An Analysis of Communication Practices in the Field of Software Development

Evelyn Drake

Case Western Reserve University

ENGL 217A: Business and Professional Writing

Dr. Stephanie Redekop

October 24, 2024

Abstract

This report examines the communication demands of a professional software developer, focusing both on written and oral communication requirements in daily and long-term project work. Through an interview with a software developer with 2 years of professional experience, combined with research on communication in technical fields, this report highlights the crucial role of concise and clear communication in client interactions, development team collaboration, and documentation processes. These findings reveal that successful developers need strong technical writing skills for code documentation, effective interpersonal skills for team coordination, and the ability to communicate complex technical issues to non-technical stakeholders. Communication in this industry commonly takes the form of text (through messaging applications such as Slack) and video conferences (with video applications like Microsoft Teams and Zoom). These communication practices are essential for the career development of junior software developers, particularly for college students entering the field for the first time.

Introduction

The purpose of this document is to provide college-level aspiring software developers with insights into the real-life communication demands of the profession. Drawing from an interview with a Cleveland-based software developer and research on workplace communication in the tech industry, this report identifies key communication skills that are essential for success in the field.

The report begins by describing background information about the workplace and the research methods used, followed by a detailed discussion of the communication tasks associated with software development. The findings first emphasize the importance of communication in both documentation and collaboration, and then move on to highlight the specific genres and forms of communication used in daily work, such as emails, instant text messages, written documentation, and video conferencing. The report concludes by offering recommendations on how students can better prepare for the communication demands of the software engineering field.

Methods

For this report, I interviewed Serena Lynas, a full-stack web developer who has studied at Case Western Reserve University with 2 years of experience in professional software development. The interview took place over the phone and lasted around 20 minutes. I prepared a list of open-ended questions and took brief notes as she was speaking. To supplement this interview, I conducted research on professional communication within the tech industry. I used digital academic libraries like JSTOR to identify four relevant peer-reviewed articles. Synthesizing both primary and secondary sources allows me to provide a close look at the daily responsibilities of a software developer while backing up my more generalized claims with large-scale peer-reviewed data.

Background

One of the most important aspects of a software developer's job is creating and maintaining technical documentation—written plans describing functionality and usage. This includes internal documentation, such as code comments and developer guides, as well as external documentation for clients and project managers. A consistent writing style and clarity are critical,

and experience in technical writing is an important asset for an aspiring software developer. The presence of professional technical communicators on a software development team can significantly improve project outcomes. This is largely in part due to the critical importance of a requirements analysis, which is the study of the user's organizational needs, behaviors, concerns, and current problems (Bresko 215). These analyses remain relevant throughout the entire project's life cycle, and it becomes increasingly expensive and difficult to change them as the development continues, making it critical to clearly establish this information as early as possible. Because a project relies on these analyses for design, programming, funding, and monitoring, an accurate assessment and communication of these needs can prevent cost and schedule overruns, inadequate and inaccurate documentation, inappropriate assignment of development personnel, and poor user satisfaction.

In addition to technical documentation, communication also plays a large role in effective teamwork. Software is generally developed in accordance with a clearly defined methodology, which outlines a specific system to promote communication throughout a development team. One of these approaches is the Agile model, which is the most commonly used development model in the field (Butt et al. 224). The model, which was used by Lynas's team, consists of a robust, cyclical series of stages (requirement analysis, design, development, testing, deployment, review), which are constantly repeated to adapt to changing client requirements at any stage of a software project's life cycle (Butt et al. 255). While this approach excels with smaller teams, it is now used in large-scale software development, which creates unique communication challenges as teams become larger and larger. Butt et al. identifies and presents several solutions to these challenges, and their work serves as a clear affirmation of the importance of efficient communication in the software industry.

Having established the importance of communication in both documentation and teamwork, we can now examine the specific technologies and formats that are commonly used to accomplish these tasks. One of the most common technologies used to facilitate team collaboration is GitHub, a version control system that allows developers to work on the same project simultaneously, track their changes, communicate issues, monitor progress, and delegate responsibilities (Hundhausen et al. 1). In addition to this, technologies like Slack enable developers to communicate instantly through text in an organized fashion, making collaboration easy even when the team is not physically in the same location (Hundhausen et al. 1). Furthermore, video conferencing with technologies like Microsoft Teams, Google Meet, and Zoom is commonly used in the Agile methodology mentioned above, as it facilitates daily standup and retrospective meetings (Nakhlis 35). Ms. Lynas also affirmed that she communicates with her team members using Slack and Microsoft Teams (Lynas).

