Ability Level of Git Hub Amongst Computer Science Students

Jos Timanta Tarigan, Elviawaty Muis Zamzami, Ivan Jaya, Sri Melvani Hardi, Muhammad Zarlis

Abstract*: Source code management is one of the most vital tools in modern software development. Moreover, the open source initiative requires a method to distribute their code effectively. GitHub is one of the most popular web-based version control systems that offers version control and source code management. Therefore, it is important to introduce GitHub as early as possible to computer science students. Our objective is to observe the use of GitHub as a source code management system. We perform a survey targeted at the students and collected 298 answers. We also perform a short interview amongst the lecturer regarding their familiarity with GitHub and source control management in general. The survey result shows that while GitHub is quite popular amongst the student, the knowledge and experience of using it is considerably low. Similar responses were received from the lecturer that argues that the role of GitHub in classroom is redundant. The details of the result are thoroughly discussed in this paper.

Index Terms*: Source Code Management System, Software Engineering, Social Coding.

I. INTRODUCTION

Modern software development has become more complex and complicated. It requires a collaboration of multiple developers that usually hard to implement. Problems such as conflict of changes and revisions and version inconsistency amongst developers are two of the main problem it tries to handle. Moreover, integrating multiple versions of codes without adequate information regarding the changes (version, author, and date) is difficult and error-prone. To avoid this problem during software development process, developers commonly use a Source Code Control System (SCCS). While the idea and implementation of a source code management have been around ever since the early years of software development [1]-[2], the term SCCS itself was coined in a paper Marc J. Rochkind [3]. The main purpose of this system is to help developers to manage their source code. It has the capability to keep track of changes or revisions made in a file. This tracking feature is important in software development that involves collaboration of multiple individuals and may help the integration process.

Revised Manuscript Received on August 22, 2019.

Jos Timanta Tarigan, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia.

Elviawaty Muisa Zamzami, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia.

Ivan Jaya, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia.

Sri Melvani Hardi, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia.

Muhammad Zarlis, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia.

There is various source code management software available on the current market; each with its own feature. GitHub is one of the most popular source code control system that not only keep tracks on changes on the software, but also allow users to interact to each other. It tries to apply a social networking interaction style to software development in order to achieve transparency and collaboration during the development process. The idea is by adding social media alike features, collaborative projects can easily find the most potential contributors by browsing through their profile. This feature is crucial to open source project where the availability of contributors is vital to the sustainability of the project. Moreover, since the interaction during software development is done on the web (through discussion feature), new developers can learn a lot of things only by paying attention to the progress of a software development. The GitHub owner, GitHub Inc., called this trend as 'social coding' [4]. Recent studies have shown that the use of GitHub has given various advantages to developers such as job recruitment [5], multitasking [6], and crowd-sourced knowledge [7]. Moreover, GitHub may also increase the reusability of source code [8] by allowing a source code specific navigation and search functions.

However, even though the popularity of GitHub has been growing steadily in the open source community, its use in education is unpopular. There are many issues that caused this problem such as the lack of knowledge of educators regarding the use of GitHub [9] and the fact that GitHub is designed to be fully open and transparent may encourage plagiarism [10]. Furthermore, the lack of need for source code control system in a classroom software development project contributes to the unpopularity of GitHub in education.

The objective of this research is to observe the use of GitHub in education. Our research purpose is to perform a survey amongst computer science students and lecturers regarding their knowledge of GitHub. The survey was distributed in both digital version and physical form. We only took our samples from Computer Science student. To support the result from the student, we also perform an interview with a few lecturers.

The paper is structured as follows: the next section will discuss the research that has been done previously related to our work. The third section contains basic information of GitHub and its features. The survey result will be shown in the fourth section followed by conclusion and future works of our project.

II. RECENT WORKS

There is no doubt that GitHub is one of the most popular source code control system especially in open source community. The popularity attracts researchers to study the trend and impact of GitHub in software development. Ray et al. discusses the code quality in GitHub and try to present the type of programming that is common in GitHub [11]. Net et al. presented a survey on the usage of GitHub's Issue Tracking System [12]. While it agrees that issue tracking is vital in collaborative development, the paper argues that most issues are not solved with contributors commits. Similarly, Bissyande et al. perform a research to observe the behavior of software projects in GitHub regarding the use of issue trackers [13].

Another interesting topic to discuss regarding GitHub is the interaction between users beyond the source code. The term 'social coding' coined GitHub Inc. means that the system encourages users to interact socially. There are many studies focused on observing GitHub social ecosystem such as job advertisement based on developer's profile[5], behavior of GitHub teams with time zone and time-of-day variance [14], and the sentiment analysis of comments in GitHub [15]. But the biggest part of the social feature in GitHub is the discussion. Tsay et al. perform an observation to evaluate the contribution of a discussion to the sustainability of a project [16]. Vasilescu et al. observe the diversity of GitHub teams based on gender and tenure [17]. Begel et al. performed an interview with leaders from four software development social networking application such as MSDN, Stack Exchange, GitHub, and Top Coder [18] where all of those leaders agree that social networking is crucial to create a more collaborative environment amongst developers.

