Capturing Spontaneous Software Evolution in a Social Coding Platform With Project-as-a-City Concept

Koji Toda, Fukuoka Institute of Technology, Japan
Haruaki Tamada, Faculty of Information Science and Engineering, Kyoto Sangyo University, Japan
Masahide Nakamura, Graduate School of System Informatics, Kobe University, Japan
Kenichi Matsumoto, Graduate School of Science and Technology, Nara Institute of Science and Technology, Japan

ABSTRACT

Social coding platforms ( SCPs) have realized spontaneous software evolution, where new source code and ideas are spontaneously proposed by altruistic developers. Although there are many projects operated by active communities performing spontaneous evolution, it is yet unclear that how such successful projects and communities have been formed and governed. This article proposes a method that can investigate the history of every project in the SCP. Introducing the concept of project as a city, the authors consider every project in the SCP as a city, where a government and citizens develop a city through collaborative activities. This research then identifies essential attributes that characterize a state of a city. For each attribute, the authors develop metrics that quantify the state $S(p; t)$ of a project $p$ at time $t$. An experimental evaluation investigating GitHub projects of famous code editors shows that the proposed metrics well visualize the history of the projects from essential perspectives of a city.

KEYWORDS

Governance, Project Measurement, Smart City, Social Coding, Software Evolution, Software Metrics

1. INTRODUCTION

Social coding (Trockman, Zhou, Kˇastner, & Vasilescu, 2018) is a modern approach to software development, which places an emphasis on formal and informal collaboration. Due to the emerging social coding platforms (SCP, for short) such as GitHub (Github, Inc., 2008) and Bitbucket (Atlassian, Inc., 2008), the social coding is now widely adopted not only in open-source projects, but in commercial software development.

In the SCP, software is developed and maintained collaboratively by a community of developers and users. The SCP provides a space for the people to create a project, have communication, play and build whatever they want. Moreover, every developer can hack and modify the code as he/she likes by forking the project. Then, the developer can request the project owner to merge his/her modification to the original code. If the owner finds it useful, the owner commits the merge, which creates a new

DOI: 10.4018/IJSI.2020070103

Copyright © 2020, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
version of the software. In GitHub, the above three activities are implemented as commands fork, pull request, and merge, respectively.

From the perspective of software evolution (Lehman, 1980), the above process of code modification is quite unique. Traditionally, when new features and/or better solutions are requested by users or customers, a project manager designs tasks, and directs developers to work in a top-down manner. Then, the developers modify the code according to the requirement. However, in the social coding, the code modification can be proposed by spontaneous and altruistic developers in the community. Thus, the software evolution can be triggered in a bottom-up way. We call this new type of software evolution spontaneous software evolution (Matsumoto et al., 2017).

The spontaneous software evolution by the social coding is a powerful means to achieve fast, flexible, and sustainable evolution of the software. Also, it often derives innovative ideas that a project owner has never imagined, realizing advanced features and products. For example, microsoft/vscode (Microsoft, Inc., 2015) is a GitHub project of a powerful code editor, where a lot of people have discussion, developers actively present codes, and many innovative features have been added. Thus, we can see that microsoft/vscode is a successful project in the context of spontaneous software evolution.

However, we do not know how such a successful project has been managed, or how such an active community has been formed and governed. To clarify these issues, we need to investigate the history, starting when the project was created, and leading to the current shape. Fortunately, since the SCP accumulates various event logs, we can look back how the project was for any past time. More specifically, for a given project $p$ and timestamp $t$, let $S(p, t)$ denote a state of $p$ at $t$. Observing $S(p, t)$ with varying $t$ derives a sequence of states, which characterizes how $p$ has been evolved. The challenge here is how to define $S(p, t)$ so that it well characterizes the spontaneous software evolution within $p$.

The goal of this paper is to propose a method that can cope with the following research questions:

**RQ1**: How and by what can we define the state $S(p, t)$?

**RQ2**: How has the state of a successful project changed?

To address RQ1, we first introduce the concept of project as a city. Borrowing the idea of smart city modeling (Nakamura & du Bousquet, 2015), we consider every project in the SCP as a city, where a government and citizens develop a city through collaborative activities. We investigate a mapping that associates constructs and activities of a city with entities and actions of the SCP. We then identify essential attributes that characterize a state of a city: population, size, urban problem, problem solving ability, capability of government. Finally, we define metrics NSTAR, N-MGCL, N-RIS, R-RIS, LT-CIS, R-MGPR, LT-PR, and R-PRLP. Each metric quantifies a corresponding attribute of a project $p$ at timestamp $t$, which can be calculated from data accumulated in the SCP. Thus, the values of the metrics characterize the state $S(p, t)$, reasonably.

To address RQ2, we conduct an experiment where we observe the evolution of GitHub projects of famous code editors: atom/atom (GitHub, Inc., 2014), adobe/brackets (Adobe, Inc., 2012), and Microsoft/VScode (Microsoft, Inc., 2015). The result showed that the proposed metrics well visualized the history of the three projects from essential perspectives of a city.

2. PRELIMINARIES

2.1. Github; Social Coding Platform

GitHub (GitHub, Inc., 2008) is a cloud-based platform for software development. GitHub uses Git for version control of codes and documents, and also provides many features of social networking. GitHub is now one of the biggest social coding platforms, acquiring 28 million users over the world.
Project Management in Enterprises: IT Implementation Based on Fuzzy Models
www.igi-global.com/chapter/project-management-enterprises/29516?camid=4v1a

Enhancing a SCRM Curriculum With Cybersecurity
www.igi-global.com/article/enhancing-a-scrm-curriculum-with-cybersecurity/221922?camid=4v1a