Fully Online Project-Based Learning of Software Development during the COVID-19 Pandemic

Atsuo Hazeyama[1[0000-0002-6583-1521]], Kiichi Furukawa[2] and Yuki Yamada[2]

[1] Tokyo Gakugei University, Koganei Tokyo 184-8501, Japan
[2] Graduate School of Education, Tokyo Gakugei University, Koganei Tokyo 184-8501, Japan
[1]hazeyama@u-gakugei.ac.jp

Abstract. The COVID-19 pandemic impelled educational institutions worldwide to deliver online distributed education. We have been developing project-based learning (PBL) style software engineering education since 1997. Because of the COVID-19 pandemic, our PBL course was applied fully online during the 2020 academic year. Our PBL method uses GitHub to manage artifacts and provide feedback to student groups; i.e., the teaching staff perform the inspection process and acceptance testing. To deliver the course during the 2020 academic year, we changed the lecture method from a face-to-face approach to one that uses an online meeting system. All the groups successfully completed their project. In the final assignment, we asked the students to describe any "difficulties they encountered during their remote learning activities and their solutions to these problems." As the results indicate, none of the students experienced any major difficulties. The teaching assistants and instructor did not encounter any major issues in fully online remote PBL because the progress of all groups could be ascertained during a progress meeting held each week, as well as through the inspection process conducted during the upstream phase and the acceptance testing, and thus, any problems could be addressed promptly. GitHub played an important role in this process. Our process and software engineering environments, which are a combination of GitHub and an online meeting system, are suitable for a fully online remote PBL.

Keywords: Fully Online Remote Project-Based Learning, Software Engineering Education, GitHub.

1 Introduction

Educational institutions worldwide were compelled to implement online distributed education during the COVID-19 pandemic [1–8], and our university was among those that conducted online remote lectures during the 2020 academic year.

We have been developing project-based learning (PBL) style software engineering education since 1997 [9, 10]. Because of its characteristics, this learning style relies on the teaching staff giving feedback to the students. The results of software inspection of the artifacts of the upstream phase, acceptance testing for the developed system, and progress checks during lecture time are used as feedback in our PBL approach for software engineering education. The student groups manage all such artifacts in a repository on GitHub, which is also used for the inspection process and acceptance testing.

The PBL course examined in this study was delivered fully online during the 2020 academic year because of the COVID-19 pandemic. This paper reports an overview of the course, the preparations for fully online remote operations, and the results of its implementation. In our study, we attempted to answer the following questions. What difficulties do students with less experience in PBL-style software development encounter in a fully online environment? How are these difficulties manifested?

The rest of this paper is organized as follows. Section 2 describes the state of the art of software engineering education during the COVID-19 pandemic. Section 3 provides an overview of our PBL software development course before the COVID-19 pandemic. Section 4 presents the PBL course we delivered in the 2020 academic year. Section 5 describes the results of this practical implementation. In Section 6, our fully online remote PBL course is evaluated. Finally, Section 7 summarizes this paper.

2 Related work

Several studies have been conducted in the context of software engineering education during the current pandemic [11–18].

Barr et al. described the manner in which intensive online lectures were conducted [11]. Their results showed that the students preferred watching pre-recorded videos over participating in live online and even face-to-face classes, and the authors concluded that flipped learning is an effective means of teaching in such a situation.

In addition, Schmiedmayer et al. distributed lectures in the form of live-streamed pre-recorded videos, during which the teacher and teaching assistants allowed time for questions [12].

Kanij and Grundy presented guidelines for taking full advantage of online learning in which they included the means of making announcements to students, the learning materials, and the assessment method [13].

Mues and Howar claimed that, when the participants in fully online remote PBL do not know one another, it is important to establish communication [14], and described their approach.

Plewnia et al. reported the experience of using PBL for teaching software project labs in collaboration with an industry partner during the COVID-19 pandemic [15]. They found that the greatest disadvantage in this situation was the lack of direct contact between the students and the industry partner.

Bringula et al. investigated the challenges of implementing a programming project course during the COVID-19 pandemic and proposed their solutions to these challenges [16]. They identified 13 challenges; however, they did not evaluate their solutions based on feedback from the students.

Motogna et al. presented an empirical study that was aimed to improve the understanding of the manner in which the assessment of student learning changes in response to the transition from in-class to online courses [17].

Finally, Yamada et al. noted the difficulties related to building a software engineering environment for Web application development using the students' own laptop computers in a fully online remote environment [18]. They developed a script for automatically building the software engineering environment.

