

Jason Yau

Professor Hensley

ENC 1102

5 July 2023

The Importance of Communication in Software Development

With the rise of remote work, especially in the field of software development, it is a common misconception that programming requires little collaboration, and much is done in isolation. Software has become increasingly complex and larger throughout the years, and it would be infeasible to code with little communication; Google's codebase consists of approximately two billion lines as of 2016 (Potvin and Levenberg). As a Computer Science major interested in software engineering, I would like to explore the importance and effects of communication in software development further.

I would like to explore the effects of software development methodologies, such as the principles found in agile development and DevOps. For example, the agile methodology involves breaking a project into phases and writing code in a collaborative, adaptive, and pragmatic manner instead of following a set plan. This leads to developers being more open to critiquing and inquiring about project requirements and allows for reflection, flexibility, and creativity in problem solving (Kumar 149–150). I would also like to explore the significance of open-source software (OSS), where programmers develop software and post the source code in public for free, typically in a GitHub repository or similar. Open-source projects are highly collaborative and often dependent on one another; much of the internet relies on OSS: the Linux Kernel (which powers most web servers), Android, Bitcoin, Firefox, etc. Open sourcing allows collaborators to identify and fix bugs and vulnerabilities, write and translate code documentation, and share knowledge with one another (Constantino et al., 5–6). Furthermore, I'd like to explore the effects of the lack of communication. For example, in late 1998, NASA decided to launch a spacecraft to Mars to study its atmosphere, climate, and surface. However, in 1999, the spacecraft evaporated on contact with Mars' atmosphere. Upon investigation, it was found that one team was using the

imperial system, pound-seconds, while the other assumed the metric system, Newton-seconds (NASA). Had the two teams communicated and agreed to the standard metric system used in most physics calculations, the incident, which led to hundreds of millions of dollars in financial damages, wouldn't have happened.

Overall, I believe that this topic is an excellent choice for the semester-long study in an English class, as I can write about the importance of effective communication and writing skills needed to facilitate maintainable code with a clear and consistent style and documentation. I have personally collaborated on projects that ended up collapsing due to poor communication and busy schedules. I believe that this topic would be very useful for other software engineers to study, as it will increase productivity and decrease complications in the workspace.

Works Cited

Potvin, Rachel, and Josh Levenberg. "Why Google Stores Billions of Lines of Code in a Single Repository." *ACM*, 1 July 2016,

<https://cacm.acm.org/magazines/2016/7/204032-why-google-stores-billions-of-lines-of-code-in-a-single-repository/fulltext>.

Kumar, Shreya. *Communication Patterns and Strategies in Software Development Communities of Practice*. Michigan Technological University, 2016,

<https://www.mtu.edu/cs/graduate/dissertations/pdfs/communication-patterns-and-strategies-in-software-development.pdf>.

Constantino, Kattiana, et al. *Understanding Collaborative Software Development: An Interview Study*. Carnegie Mellon University, 2020,

<https://www.cs.cmu.edu/~ckaestne/pdf/icgse20.pdf>.

"Mars Climate Orbiter." *NASA*, 6 June 2023,

<https://solarsystem.nasa.gov/missions/mars-climate-orbiter/in-depth/>.