III. GITHUB

GitHub is a web-based source code control system that uses Git version control system. Currently, it is being run by GitHub Inc., a US-based company founded in 2008 and was acquired by Microsoft in 2018 with a $2 billion deal. The company offers free accounts with the ability to create a cloud-based open repository and paid service for a private repository. Aside from the source code control system service, GitHub feature social networking-like functions such as feeds, followers, wikis, and profiles. Thus, the tagline 'social coding' is used to represent the overall activity in GitHub. Currently, GitHub is the largest repository service compared to its competitors such as SourceForge and GitLab.

A. Pull-Request Feature

Recent trend on collaborative software development tend to favor the pull-request development model; that is, instead of waiting for the task to be assigned to a single individual (which is usually called push model), the workflow creates a queue of tasks that allows available developer to pull the task. This model, as explained by Eric Lee [19], is favorable due to the capability to increase team's productivity. In this development style, the role of an integrator is significant. The ability to review and merge changes from multiple (independent) developers affects the pace of the development process. While some integrators struggle to maintain the quality of the project [20], most developer still considers pull-based model as the ideal solution for a more dynamic collaborative software development. Moreover, most open source project uses this model due to its requirement to involve a lot of developers.

GitHub allows contributors to work on a push-based development model, where contributors clone the repository and make their changes independently. When a contributor has finished revising the work, he/she will create a pull-request which in result will create a branch in the main repository. The project main developer, which is called the integrator, will then reviews and integrate the changes to the main project.

B. Discussion

Openness is both the strength and weakness of open source project. Evaluating and ensuring the quality of the contribution is the most essential task considering not all contribution is suitable for the project. Moreover, some contribution can be a threat to the technical integrity of the project. Hence, it is common to find integrators and contributors having a discussion whether a contribution is applicable to the project another interesting feature of GitHub is the ability to communicate with contributors to discuss the technical aspect of the project development.

In the process, this feature apparently has a significant role in improving the project. As suggested by Tsay et al., not only the discussion decide whether an input is acceptable or not, it also provides an alternative to fulfill the goal attempted by the contributors which may enrich the goal of the project and its requirement [16].

C. Social Coding

Social coding is probably the vital feature of GitHub. The feature allows developer to broadcast their activities and listen to others. As one of the biggest (if not the biggest) social coding site, GitHub maintains more than 3 million repositories. It able to reach this milestone by understanding that project sustainability depends not only on the role of the creator but also on the contribution of the other users. By allowing projects to have a treatment similar to social media applications, creators can easily be connected to potential contributors. Thung et al. conducted an investigation in GitHub social coding by collecting information data from 100,000 projects and 30,000 developers[21]. The work shows that there is a strong relationship between social coding and collaboration. The data shows that social coding enables substantially more collaborations among developers.

IV. SURVEY RESULT

We perform our survey by interviewing students and lecturers. We collect sampled 298 students from various universities and collected their answer. The first part of the survey is to understand the habit of the sample regarding their choice of software or development library. Most of our samples still consider proprietary and commercially distributed software tend to have advantages that open source software. Only 16.7% of our samples admitted they prefer to choose an open source

alternative while 9.3% of them admitted they have no interest in looking through open source alternatives. However, more than half of our samples agree that open source software is the best option for software development and the current state of open source software can compete with commercial software in both quality and quantity. More than 85% of our samples admitted they have used an open source project, either as a library or only part of the code, as part of their program.

The second part of the survey is to observe our samples' understanding regarding GitHub. Based on our survey, 71.4% admit they have accessed GitHub at least once. Most of the activity, however, is limited to downloading open source project or observing technical discussion. However, based on the type of activity, only 24% admit they have at least once contributed to an open source project.

The third part of our survey is to observe student understands regarding collaborative software development, source code control system, and open source software development process. 92.7% of our samples understand and have the capability to define the problem in collaborative software development process and the need of source code management such as version control. This finding is expected since most of them have been involved in collaborative software development. However, due to most of the experience occurred in a small-scale and short-term group assignment, our samples were not capable to describe in detail the cause and effect of problems that often occurred in large scale long term project. Moreover, none of our samples has used a source code management application in the development process; all of our samples perform code integration manually by copying source code from other group members' work. Moreover, none of our surveyed students has ever used source code management software in developing a project and none has ever encountered a fatal error during the merging process.

Additionally, we perform our survey to 5 lecturers. Three of the samples have experience in working with a large group software development project and all three have experience in using source code management system. All lecturers are capable to describe the use and capability of GitHub and the importance of the software in open source development process. However, all lecturers agree that the use of source code management software in classroom in non-existent due to the small size of the project. All agreed that forcing student to use source code control system will be redundant and may cause bottleneck in the learning process.