Few studies have examined the transition to a fully online remote project-based software engineering education during the COVID-19 pandemic. In particular, to the best of our knowledge, no studies have been conducted on whether students having less experience with project-based software development find it difficult to learn in a fully online remote environment, and if so, what types of difficulties they experience.

3 Overview of our PBL of software development (before the COVID-19 pandemic)

The PBL software development course is offered to third-year undergraduate students in the Department of Informatics Education of Tokyo Gakugei University. The quota of the department is 15 students. Therefore, the PBL course is provided at a small scale. The course consists of 15 weekly 90 min lectures. The task the students are set is Web application development using Java. In the semester preceding that in which the PBL software development course is offered, we deliver an introductory course on software engineering.

Among other aspects of PBL, we specify the software development process, artifacts, and approach to grading in the information provided to the students about the operation of the course and explain them during the first lecture. Each lecture consists of announcements from the instructor or teaching assistants (TAs), explanations of the usage of tools (e.g., GitHub), and group meetings and progress checks. The TAs are Master's students who previously passed this course. Certain activities may need to be conducted outside lecture hours.

Each group consists of 3–5 students. We perform a questionnaire survey for organizing the groups and determine the organization of each group based on the results of both the questionnaire survey and the students' grades in the introductory software engineering course.

The development process is based on the waterfall model. The types of artifacts are requirements specification, user interface design document, class diagram, database design document, sequence diagram, source code, unit/system testing report, development plan, group progress report (each week), and project completion report. A sequence diagram and source code are created, and unit testing is conducted for each function by each student.

The group progress report is presented in turn by each group member. Progress checks during the lecture time, the inspection of artifacts created during the upstream phase, and acceptance testing of the application developed by each group are conducted by the teaching staff to provide feedback to the student groups.

As the development environment, the students use their own laptop computer. We use GitHub as the source code and document repository. We use version control in the

documents, the “Issues” function (the formal location of our text communications, including discussions and bug reporting, among other exchanges), and the “Pull Request” function for the review process for artifacts. The groups are allowed to use a variety of tools.

4 Our PBL in the 2020 academic year

All face-to-face learning activities were prohibited in the PBL course of the 2020 academic year.

The task in this year’s PBL software development course was based on a request from a professor at our university. It consisted of software development for an actual client, constituting a Web application for home economics education, where the students select items from the food menus stored in the system and the system calculates the adequate level of nutrition and provides advice to the students.

We used Microsoft Teams as the infrastructure of online remote lectures and applied the following functions provided by the application:

- Creation of a “team” (private)
- Creation of a “channel” as a place of communication per group under a “team”
- Text chat
- Voice communication
- Video communication
- Screen sharing
- Creation of folders and file uploading.

We also implemented a function to transfer the content of the Issues function in GitHub to Teams. GitHub was used in the same manner as prior to the pandemic.

We used Teams and GitHub for different purposes. Live lectures were delivered using the voice communication and screen-sharing functions of Teams. Discussions between the developers and the members of the Faculty of Home Economics, who were the clients, were held using Teams. By contrast, communications between the teaching staff and developers regarding the creation of artifacts based on text were typically exchanged on GitHub.

The progress meeting during the lecture hours required a longer time in the remote online environment than in a normal learning environment (10 min per group during the previous years versus 15 min per group during the 2020 academic year).

5 Result

Nine students, organized into two groups, participated in the course during the 2020 academic year. The PBL was conducted from October 23, 2020 to February 5, 2021. Both groups completed their project, presented their processes, and demonstrated their system on February 5. All the students completed the course.

Table 1 shows partial data on the developed system and the development process during the 2019 and 2020 academic years. As the table indicates, in 2020, the student groups produced more lines of code, Issues, and Pull Requests than in 2019. Because

the project for 2020 was the development of a system required by actual users, it was more complicated than those for past years. The table thus shows that the groups worked actively in a fully online remote environment. Figure 1 shows the partial communication history of a team in Microsoft Teams, and Figure 2 shows an example screen shot of the Issues function provided by GitHub.

Table 1. Some quantitative data regarding the process of groups in the 2019 academic year and the 2020 academic year.

