International Journal of Open Source Software and Processes
Volume 14 • Issue 1

Organizational Influencers in Open-Source Software Projects

Roland Robert Schreiber, Information Systems and Social Networks, University of Bamberg, Germany*

ABSTRACT

Traditional software development is shifting toward the open-source development model, particularly in the current environment of competitive challenges to develop software openly. The author employs a case study approach to investigate how organizations and their affiliated developers collaborate in the open-source software (OSS) ecosystem TensorFlow (TF). The analysis of the artificial intelligence OSS library TF combines social network analysis (SNA) and an examination of archival data by mining software repositories. The study looks at the structure and evolution of code-collaboration among developers and with the ecosystem's organizational networks over the TF lifespan. These involved organizations play a particularly critical role in development. The research also looks at productivity, homophily, development, and diversity among developers. The results deepen the understanding of OSS communities' collaborative developer and organization patterns. Furthermore, the study emphasizes the importance and evolution of social networks, diversity, and productivity in ecosystems.

KEYWORDS

Ecosystem, Evolution, Open-Source, Organizational Influence, Social Network Analysis, Software Development, TensorFlow

INTRODUCTION

In an environment of technological complexity and competitive challenges, software is no longer developed only in-house by a single firm. Instead, there is a collaboration among a community of volunteers, in-house developers, developers from partner companies, universities, and even competitors (Bengtsson & Kock, 2000; Ghobadi & D'Ambra, 2012). Numerous famous firms support OSS projects and cooperate with others (Capiluppi et al., 2012).

This collaboration with competitors has given rise to the concept of "coopetition," a term describing the coexistence of competition and cooperation and the interaction between companies that have a partial congruence of interests (Linåker et al., 2016; Nguyen Duc et al., 2017). There are many examples of open-source initiatives exemplifying coopetition such as WebKit, Blink, OpenStack, and CloudFoundry (Jansen, Finkelstein, & Brinkkemper, 2009; Teixeira & Lin, 2014; Nguyen Duc et al., 2017). Another example of a large-scale programming cooperation is autonomous vehicles. In this context, the experience

DOI: 10.4018/IJOSSP.318400 *Corresponding Author

of all autonomous cars will train every autonomous car and learning from others' failure (Wiseman, 2022). The rate of learning and gaining experience is faster with this approach than when compared to a sole human driver whose only training source is personal experience.

There are, however, increasing software development challenges in this space, including the development of essential innovative products, long geographic distances between participants across different time zones, cooperation, and open software standards (Frischbier et al., 2012; Holmstrom et al., 2006; Ouriques et al., 2019; Xia et al., 2016). In addition, considerable participatory development becomes more important as the complexity of software projects increases (Çaglayan & Bener, 2016; Wohlin et al., 2015). These OSS projects also depend significantly on the effective use of basic software components, interactions among people, and the human factors identified as key to successful software projects (Biçer et al., 2011; Chow & Cao, 2008; Holmstrom et al., 2006; Oliveira et al., 2018; Wohlin et al., 2015; Wu & Tang, 2007).

This paper explores the developers' and organizations' dynamic informal structure in the TensorFlow (TF) OSS ecosystem over all releases. Employing a case-study approach (Basili et al., 1999), we analyze how collaboration works in the open-source space and, specifically, in the TF OSS ecosystem. We also examine the importance of the top-contributing organizations within the TF OSS ecosystem by considering the social relationship between developers and their affiliations with an organization. The longitudinal case study design is based on applying well-known social network analysis (SNA) techniques, which we use to mine development data over five years. Furthermore, we use the dynamic evolution of the TF networks over time to analyze the characteristics and changing collaboration structures for developers and organizations. Thus, we contribute significantly to closing the software research gap by exploring the role of organizations, developers, sub-communities, and the social network for OSS ecosystems (Abufouda & Abukwaik, 2017; Schreiber & Zylka, 2020). The results reveal new and unexplored aspects of the TF OSS ecosystem that increase our knowledge of informal coordination structures and their evolution with increasing organization participation (Gonzalez-Barahona & Robles, 2013). These aspects are crucial for understanding how a large OSS ecosystem works and help to take appropriate actions if problems emerge, e.g., an ecosystem becomes dominated by a single company. Moreover, the results of this paper are relevant for researchers seeking an overview of TF OSS ecosystem collaboration, insights into ecosystem software development, the role of social networks, and opportunities to contribute to research in this specialized field.

The remainder of this paper is organized as follows. We first present the background and related work. We then formulate the research questions and our research focus before we describe the case study methodology, including the data collection and research methods. The following section details the results of the TF OSS project case study and our evaluation of the results before we provide the limitations and then offer relevant summarizing conclusions.

BACKGROUND

A software ecosystem is the interaction of a set of actors with a common technological platform used by several software solutions (Jansen et al., 2013; Manikas & Hansen, 2013). Table 1 offers several examples of OSS ecosystems and cases of open coopetition.

OSS ecosystems evolve from self-organized and dynamic processes in which volunteers and different firms worldwide contribute to a software product (Gerber et al., 2010; Madey et al., 2002). These ecosystems have a tremendous impact on computing and OSS software development (Zhou et al., 2017), and are gaining importance and supply vital software components and infrastructures, such as operating systems, libraries, component stores, and entire platforms (Ghafele & Gibert, 2014; Jansen, Brinkkemper, & Finkelstein, 2009).

Volunteers from different companies collaborate in these OSS ecosystems. The participating companies gain a mutually beneficial competitive edge through this collaboration that allows them to leverage their own software and services (Barbosa & Alves, 2011; Dagnino & Rocco, 2009; Morgan