V. CONCLUSION AND FUTURE WORKS

In this paper, we perform a survey to observe the understanding and usage of GitHub in computer science students in various Universities in Indonesia. We collect answers from 298 students by asking a set of question regarding their knowledge of GitHub. The result shows that while some of our samples have ideas what GitHub is, their knowledge regarding the application is low. Most of our samples do not know the main features in GitHub such as pull-request and social coding. Moreover, the result shows that our samples have lack of knowledge and experience regarding source code control system; while most of the students have experience in collaborating software development, they perform the code management manually.

Based on this research, our suggestion is computer science education institution have to integrate GitHub as part of the class. While the use of the application may be excessive for a small and short-term project, the experience and knowledge of students about GitHub is crucial. Not only this will teach the students how to collaborate in a large group development, it can also introduce and encourage them to contribute in many open source projects.

ACKNOWLEDGMENT

The authors would like to thank the head and staff of "Lembaga Penelitian USU (Research Center of Universitas Sumatera Utara)", The author would also like to thank the Dean of Faculty of Computer Science and Information Technology University of Sumatera Utara, Prof. Dr. Opim Salim Sitompul, M.Sc for their widest to support this work.

REFERENCES

1. D. M. Ritchie and K. Thompson, "The UNIX Time-Sharing System," Bell System Technical Journal, vol. 57, no. 6, pp. 1905–1929, 1978.
2. D. M. Ritchie, "The UNIX System: The Evolution of the UNIX time-sharing System," AT&T Bell Laboratories Technical Journal, vol. 63, no. 8, pp. 1577–1593, 1984.
3. M. J. Rochkind, "The source code control system," IEEE Transactions on Software Engineering, vol. SE-1, no. 4, pp. 364–370, Dec. 1975.
4. L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in GitHub: transparency and collaboration in an open software repository." In Proceedings of the ACM 2012 conference on computer supported cooperative work, 2012.
5. C. Hauff and G. Gousios, "Matching GitHub Developer Profiles to Job Advertisements," In Proceedings of the 12th Working Conference on Mining Software Repositories, 2015.
6. B. Vasilescu et al., "The sky is not the limit: multitasking across GitHub projects," In International Conference on Software Engineering, 2016.
7. B. Vasilescu, V. Filkov, and A. Serebrenik, "StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge," In International Conference on Social Computing, 2013.
8. M. Gharehyazie, B. Ray, and V. Filkov, "Some from Here, Some from There: Cross-Project Code Reuse in GitHub," In International Conference on Mining Software Repositories, 2017.
9. A. Zagalsky, J. Feliciano, M.-A. Storey, Y. Zhao, and W. Wang, "The Emergence of GitHub as a Collaborative Platform for Education," In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, 2015.
10. A. Kahlon, B. MacKellar, and A. Kurdia, "Combating the Wide Web of Plagiarism," In Proceedings of the 49th ACM Technical Symposium on Computer Science Education, 2018.
11. B. Ray, D. Posnett, V. Filkov, and P. Devanbu, "A large scale study of programming languages and code quality in GitHub," Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2014.
12. C. C. M. Neto and M. de O. Barros, "A structured survey on the usage of the issue tracking system provided by the GitHub platform," In Proceedings of the 11th Brazilian Symposium on Software Components, Architectures, and Reuse, 2017.
13. T. F. Bissyande, D. Lo, L. Jiang, L. Reveillere, J. Klein, and Y. L. Traon, "Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub." In 2013 IEEE 24th international symposium on software reliability engineering, 2013.
14. P. Devanbu, P. Kudigrama, C. Rubio-González, and B. Vasilescu, "Timezone and time-of-day variance in GitHub teams: an empirical

method and study." In Proceedings of the 3rd ACM SIGSOFT International Workshop on Software Analytics, 2017.
15. E. Guzman, D. Azócar, and Y. Li, "Sentiment analysis of commit comments in GitHub: an empirical study." In Proceedings of the 11th Working Conference on Mining Software Repositories, 2014.
16. J. Tsay, L. Dabbish, and J. Herbsleb, "Let's talk about it: evaluating contributions through discussion in GitHub." In Proceedings of the 22nd ACM SIGSOFT international symposium on foundations of software engineering, 2014.
17. B. Vasilescu, D. Posnett, B. Ray, M. G. Van den Brand, A. Serebrenik, P. Devanbu and V. Filkov. "Gender and tenure diversity in GitHub teams." In Proceedings of the 33rd annual ACM conference on human factors in computing systems, 2015.
18. A. Begel, J. Bosch, and M.-A. Storey, "Social Networking Meets Software Development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder," IEEE Software, vol. 30, no. 1, pp. 52–66, 2013.
19. E. Lee, "Push vs. Pull in Scrum," 2010, [Online]. Available: https://blogs.msdn.microsoft.com/elee/2010/01/21/push-vs-pull-in-scrum/ . [Accessed: 5 August 2018]
20. G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, "Work Practices and Challenges in Pull-Based Development: The Integrator's Perspective." In Proceedings of the 37th International Conference on Software Engineering, 2015.
21. F. Thung, T. F. Bissyande, D. Lo, and Lingxiao Jiang, "Network Structure of Social Coding in GitHub," In 2013 17th European Conference on Software Maintenance and Reengineering, 2013.