Information and Software Technology 70 (2016) 30–39

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Understanding the popular users: Following, affiliation influence and leadership on GitHub

Kelly Blincoe [a,*], Jyoti Sheoran [b], Sean Goggins [c], Eva Petakovic [c], Daniela Damian [b]

[a] Auckland University of Technology, New Zealand
[b] University of Victoria, Canada
[c] University of Missouri, USA

ARTICLE INFO

Article history:
Received 12 April 2015
Revised 10 October 2015
Accepted 12 October 2015
Available online 20 October 2015

Keywords:
GitHub
Open source software
Social computing
Following
Influence
Leadership

ABSTRACT

Context: the ability to follow other users and projects on GitHub has introduced a new layer of open source software development participants who observe but do not contribute to projects. It has not been fully explored how following others influences the actions of GitHub users. *Objective:* this paper studies the motivation behind following (or not following) others and the influence of popular users on their followers. *Method:* a mixed methods research approach was used including a survey of 800 GitHub users to uncover the reasons for following on GitHub and a complementary quantitative analysis of the activity of GitHub users to examine influence. Our quantitative analysis studied 199 popular (most followed) users and their followers. *Results:* we found that popular users do influence their followers by guiding them to new projects. As a user's popularity increases, so does their rate of influence, yet the same is not true for a popular user's rate of contribution. *Conclusions:* these results indicate that a new type of leadership is emerging through GitHub's following feature and popularity can be more important than contribution in influencing others. We discuss implications of popularity and influence and their impact on social structure and leadership on OSS projects.

1. Introduction

Understanding how influence is exerted on social computing platforms is critical for participants and leaders because it impacts their patterns of work, interactions, and knowledge management in collaborative environments. While much work has been done to understand influence on purely social platforms like Twitter and Facebook [1–3], influence on social software development environments like GitHub has been understudied. GitHub enables users to "follow" individuals, much like one follows another user on Twitter. However, users on Twitter broadcast their own messages and diffuse only ephemeral information, links and perspectives 140 characters at a time. In contrast, notifications sent to GitHub followers embody evidence of work that has occurred and are not broadcast by the user, but rather, by the system [4]. The GitHub following feature is, therefore, interesting to study since structure and influence in OSS seem connected to the types of activities participants engage in.

Previous studies of influence in Open Source Software (OSS) found a "pyramid meritocracy" [5–7], where there is hierarchy and centralization, but usually not an authoritarian "great leader" at the top. Leadership is shared by a group of developers who "act as [influential] peers at the top echelon of the pyramid" [7]. The metaphor of a pyramid depicts new contributors at the bottom, "leaders" in the middle and "elders", who may have previously led projects but now advise, at the top. Elders are particularly common on long running OSS projects. OSS projects that are led by a software company may also have a community manager who acts as a liaison between the company and the OSS community [8]. With the advent of GitHub, new types of leadership are emerging because GitHub fosters a lower barrier to entry than other OSS platforms like SourceForge [9].

Other studies of OSS projects (e.g. [10–14]) reveal patterns of organization participants as "a series of concentric circles", where at the center are the core developers. The core developers are surrounded by a ring of "maintainers" who are responsible for one or more modules of a project. The outer-most circle contains the "patchers" who fix bugs, "bug reporters", "documenters" and "users". As Ducheneaut [10] points out, even users can be "highly skilled", suggesting that the "periphery" is a "nebulous arrangement of both skilled and unskilled individuals."

This nebulous skill arrangement is further extended on GitHub through its following feature, which allows users to "observe" without participating. This type of participation has the capacity to alter OSS engagement models. Our research goal is to understand how the lightweight connection of following another user builds into cascades

* Corresponding author. Tel.: +64 226107330.
E-mail address: kblincoe@acm.org (K. Blincoe).

http://dx.doi.org/10.1016/j.infsof.2015.10.002
0950-5849/

of influence and to study the relationship between emerging social structures in projects on GitHub, the most followed (popular) users, and leadership in these projects. In this paper we take a first step in this direction by investigating the most followed users and their influence on their followers.

In this paper, we investigate following behavior and the influence of following others using a mixed-method research approach. We surveyed 800 GitHub users to identify motivations behind following other users on GitHub and why some users choose not to follow others. To examine influence of following, we conducted an analysis of the 199 most popular GitHub users (measured by number of followers). By analyzing the actions of popular GitHub users and their followers, we found that popular users often attract their followers to new projects. This is in line with previous results [15] that found that users who are both very popular and very active influence their followers. In this previous work, only users who fall into both of these categories were included; therefore, the two dimensions were intertwined and unable to be studied individually [15]. Instead, we study all popular users regardless of activity level. This allows a deeper investigation into the effect of popularity. Through our investigation, we found that the rate of a popular users' contribution does not impact her rate of influence. However, the rate of influence of a popular user does increase as the user accumulates more followers. Our findings indicate that GitHub's following feature may be enabling a new type of leadership in GitHub-hosted OSS projects and that popularity may be more influential than actual contribution.

