Jack Pearce

Prof. Brad Stiles

EN280 – Writing in the Disciplines

October 3, 2023

Discourse Community Ethnography Project

The aim of this project is to provide an in-depth understanding of the software development discourse community. This software development ethnography includes the community's shared goals, communication mechanisms, purposes of communication, genres, specialized language, and the intricate dynamics of expertise and learning. Input will be included from an interview I conducted with Luke Hewitt, a software developer at Hanson Professional Services and an expert in this discourse community.

The definition of the software development discourse community is a group of individuals who share a common interest in designing, developing, and maintaining software systems. This community includes those trying to learn their first line of code, software engineers, developers, testers, project managers, and others involved in the software development process.

The software development discourse community represents more than a professional sphere; it's a closely-knit ecosystem characterized by shared objectives, unique communication practices, and a distinctive language. At the heart of this community is the ACM (Association for

Computing Machinery) citation format, serving as a cornerstone of scholarly communication. This paper delves into its inner workings and how its members collaborate and communicate.

The shared goals that are a part in the software development discourse community's existence is what makes the group's identity. Extensive observation, interviews, and document analysis uncover the overarching objectives that bind its members together. Beyond the lines of code and digital platforms, there's a shared pursuit of excellence, innovation, problem-solving, and knowledge dissemination. These shared goals form the cornerstone of the community's identity.

Effective communication is the lifeblood of any discourse community, and software developers are no exception. Soon after exploring the community, an outsider would quickly realize the diverse means by which they interact and collaborate. When working in teams, there are traditional in-person meetings or modern digital platforms like GitHub and Slack, there's adaptability and fluidity in their communication mechanisms. Within the software development community, each communication mechanism serves a distinct purpose. Meetings facilitate real-time brainstorming and decision-making, while emails streamline formal exchanges. Platforms such as Stack Overflow and Stack Exchange cater to specific knowledge-sharing needs. Whether it's enhancing performance, troubleshooting challenges, or disseminating research findings, motivations driving these interactions are explored. In the interview with Luke Hewitt, he was asked "How do you communicate with others to solve problems whether it be through the means of team members or the internet?". His response was that he used Microsoft Teams to chat or call daily to ask team members or clients questions. Whenever there was a problem that could not be resolved in house, he would use Stack Overflow to get an answer to his problem.

Textual responses or genres are the threads that weave the fabric of communication within the community. From writing code in a programming language to code repositories, discussion threads, and code comments, each genre fulfills a specific function in mediating interactions. These genres not only streamline discussions but also provide structured formats for sharing knowledge and offer platforms for peer review. Sites such as Stack Overflow, GeekforGeeks, and a specific programming language's documentation site like Python.org for learning Python all help a developer that is budding in a new area of the field figure out how to solve the problem they are having. Each of these websites have two or three of these genres interwoven into them.

One of the most detailed and confusing pieces of the software development discourse community is its lexicon of specialized language and terminology. In many cases, referencing one specific thing may have multiple terms that can be utilized and are preferred depending on what the programmer's primary language is. For instance, the use of the term "function" in Python differs from "method" in Java, yet both refer to similar programming concepts. This specialized language streamlines communication, enabling members to convey complex ideas efficiently.

Here is an extremely small sample of terms that are used by software developers with the discourse community:

Code is a sequence of instructions that a programmer writes to tell a device (like a computer) what to do. The device cannot know by itself how to handle a particular situation or how to perform a task. So, developers are in charge of analyzing the situation and writing explicit instructions to implement what is needed.

Python is a programming language that consists of object-oriented instructions for web and app development. It is an open-source language that is easy to read and write.

Java is an object-oriented programming language that produces software for multiple platforms. When a programmer writes a Java application, the compiled code runs on most operating systems, including Windows, Linux and Mac OS.

Variables in coding are names for data that can change. They are like containers or boxes that store information. Variables can be used to remember and use the information in a program.

In any community, the distribution of expertise is a critical aspect. Through terminology used and the way a programmer codes, the "old-timers" with extensive knowledge and the newcomers navigating the community's intricacies are quickly identified. Luke Hewitt helps confirm this in the interview saying, “It is hard to fathom how far I’ve come since I was a freshman in college. It is like night and day if you were to go back and look at my old code and look at what I wrote up today. My current boss Bob offered me a position my sophomore year as an intern here at Hanson and it gave me the opportunity to learn from the real-deal software developers. I am very thankful for that.” Hewitt reflects on his past abilities and recognizes the vast difference between now and then. He also recognizes how the member of the discourse helped out someone they didn’t necessarily need to in order to pass on their knowledge. This is ultimately how knowledge and language are transferred, fostering the growth of new members.

Whether it is through in-person teaching or advice through a text, it is crucial that the community passes knowledge onto future members in order to maximize efficiency and technology growth in the future. Mentorship and knowledge-sharing, often occurring through code reviews and collaborative projects, play an indispensable role in maintaining the community's vibrancy.

Beyond the technical aspects, the cultural and behavioral nuances of the software development discourse community come with an expectation to be followed. In software development, it is a must to know how to properly name your variables in your code if a solution is wanted to a problem that is occurring. Otherwise, if on a site such as Stack Overflow, the community will not answer a question unless variables are named properly. This was confirmed by Luke Hewitt in the interview, saying that he “has seen it happen firsthand” when being asked whether people judge variable names on Stack Overflow. One also should know their terminology and use it properly to avoid being talked down to. It may not always be the case that someone will be criticized for the phrasing of their question or syntax of their code, nevertheless it is best to ensure one has checked those things before creating a post. Non-compliance often leads to exclusion or criticism, highlighting the community's particular nature.

This ethnography provides an expansive definition and understanding of the software development discourse community. Through a thorough exploration of its shared goals, communication mechanisms, purposes of communication, genres, specialized language, expertise dynamics, and community etiquette, valuable insights into the identity and functioning of this vibrant and interconnected community are offered. The findings emphasize that the software development discourse community transcends professional boundaries, evolving into a dynamic ecosystem of collaboration and knowledge exchange. Regardless of geographical locations, members are brought together through the internet by their shared passion for innovation and problem-solving, facilitated by a lexis that streamlines communication.

Works Cited

Anon. 2023. Common programming terms used in software development. (April 2023). Retrieved October 2, 2023 from https://fullscale.io/blog/common-programming-terms/

Anon. A computer science portal for geeks. Retrieved October 2, 2023a from https://www.geeksforgeeks.org/

Anon. Stack exchange. Retrieved October 2, 2023b from https://stackexchange.com/

Anon. Welcome to Python.org. Retrieved October 2, 2023c from https://www.python.org/

Anon. Where developers learn, share, & build careers. Retrieved October 2, 2023d from https://stackoverflow.com/

Jack Pearce and Luke Hewitt. 2023. (September 2023).