Items	Group A in 2019	Group B in 2019	Group B in 2020	Group B in 2020
Lines of code	4591	7088	61036	9073
Number of Issues	117	77	203	187
Number of Pull Requests	228	207	250	254

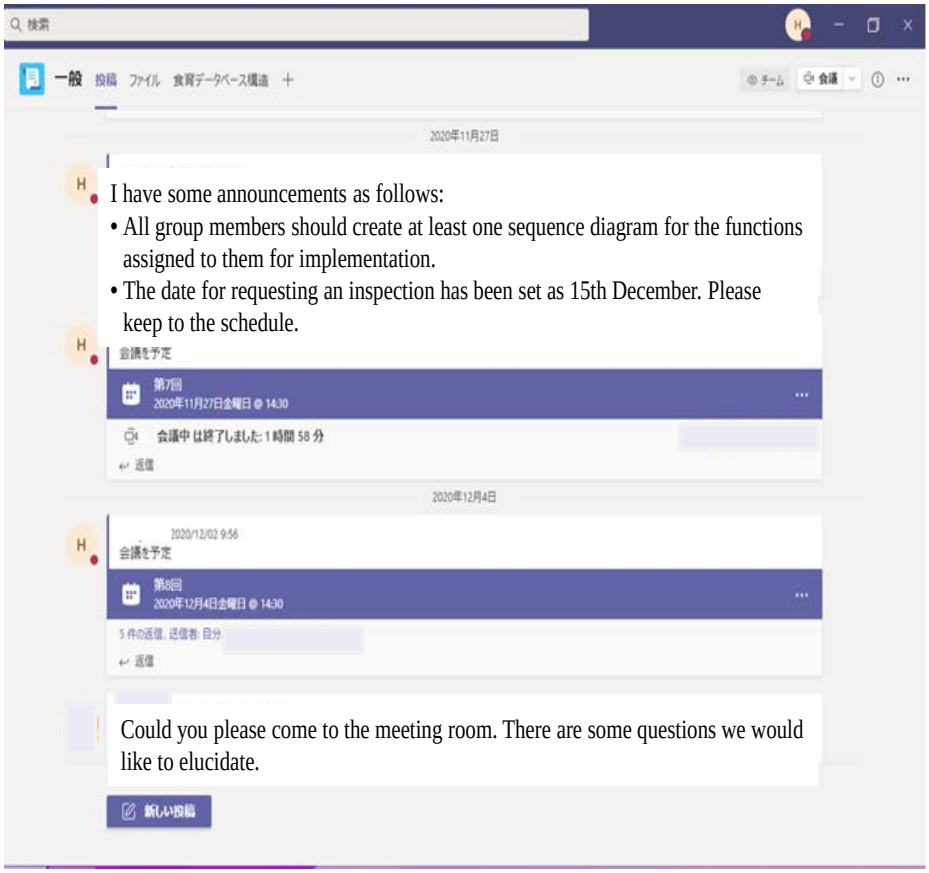

Fig. 1. Screen shot of Teams.

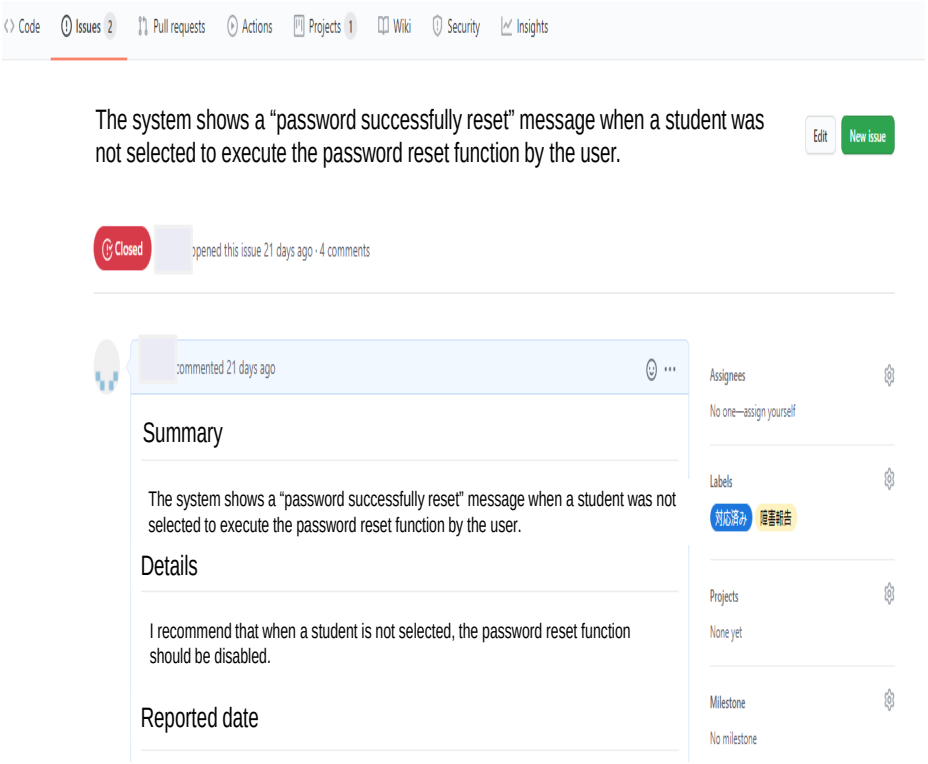

Fig. 2. Screen shot of the Issues function provided by GitHub.

