Highly Valued But Often Overlooked: A Field Study of User Documentation Tools In Open Source Software

Chris Holland

North Carolina State University

Cmholla6@ncsu.edu

Research Summary

Changing rhetorical situations and technologies in the past decade have placed technical communicators in a unique position: adapt and innovate or stagnate and become obsolete. Much like the technologies we study and interact with, the future of technical communication needs to adapt quickly to shifting needs of the user especially in the open source software industry. Despite its origins in the computer science programs of academia, open source software has shifted away from its roots, and where many proprietary softwares have adopted a user-center model of design, some open-source technologies still rely on a developer-first, or systems-centered model of production(Yeats,2008) which creates a disruption in the information flow between developers, technical communicators and ultimately, the user. These disruptions can have dire consequences for the user: often, the user is unable to find the information that they need to complete a task with the product documentation are forced to search for information elsewhere, like in community discussions boards or forums(Swarts,2018) either facilitated by the software itself or hosted on a third-party site. This presents a problem of inefficiency and frustration for users, leading to user dissatisfaction and an increased workload just to find an answer to a problem or to complete a task. My research seeks to find ways that tools can improve communication flow in the open source software community by examining its activity systems.

I find that activity systems theory is a useful lens to explore the information delivery problems in open source software because at its core, activity theory is concerned with how people work together, using tools, toward outcomes(Kain and Wardle,1986). Considering this lens, activity systems theories can be easily adapted to the open source software model, which relies heavily on tools and community collaboration to produce products and communication. Using the activity systems framework, the tools of open source communication--especially in regards to user support-- can be examined within the context of their larger socio-cultural systems: As people participating in activity systems learn, and as new people join the activity, they refine their tools and create new ones.(Kain and Wardle, 1986) To better examine the activity systems of open source software, I will conduct a field study which will employ data collections methods such as document collection, field observation, and semi-structured interviews. It is my hope that this research will identify opportunities for technical communicators to explore new tools and methods that facilitate information sharing in open source software projects.

Problems In Open Source Software

When viewing complex problems such as information flow in organizations, a sociocultural model is often helpful to view the systemic issues(Spinuzzi,2013) that are often the cause of communication problems within communities like open source software. Activity theory posits a clear asymmetry between communicators and their tools and technologies (McNely, Spinuzzi,Testa, 2015) and I believe that further research of open source software communities using the activity systems lens will be helpful for uncovering problems, or conflicts between the various dimensions of the open source activity system. The individual dimensions that make up an organization’s activity system are: Objective, outcome, division of labor, community stakeholders, rules, actors and tools. The use of tools in community information spread is perhaps one of the most crucial elements of an activity system. Therefore, a bulk of my research will focus on methods to analyze and identify opportunities for technical communicators to improve user experience through tools by adopting activity systems maps and contradictions maps(Spinuzzi,2013).

Research Questions

1. How can activity systems methods help technical communicators pinpoint where communication issues are happening in open source software as they relate to user software help?
2. How can activity systems uncover potential opportunities for technical communicators to implement new tools for an improved user experience for help-based communication?

Literature Review

Contemporary research in this niche area has been difficult to locate; however, Dave Yeats directly addresses it in *The Role of The Technical Communicator in Open Source Software.* Through the dialog of scholar- critic , Yeats argues that open source software often fails to meet the needs of users due to the systems-centered framework that open source technology is typically developed under. This approach, rather than a user-centered design often happens because many users of open source are the same people who write the code that makes it work(Yeats,2008). This is still true to an extent but in 2020, open source projects have become much more dynamic in that developers have realized that users are important contributors. Even still, users cannot get the help they need from documentation-- a phenomenon that Jason Swarts covers in *Wicked, Incomplete and Uncertain: User Support in the Wild and the Role of the Technical Communicator*. Although Swarts only examines a few open source software projects in his research, his work is focused on the shifting roles of the technical communicator as more users look outside of software documentation resources and increasingly to help communities for solutions to their wicked, often incomplete problems that traditional user documentation fails to address.(Swarts 2013) This is a problem of communication between key actors so I have decided to use the frameworks of activity systems theory adapted from Clay Spinuzzi's *Topsight* methods of activity systems maps and contradiction maps. These theoretical lenses and methods, when applied to an open source software project, will help technical communicators not only visualize the communication problems but will also help codify them into actionable insights that can be used to improve tools and methods to deliver information to the end users.

