Network Driven, Open Source Collaboration and Higher Education: Academic Parallels

Jess Sullivan

Abstract - Devoid of admission costs, paywalls, busywork, career choice implications and social biases, the online software community exemplifies the longstanding principles of higher education; an economy of knowledge, specialty, and sharing. As public, digital collaboration becomes ubiquitous across occupational specialties and academic study, a natural extension of leadership and authority from institutional Higher Education will follow. This paper investigates the parallels between digital, open source collaboration and the traditional Higher Education system through which academic values are upheld.

Introduction

"The OSS development process mimics the academic knowledge creation process where gift economies are central to the social system" (Quint-Rapoport 2012)

"While the university in the learning economy can act as a cluster to promote regional development through thickening its layers of collaboration with government, business and the local community, it can also play a role, based on these synergies and local clusters, to initiate a wider international development role." (Peters 2006)

These economy analogies from Peters and Quint-Rapoport describe the most recent evolution in scholarly discussion and collaborative research. Human inquiry has seen countless iterations of organization and institutional support, continuing to reside in concentrated, brick and mortar institutions of Higher Education. The rapid growth of network-driven, open source development methods can be described as Computer Science (CS) scholars in traditional University environments assimilating familiar academic expectations to their blossoming field (Peters

2006).

Involvement in OSS is a continuation of our innate, insatiable need to learn and engage academically, identical to the grounding of Higher Education. The scholarly origin of OSS is easily overlooked with the following global adoption of general purpose internet connectivity. Despite a widespread acceptance of the internet, systematic collaboration models investigated during the pioneering decades of institutionalized CS continue to evolve among CS scholars and developers, remaining largely uncharted for practitioners of other disciplines (Charles et al. 2005). Fundamentally, the geographically dispersed development models prevalent across the digital OSS and CS landscape are not specific to the focused fields from which they grew.

Core Features of a Public OSS Project

Regardless of an individual developer's niche or motive, essentially all OSS projects share three important features: shared commentary and work organization (Haddad et al. 2011), a Version Control System (VCS) (Ram 2013), and a way to ask for and give advice (Raymond 1999).

A well implemented public software organization system allows any visitor or contributor to efficiently understand the basis of a project, the source material structure, how to use or contribute to the work, find external reference materials and information on prerequisite technologies, and read about any current approaches, issues, work goals, and points of discussion. Because of the inherent disconnect between internet collaborators, clear documentation and sharing helpful resources is by far the most important part of OSS development. One must assume each visitor to public project materials does not have an identical background or reason to be interested. Providing an "in" for those less experienced in OSS or developers who are simply unfamiliar with the languages or solutions employed by other contributors is crucial to the success of a public OSS project. While there are numerous solutions to provide this organizational structure, the formats and nomenclature generally stay the same; at the time of writing, many developers default to providing these systems through the ubiquitous Version Control host Github. Some common elements of OSS organization include a "readme"- often formatted with markdown- containing vital project information, instructions to start working with the project, and links for the various resources expected from a shared organizational system. If applicable, usage instructions are frequently formatted for UNIX-like consoles, such as the Bourne-again Shell (Bash) or Fish.

Using a Version Control System such as SVN, Mercurial, or Git is an expected practice across all software teams, both public and private. In the simplest form, a VCS allows team members to work on the same files independently while systematically keeping track of edits, giving each developer control over conflicting modifications (Atlassian n.d.). However, the nature of most VCS implementations provide OSS groups with a number of advantages in addition to tracking changes. Because edits are worked through on copies of project files in an individual developer's computer prior to making the new changes available to everyone as a new, shared set of project files, there is no danger of disrupting the shared assets as participants make and work through mistakes. This large reliance on independent learning and problem solving inherently leads participants to document their struggles and successes- a logical intersection of personal note taking, communication for peers, informal attribution and simple pride of one's work. For OSS projects, the culture and language of clearly sharing one's thinking and discoveries accelerates participation among new contributors and interested individuals.

References

Atlassian et al. (n.d.). What is Version Control. Retrieved from https://www.atlassian.com/git/tutorials/what-is-version-control

Charles, S., Tom, E., & J, M. (2005). Open source and open content: A framework for global collaboration in social-ecological research. Ecology and Society, 10(1). doi:10.5751/ES-01287-100133

Haddad, I., Warner, B. (2011). Understanding the Open Source Development Model. The Linux Foundation.

Peters, M. (2006). Higher education, development and the learning economy. Policy Futures in Education, 4(3), 279-291.

Quint-Rapoport, M. (2012). Open source in higher education: Towards an understanding of networked universities. Policy Futures in Education, 10(3), 315-327.

Ram, K. (2013). Git can facilitate greater reproducibility and increased transparency in science.

Raymond, E. (1999). The cathedral & the bazaar: Musings on linux and open source by an accidental revolutionary. Beijing: O'Reilly. (1999).

Torvalds, L. (1999). The linux edge. Communications- Acm, 42(4), 38-39.