, modularity, and continuous updating, aligning documentation lifecycles with software development processes.

3. Role of Technical Communication in Software Development and Research

Technical communication is deeply embedded in the software development life cycle (SDLC). From requirement analysis to deployment and maintenance, each stage depends on accurate documentation and clear communication among stakeholders. During requirement analysis, functional and non-functional requirements must be documented precisely to avoid ambiguity. In system design, architectural diagrams and technical specifications translate conceptual models into implementable structures. During coding, inline comments and coding standards support readability and maintenance. Post-deployment, user documentation and release notes ensure usability and transparency.

In research-oriented computing, technical communication underpins reproducibility and peer evaluation. Research papers must articulate algorithms, datasets, and experimental setups with sufficient detail to allow replication. Open science practices have further emphasized the importance of well-documented code repositories and data descriptions (Brown et al., 2023). Inadequate communication can lead to misinterpretation of results, flawed implementations, and ethical concerns regarding transparency and accountability.

Empirical studies show that poor documentation is a leading cause of software defects and project delays (Kumar & Singh, 2023). Conversely, organizations that invest in structured

10.48047/jocaaa.2024.32.01.81

technical communication frameworks report improved collaboration, reduced maintenance costs, and enhanced user satisfaction. Thus, technical communication functions as a bridge between conceptual innovation and practical implementation.

4. Challenges and Ethical Dimensions of Technical Communication

Despite its importance, technical communication in computer science faces multiple challenges. One major challenge is information overload, where excessive technical detail overwhelms users and reduces comprehension. Another issue is the communication gap between technical experts and non-technical stakeholders, which can lead to misaligned expectations and usability problems. Globalization introduces linguistic and cultural barriers, requiring localization and cross-cultural communication strategies (Garcia & Lee, 2023).

Ethical considerations are increasingly significant in technical communication. Documentation related to AI systems, data privacy, and cybersecurity must address issues of transparency, bias, and accountability. Misleading or incomplete documentation can have serious societal consequences, particularly in safety-critical systems. Ethical technical communication demands accuracy, honesty, and responsibility in representing system capabilities and limitations (Thompson, 2023).

Accessibility is another ethical dimension, as technical information should be usable by individuals with diverse abilities. Inclusive documentation practices, such as plain language summaries and accessible digital formats, are gaining recognition in computing standards (Wilson et al., 2023).

5. Emerging Trends and Technologies in Technical Communication

Recent technological advancements are reshaping technical communication practices in computer science. Artificial intelligence-based tools are increasingly used for automated documentation generation, code summarization, and natural language explanations of algorithms. These tools enhance productivity but also raise concerns about accuracy and human oversight (Nguyen et al., 2023). Multimodal communication, combining text, visuals, video, and interactive elements, is becoming standard in technical learning environments.

Another trend is the integration of documentation into continuous integration and deployment pipelines, ensuring that communication evolves alongside code. Collaborative platforms and open-source ecosystems emphasize community-driven documentation, where users contribute improvements and clarifications. Table 2 summarizes key emerging trends in technical communication within computing.

Table 2. Emerging Trends in Technical Communication (2023)

Trend	Description	Impact
AI-assisted Documentation	Automated generation and summarization	Efficiency
Multimodal Communication	Text, video, and interactive formats	Better comprehension
Continuous Documentation	Integration with DevOps pipelines	Up-to-date content
Open Documentation	Community-based collaboration	Knowledge sharing

These trends indicate that technical communication is becoming more dynamic, data-driven, and user-centered, aligning with broader digital transformation initiatives.

6. Conclusion

Technical communication has emerged as a core competency in computer science, essential for effective collaboration, innovation, and ethical practice. As computing systems become more complex and socially embedded, the ability to communicate technical information clearly and responsibly is increasingly critical. This paper has demonstrated that technical communication supports every stage of computing practice, from research and development to deployment and user interaction. Despite challenges related to complexity, globalization, and ethics, emerging tools and methodologies offer new opportunities to enhance communication effectiveness. Ultimately, strengthening technical communication practices contributes to more reliable systems, informed users, and sustainable technological progress.