1.1. GitHub: open, collaborative software development environment

GitHub is a web-based, social software development environment that provides source code management, issue tracking and other features. GitHub allows users to set up a public repository that anyone can fork and use for their own code and/or to contribute changes to the code. Pull requests are a way in which code from one developer is contributed back to a GitHub repository publicly. A "fork" is a clone or copy of a repository. Forks are made for two main reasons: first, to use the code in some derivative way; and second, as a precursor to contributing back to the original project through a pull request, which can then be merged with the main branch of code by those with access. All of these activities are published in an open, visible stream on GitHub, and content and discussion associated with issues, commits and pull requests are also public. Users of GitHub can receive alerts (via desktop clients and email) about code changes, pull requests, comments, issues, etc. for any public project.

Some of the social features on GitHub include following other users and starring other projects. Starring a project is a way for users to indicate that the project is interesting to them. Since there are often multiple GitHub OSS projects providing the same functionality, stars are often used as a 'vote' to indicate that the project is worth using and, thus, are a signal of the health of GitHub-hosted OSS projects.

2. Related work

Although research on ways of working in GitHub has been growing rapidly, aspects of project organization, social structures, and leadership and its influence on work performance in such open environments have not received much attention. Dabbish et al. [9] do provide insight into how GitHub's design enables new types of collaboration and collaboration patterns through increased platform transparency for individual users. They discuss the "collaborative utility" and value when "transparency is integrated into a web-based workspace" as well as the value of social features that make activity visible to users – GitHub supports learning "better ways to code and access to superior knowledge" [9]. Not only are GitHub users able to evaluate contributions by examining discussions around those contributions [16], GitHub's social interface also allows people to make inferences about other contributors' "technical goals and vision when they edit code" [9]. This transparency of activity allows users to define "effective strategies for coordinating work, advancing technical skills and managing their reputation" [9]. Through this transparency, new types of leadership and influence may have emerged, but this has yet to be studied.

2.1. Social structure and leadership in open source software development

Studies of social structures and interaction in other working environments that offer transparency exist in the literature prior to GitHub's popularity. Here we highlight studies from the larger body of research on OSS development examining social interactions, through studies of online systems and how norms are developed among online teams (e.g. [10]).

Long and Siau [17] elaborate on "social structure" in SourceForge OSS projects by examining changes in communication network structures over time. They focus on interaction patterns in the bug tracking system in three SourceForge projects, showing that projects evolve from a single "hub" to a core/periphery structure. While these networks are centralized in the sense that a core group of individuals (or "key members") are at the center of communication, "it is decentralized in the sense that the decision or communication core is not concentrated on one or two members but a group of key members" [17].

Similarly, Crowston and Howison's [18] study of communications interactions in SourceForge's bug tracking system reveals the extent to which 120 project teams show uniformity and difference in their social structure. The communication structures surrounding bug fixes were neither consistently centralized nor decentralized. Instead, they display a range of centralization. One key pattern they uncover is that centralization is negatively correlated to both the number of developers and the number of active users contributing to bug reports. The authors offer their interpretation that as "projects grow, they have to become more modular, with different people responsible for different modules ... resulting in what might be described as a 'shallot-shaped' structure with layers around multiple [cores]" [18].

This multi-layered organizational structure was also reported in the study of Mockus, Fielding and Herbsleb [19] who found that, although development is quite centralized (with only 15 developers responsible for more than 80% of the code), centralization decreases dramatically in the context of new code contribution, bug fixes and bug reporting. Their findings lead them to this hypothesis: "In successful open source developments, a group larger by an order of magnitude than the core will repair defects, and a yet larger group (by another order of magnitude) will report problems" [19].

GitHub allows users to "follow" other individuals. The actions (commits, pull requests, comments, etc.) of followed users appear on a dashboard. This creates an additional layer to organizations of individuals who observe but do not participate. Due to the newness of GitHub's following feature, the social structure of followers has yet to be fully explored.

2.2. Following and followers on GitHub

Lee et al. studied a small set of GitHub users and examined how four users who are both the most popular and the most active (rockstars) influenced their followers [15]. They found that when rockstars are more active on the projects they own, they attract more of their followers to contribute to that project. They also saw that these rockstars guided their followers to new projects when they contributed on a new project. This study was limited to three months of activity of the four rockstars. Our study builds on these findings by studying all GitHub users with a large number of followers and