Findings

Clear, effective documentation is arguably one of the most important foundations for a software project's success. A study found that 20% to 50% of information systems fail, and one of the major causes of these failures is an inadequate understanding of the user's needs and problems (Bresko 215). As mentioned previously, an accurate requirements analysis performed by someone with technical writing skills is key to alleviating these issues. A survey conducted on the owners, managers, and system analysts of 240 software companies in the United States highlights the importance of technical communication in the field and identifies the most critical

qualities of a technical communicator. It found that 67% of the businesses included technical writers on their teams, 63% of the managers reported that these writers wrote documentation for software, and 15% of the average development team included technical writers alongside system analysts, programmers, managers, and office support staff (Bresko 216). Furthermore, 85% of managers considered their company's documentation techniques to be either "effective" or "very effective" (Bresko 216). Clearly, technical writing and communication is a vital component of the software industry. While this survey was primarily focused on the role of technical writers, it also emphasizes the importance of having software development skills alongside communication skills, demonstrating that it is advantageous for a software developer to also possess technical writing skills. The survey found that a multidisciplinary approach to the role was essential, where 45% of respondents claimed that a computer science background was the second most important discipline for a technical writer to have, and 83% of respondents claimed that they would hire technical writers with this knowledge to work on their development teams (Bresko 217). My interview with Ms. Lynas also revealed that writing documentation for both technical and non-technical stakeholders was a large component of her work (Lynas). Given the multitude of problems that result from poor documentation, communication proficiency clearly improves project outcomes, and programmers with strong technical communication skills are demonstrably valuable assets to software companies.

Additionally, the specific methodology and practices used to enable a development team to communicate and work with one another play a critical role in project outcomes. While there are a variety of different development methodologies, the Agile approach is the most frequently used, guiding both small and large-scale development teams (Butt et al. 225). It extends beyond rigid documentation, as its cyclical nature allows the team to learn as they go and adapt to changing customer requirements. Another component of this approach is daily standup meetings, which enable developers to work more efficiently and clearly identify and communicate their issues and progress (Butt et al. 225). However, when this methodology is applied on a larger-scale (as it is increasingly today), several issues arise, including resistance to change, an overaggressive roll-out schedule, issues with quality assurance, and struggles integrating with pre-existing non-Agile development operations (Butt et al. 225). In an attempt to solve these communication issues, a survey was conducted on three software development companies to identify potential solutions. The authors found that daily standup meetings were hindered by developers' struggles to gain a clear sense of their teammates' work, unclear requirements with inconsistent complexities, the team's inability to integrate the project's modules, and a lack of team motivation (Butt et al. 230). The solutions to these issues all involved communication—all teams should have increased ability to see their teammates' changes, all members should be involved in complexity analysis, and individual responsibilities should be clearly coordinated upfront (Butt et al. 230). This study emphasizes the importance of communication in the software development process and provides evidence-based solutions to help developers excel in their respective teams.

As mentioned previously, this communication often takes the form of GitHub contributions, Slack messages, and Microsoft Teams video conferences. One study shows that a team member's chat contributions are positively correlated to their peer evaluation ratings and their number of GitHub contributions. The researchers focused on two teams working on software development projects for undergraduate online computer science courses at two universities (Hundhausen et

al. 7). Both teams used the Agile method to organize their development, GitHub to track features and issues, and Slack (used by Lynas's team) or Microsoft Teams for online chat communication (Hundhausen et al. 7). The researchers found that chat activity is significantly correlated to development progress (tracked with GitHub) and peer evaluation scores, which matched a study the authors reviewed which found that chat activity is strongly associated with increased development progress in a professional setting (Hundhausen et al. 16). In Lynas's workplace, developers generally message each other on Slack at arbitrary times as they run into issues, and they respond to each other quickly (Lynas). Their tone in these discussions is generally informal, but they always communicate especially professionally over emails, which are sent out less frequently. This type of spontaneous written communication allows developers to quickly solve problems as soon as they arise, improving project outcomes and productivity.