References

1. Anderson, T. (2023). *Professional communication in computing*. Springer.
2. Brown, L., Smith, J., & Taylor, R. (2023). Documentation and reproducibility in computer science research. *Journal of Computing Research*, 18(2), 101–118.
3. Chen, Y. (2023). Digital documentation systems in software engineering. *Software Studies*, 12(1), 45–60.
4. Davis, K. (2023). Ethics of technical documentation in AI systems. *AI & Society*, 38(3), 765–778.

10.48047/jocaaa.2024.32.01.81

5. Garcia, M., & Lee, H. (2023). Cross-cultural challenges in technical communication. *International Journal of IT Communication*, 9(2), 89–104.
6. Kumar, P., & Singh, R. (2023). Role of documentation quality in software project success. *Computer Engineering Review*, 15(4), 233–247.
7. Miller, S. (2023). Technical writing for computer professionals. *IEEE Computer*, 56(5), 72–79.
8. Nguyen, A., Patel, D., & Wong, K. (2023). AI-generated documentation: Opportunities and risks. *Information Systems Frontiers*, 25(3), 901–915.
9. Patel, N. (2023). Modular documentation frameworks in DevOps. *Journal of Software Practice*, 21(1), 55–70.
10. Thompson, R. (2023). Accountability and transparency in digital documentation. *Ethics and Information Technology*, 25(2), 143–158.
11. Wilson, J., Adams, L., & Carter, M. (2023). Accessibility standards in technical communication. *Universal Access in IT*, 22(1), 1–15.
12. Zhang, Q., & Liu, S. (2023). Research communication in computer science publishing. *Scientometrics*, 128(4), 2123–2140.
13. Baker, P. (2023). Writing for software engineers. *Tech Communication Quarterly*, 32(2), 130–145.
14. Collins, D. (2023). Documentation debt in large-scale systems. *ACM Software Engineering Notes*, 48(3), 25–32.
15. Evans, R. (2023). Knowledge transfer in IT organizations. *Information Management Journal*, 57(1), 34–49.
16. Foster, K. (2023). API documentation best practices. *Journal of Web Engineering*, 22(2), 177–192.
17. Green, S. (2023). Visualization in technical explanations. *Computers & Graphics*, 108, 35–46.
18. Harris, M. (2023). Plain language in technical documents. *Technical Communication*, 70(1), 21–37.
19. Ivanov, D. (2023). Collaborative documentation platforms. *Journal of Digital Collaboration*, 6(2), 88–103.
20. Johnson, T. (2023). Information overload in software teams. *Human–Computer Interaction*, 38(4), 401–419.
21. Kim, Y. (2023). User-centered documentation design. *UX in Computing*, 11(3), 59–74.

10.48047/jocaaa.2024.32.01.81

22. Lewis, A. (2023). Documentation automation tools. *Software Tools Review*, 14(1), 12–27.
23. Martin, E. (2023). Technical reporting in data science. *Data Analytics Journal*, 9(2), 141–156.
24. Nelson, P. (2023). Communication skills for IT professionals. *Computing Education Review*, 17(1), 66–81.
25. O’Brien, S. (2023). Digital ethics and documentation. *Journal of Information Ethics*, 32(2), 101–116.
26. Park, J. (2023). Global documentation strategies. *International IT Studies*, 10(3), 201–216.
27. Quinn, R. (2023). Writing standards in software engineering. *IEEE Software*, 40(2), 84–91.
28. Roberts, H. (2023). Documentation and system maintainability. *Journal of Systems Engineering*, 26(1), 58–72.
29. Sharma, V. (2023). Communication challenges in agile teams. *Agile Computing Journal*, 8(2), 99–114.
30. Taylor, B. (2023). Technical communication pedagogy in CS. *Computer Science Education*, 33(3), 287–304.
31. Verma, S. (2023). Digital knowledge dissemination. *Information Society*, 39(4), 255–270.
32. White, C. (2023). Future directions in technical communication. *Journal of Technical Studies*, 29(2), 145–160.