Research Methods

Phase One

Data collection will be the first phase of this project and will allow me to examine the nature of the problem with users. I will collect various types of user documentation from the carefully selected open-source projects ranging from online help, user guides and quick start guides to gain insight of the project's documentation flow. It will also give me an idea of the various tools used to deliver information to users, a crucial data point that will help me recognize any gaps that need to be addressed with participants. Using this data, I will create an initial activity system map with a focus on the relationships between tools, actors and objectives.

Phase Two

Activity systems are a sociocultural study and as their activities change, people use their tools differently and modify their tools to meet their changing needs(Kain and Wardle,1986). To understand how current tools are used within open source software, interviews and observations of technical writers in the field are necessary design components for activity system research to uncover any systemic issues or contradictions between activity systems dimensions(Spinuzzi, 2013). I will conduct pre-observational and post-observational semi-structured interviews with 5-10 developers and technical writers within each open source software project. The pre-observational interviews will focus on questions derived from the phase one analysis with a specific focus on the dynamic between developers, technical writers and tools used. The post-observational interviews will focus on questions related to data collected from the observation sessions. These interviews will allow participants to engage in a dialog about their work which will open up discussion about their respective parts in the activity system, any miscommunications about tools and perceptions of the software user's needs.

Observations will consist of brief, 1-hour sessions with the software developers and technical communicators. During the sessions, I will watch perform their work and collect field notes for further analysis.

Phase 3

Phase three of this study will assemble and reorganize data gathered from phase two observations and interviews into new activity map. Re-structuring the activity systems maps serves a two-fold purpose: it will make visualization of the entire system easier which helps to clarify any tenuous relationships and it will highlight any gaps in communication, or contradictions between dimensions(Spinuzzi,2013). For these contradictions, a separate map called an activity systems contradictions map(Spinuzzi, 2013) will be useful in determining which dimensions have active contradictions. The goal of this method is to evaluate each contradiction and their connection to established problems in user documentation.

Contradictions can also be a helpful way to open up important dialog about any uncovered issues. These conversations can then be transformed into useful sites of innovation(Spinuzzi,2013) and potential problem solving catalysts for technical communicators in open source projects.

Schedule For Completion

Before any activity systems work can be performed, I must first complete some initial tasks. I need to first identify four potential open source software projects and create an initial case study that outlines the problems of user documentation. This work will be completed no later than January 2021. I will have document collection and initial activity systems maps completed by late January 2021. In early February 2021, I will conduct initial interviews with the software developers and technical writers and complete field observation. By March 2021, I should be ready to complete post-observation interviews and restructure the activity maps. This will leave me enough time to build out my research methods and create contradictions maps by the end of the month. April 2021 is scheduled for drafting and editing with a project completion deadline of May 3rd, 2021.

Potential Barriers

I anticipate that there will be some barriers involved in this research, primarily because many open source projects have a completely digital infrastructure and lack physical buildings. Gaining permission to study the project should be fairly easy due to the nature of open source projects. Field observations; however, may prove to be challenging. The participants may be willing to use a screen sharing software so I can observe their work. This will need to be discussed during the initial project proposal. Locating participants could also be difficult. Often, open source software developers and technical writers are volunteers and may be weary of any perceived extra work of being involved in a research study. Even though the projects themselves will be within the open source community, privacy and confidentiality are still expected and standard research protocols will be deployed to manage sensitive information.