6 Evaluation

We clarified the difficulties encountered by the students with less experience in conducting PBL style software development in a fully online remote environment. As a part of the final assignment, the instructor asked the students to describe the “difficulties they encountered during their remote learning activities and their solutions to these problems.”

As the results indicate, none of the students encountered any major difficulties. Some of the students’ comments are as follows.

The positive opinions regarding remote learning activities are as follows.

- I had no problem not meeting with other members face-to-face and saw the possibility of working remotely.
- I did not have any difficulty working remotely.
- It was easy to prepare for group meetings (no need to travel to class).
- No delay was caused by working remotely.
- I experienced no difficulties because we were able to communicate through social networking services.

The students also noted some of the difficulties they experienced:

- It was difficult to ascertain the progress of the other group members.
- I felt stress having text conversations because doing so takes hours and more time is required until a resolution is found.
- It was difficult for us to understand the requirements of the users. In addition, it was difficult to communicate my thoughts about the development to the other group members.
- It was difficult to fix my problems outside of class because the team members were not staying on campus.
- It was not easy to hold talks and/or chats (early chats may lead to group cohesion).
- I felt unmotivated because I was unable to view the environment surrounding the other members.

We asked the teaching assistants to answer the same question from their perspective and obtained the following opinions.

- I experienced no problems with remote PBL from either the student or the teaching staff side.
- I was able to ascertain the progress of the groups through the progress meeting held once a week during the lecture time.
- I could ascertain the artifacts in the upstream phase clearly because I participated in the inspection process as an inspector.
- Because the course does not introduce code inspection by the teaching staff, I experienced difficulty ascertaining the progress (this was also true during face-to-face PBL).

In general, the students who had less experience in software development conducted in a fully online remote PBL environment did not experience difficulties. Some students noted difficulties in ascertaining their progress, the extra effort needed to communicate, the difficulties that resulted from not being promptly supported when they needed help, and issues of motivation. These difficulties may arise even when this course is conducted in a collocated and not fully online remote environment because the group members do not meet frequently. Clearly, fully online remote environments make these difficulties tangible.

This evaluation shows that a project-based software development course can be conducted in an effective manner and without pauses in learning even when the students are unable to meet in a collocated manner. The evaluation results show that a learning environment in which GitHub is used in combination with a remote meeting system is effective in allowing groups to conduct software development activities.

7 Summary

This paper reported a practical implementation of PBL-style software development education during the COVID-19 pandemic. Most students in our PBL course are novice software developers. As one of the characteristics of this learning style, the teaching

staff provide feedback regarding the output of student groups, which is based on the results of the inspection process during the upstream phase, acceptance testing, and progress checking. Our PBL course was conducted before the COVID-19 pandemic, using GitHub for managing the artifacts and communication between the teaching staff and groups. Then, because of the COVID-19 pandemic, the weekly face-to-face lectures were also moved to a remote meeting system. No additional changes were made to the course during the pandemic. The PBL course of the 2020 academic year was successfully completed. The students experienced no serious difficulties regarding their fully online remote project-based software development course. In our opinion, the process for providing feedback, as outlined above, may be effective. In addition, we are also of the opinion that artifact management using GitHub and communication between the teaching staff and the groups using GitHub and Teams, which played an important role in this situation, may be effective in a fully online remote environment. COVID-19 has not yet been eradicated worldwide. Therefore, strong measures such as lockdowns may be required again in the future. Our approach may be effective in such situations.

In the development of modern software, such as open-source software, artifacts are created in an electronic format, communications are exchanged electronically, and the development is conducted in a distributed fashion. These characteristics are resilient to the conditions enforced by the COVID-19 pandemic. Our PBL has these characteristics. In addition, the proportion of activities that require face-to-face communication is low, and therefore, the transition to PBL can be achieved smoothly.

This paper described the interaction processes between the teaching staff and the student groups from the viewpoint of the teaching staff. We plan to further investigate the micro-processes within the student groups that occurred during the COVID-19 pandemic.