In addition to online written communication, software teams often communicate through video conferencing, utilizing services such as Zoom, Microsoft Teams, and Google Meet, especially in light of the COVID-19 pandemic, which caused the amount of time spent in voice calls and video meetings to double (Nakhlis 1-3). While video conferencing may have a variety of technical issues and a unique potential for stress, it can also promote job satisfaction, project success, and team cohesion (Nakhlis 4). In 2022, a study was conducted which surveyed 180 members of remote development teams to explore these relationships (Nakhlis 46). They found that higher time spent in weekly meetings creates a higher sense of team cohesion, higher team sizes increase job satisfaction, and higher numbers of team members in a call simultaneously increase social cohesion and team pride (Nakhlis 110). The author also collected open-ended feedback from the respondents in an attempt to find recommendations for improving the experience of video conferencing. 24 answers pertaining to job satisfaction and 35 answers pertaining to team cohesion included concerns with audio and video quality, and 21 answers pertaining to job satisfaction suggested including new capabilities to the video conferencing technologies such as improved screen sharing and calendar invite functionality (Nakhlis 109). Ms. Lynas attends daily standup meetings over Microsoft Teams, allowing her to speak with her team and communicate issues and progress (Lynas). Overall, while improvements can be made, video conferencing is indisputably a vital part of the software development process that promotes progress, collaboration, job satisfaction, and team cohesion. Because standup meetings for Agile development teams occur daily, they are one of the most important communication demands of the software development field.

Discussion

The findings from both the interview and research conducted underscore the important role communication plays in every stage of software development. From writing technical documentation, participating in Agile standup meetings, and communicating complex issues to non-technical stakeholders, a developer's ability to clearly convey ideas is vital to their success.

In addition, tools like GitHub, Slack, and video conferencing platforms (Microsoft Teams, Google Meet, Zoom, etc.) have become integral to modern software development, especially after the COVID-19 pandemic. These tools streamline collaboration and bridge the gap between remote and on-site developers. In our interview, Lynas emphasized how her communication generally consists of both structured daily video conferences and spontaneous Slack messages, and she also spoke about the importance of these practices. The literature reviewed further

supports the idea that clear, efficient communication correlates with better project outcomes, higher team cohesion, and increased job satisfaction. Thus, developing strong communication skills is essential for software developers at all stages of their careers.

Conclusion

In conclusion, effective communication is crucial for a software developer's success. Communication plays a role in precise documentation, team collaboration, and client engagement, making it vital for college students to cultivate communication skills in addition to their technical ones. Students should practice writing clear and concise technical documentation because of how critical it is to the development process. These skills can be practiced by documenting personal projects, even if no one else will ever work on them, and contributing to the development and documentation of open-source project, which can be contributed to by anyone with a GitHub account. Secondly, students should familiarize themselves with industry-standard communication tools like Slack, Microsoft Teams, and GitHub, which are all used to facilitate collaboration in real-world development environments. Gaining proficiency with these technologies will allow aspiring developers to be better equipped to integrate seamlessly into development teams, communicate progress efficiently, and troubleshoot issues collaboratively. As mentioned previously, making contributions to open-source projects on GitHub is especially important for aspiring developers who may not have had the experience of working with a development team before, because they can practice skills that they may not have encountered when working on a solo personal project (pull requests, formal documentation of issues, etc.).

Finally, developing strong oral communication skills is equally important. Participating in group projects and internships can provide opportunities to practice explaining ideas, developing solutions, and giving presentations, preparing developers to lead Agile standup meetings, contribute to design discussions, and communicate effectively with both technical and non-technical stakeholders. In addition to being beneficial to the software development process, these communication skills can be helpful for networking and career growth, which is especially important for a developer who is entering the workforce for the first time. Being able to clearly communicate professionally through text improves networking prospects when using online services like LinkedIn, and effective verbal communication helps networking efforts in person. Furthermore, being able to communicate with non-technical stakeholders can be extremely beneficial for those looking to gain experience through freelance work, as a small business might only contract a single software developer who is expected to present solutions and provide updates on their progress. These efforts will better prepare students to meet the communication expectations of the software development industry, positioning them for career success and enabling them to contribute meaningfully to their future teams.

References

Bresko, Laura L. “The Need for Technical Communicators on the Software Development Team.” *Technical Communication*, vol. 38, no. 2, 1991, pp. 214–20.

Butt, Shariq Aziz, et al. “The Importance of Robust Communication in Large-Scale Agile Development.” *Procedia Computer Science*, vol. 236, Jan. 2024, pp. 224–32. *ScienceDirect*, https://doi.org/10.1016/j.procs.2024.05.025.

Christopher Hundhausen, et al. “Combining GitHub, Chat, and Peer Evaluation Data to Assess Individual Contributions to Team Software Development Projects.” *ACM Transactions on Computing Education*, vol. 23, no. 3, Jan. 2023. Association for Computing Machinery, *EBSCOhost*, https://doi.org/10.1145/3593592.

Lynas, Serena. *Software Developer Interview*. Phone, 20 Oct. 2024.

Nakhlis, Boris. *Relationship of Video Conferencing to Job Satisfaction and Team Cohesion in Virtual Software Development Teams*. 2024. Open access content.