Acknowledgement

This study is partially supported by the Grant-in Aid for No. (C) 20K12089 from the Ministry of Education, Science, Sports and Culture of Japan. The authors thank Professor Michiko Minami and Professor Sakura Kushiyama and all the students who participated in this course for their contributions in the 2020 academic year.

References

1. Bao, W.: COVID-19 and online teaching in higher education: a case study of Peking University. Human Behavior and Emerging Technologies 2(2), 113-115 (2020).
2. Luburić, N., Slivka, J., Sladić, G., Milosavljević, G.: The challenges of migrating an active learning classroom online in a crisis. Computer Applications in Engineering Education, 1-25 (2021).
3. Vogel-Heuser, B., Bi, F., Land, K., Trunzer, E.: Transitions in teaching mechanical engineering during COVID-19 crisis. Interaction Design and Architecture(s) Journal, 27-47 (2020-2021).

4. Serhan, D.: Transitioning from face-to-face to remote learning: students' attitudes and perceptions of using Zoom during COVID-19 pandemic. International Journal of Technology in Education and Science, 4(4), 335-342 (2020).
5. Aristovnik, A., Keržič, D., Ravšelj, D., Tomaževič, N., Umek, L.: Impacts of the COVID-19 pandemic on life of higher education students: A global perspective. Sustainability, 12(20), 34 pages (2020).
6. Ghazi-Saidi, L., Criffield, A., Kracl, C. L., McKelvey, M., Obasi, S. N., Vu, P.: Moving from face-to-face to remote instruction in a higher education institution during a pandemic: multiple case studies. International Journal of Technology in Education and Science, 4(4), 370-383 (2020).
7. Rashid, C. A., Salih, H. A., Budur, T.: The role of online teaching tools on the perception of the students during the lockdown of Covid-19. International Journal of Social Sciences & Educational Studies, 7(3), 178-190 (2020).
8. Mahmood, S.: Instructional strategies for online teaching in COVID-19 pandemic. Human Behavior and Emerging Technologies, 199-203 (2020).
9. Hazeyama, A.: An education class on design and implementation of an information system in a university and its evaluation. In: Proceedings of the 24th Annual International Computer Software and Applications Conference (COMPSAC2000), pp. 21-27, IEEE (2000).
10. Miyashita, Y., Yamada, Y., Hashiura, H., and Hazeyama, A.: Design of the inspection process using the GitHub flow in project based learning for software engineering and its practice, arXiv:2002.02056, (2020).
11. Barr, M., Nabir, S. W. Somerville, D.: Online delivery of intensive software engineering education during the COVID-19 pandemic. In: Proceedings of the 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T2020), pp. 1-6, IEEE (2020).
12. Schmiedmayer, P., Reimer, L., Jovanović, M., Henze, D. and Jonas, S.: Transitioning to a large-scale distributed programming course. In: Proceedings of the 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T2020), pp. 1-6, IEEE (2020).
13. Kanij, T., Grundy, J.: Adapting teaching of a software engineering service course due to COVID-19. In: Proceedings of the 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T2020), pp. 1-6 IEEE (2020).
14. Mues, M. and Howar, F.: Teaching a project-based course at a safe distance: an experience report. In: Proceedings of the 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T2020), pp. 1-6, IEEE (2020).
15. Plewnia, C., Steffens, A., Wild, N., Lichter, H.: A lightweight collaborative approach for teaching software project labs with industry partners, In: Joint Proceedings of SEED & NLPaSE, CSUR-WS, Vol. 2799, pp. 1-8. http://ceur-ws.org/Vol-2799/ (2021).
16. Bringula, R., Geronimo, S., Aviles, A.: Programming project in an undergraduate software engineering in the new normal: challenges and proposed solutions. In: Joint Proceedings of SEED & NLPaSE, CSUR-WS, Vol. 2799, pp. 29-37. http://ceur-ws.org/Vol-2799/ (2021).
17. Motogna, S., Marcus, A., Molnar, A. J.: Adapting to online teaching in software engineering courses. In: Proceedings of the 2nd ACM SIGSOFT International Workshop on Education through Advanced Software Engineering and Artificial Intelligence, pp. 1-6, ACM, New York (2020).
18. Yamada, Y., Furukawa, K., and Hazeyama, A.: Conducting a fully online education of a software engineering course with a web application development component due to the COVID-19 pandemic and its evaluation. In: Joint Proceedings of SEED & NLPaSE, CSUR-WS, Vol. 2799, pp. 20-28. http://ceur-ws.org/Vol-2799/ (2021).