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

Yuki Yamada
Graduate School of Education.
Tokyo Gakugei University *

Keychi Furukawa
Graduate School of Education.
Tokyo Gakugei University *

Atsuo Hazeyama
Department of Information Science.
Tokyo Gakugei University *

Abstract

On account of the COVID-19 pandemic, a number of educational institutes have been forced to execute online distributed education. We have been providing an introductory software engineering course that conducts the practice of web application development in addition to lectures. Our goal was to provide students with a face-to-face lecture-like experience in a fully online distributed environment. We have redesigned the course to achieve this goal, with less preparation time (about two weeks). We have been holding an introductory software engineering education that conducts practice of web application development in addition to lectures. In traditional subjects, conducting lectures as well as practices, the instructor and teaching assistants walk around the class, check how the students are doing, identify and guide the students who have trouble with their practices. In distributed education, it is difficult to do accomplish these tasks; therefore, some measures are required. For this reason, we have developed tools to automate the building of a software development environment on students' PCs and to support questions and answers during web application development tasks. In this paper, we report on the practice of web application development in this course and the students' evaluations thereof.

Joint Proceedings of SEED & NLPaSE co-located with APSEC 2020, 01-Dec, 2020, Singapore
EMAIL: m198132f@st.u-gakugei.ac.jp(A. 1); am208125n@st.u-gakugei.ac.jp (A. 2); hazeyama@u-gakugei.ac.jp(A. 3)

1 Introduction

On account of the COVID-19 pandemic, several educational institutes have been forced to execute online distributed education. In Japan, a new school semester usually starts in April. In 2020, this start was delayed for about a month, and the semester started in May in our university. Two weeks before the start, our university announced that all courses must conduct fully remote online education. Our goal is to provide students with a face-to-face lecture-like experience in fully online distributed education. We have to redesign the course to achieve this goal, with less preparation time. Bourne et al. pointed out the difficulty of building an environment for practice, and coaching including communication, for online engineering education[1]. Some measures are required to address these difficulties. Thus far, we have provided an introductory software engineering education. This course conducts practice of web application development in addition to lectures given by an instructor. For practice, students are required to build their software engineering environment on their own personal computers (PC). After setting up their development environment, they create projects using the source program provided by the instructor and work on the assignments. The development environment consists of plural open source software. While building their development environment, the differences among students' PCs cause trouble. When building their software engineering environment and troubleshooting during application development, coaching involving communication between the teaching staff and students become important. The teaching staff provides support to students in the form of questions and answers. As such, in the fully online implementation of this course, we need to consider issues related to (1) building a software engineering environment on the students' PCs for practice and (2) troubleshooting when students encounter problems during application development tasks.

To solve these issues, we have developed tools to automate the construction of the development environment and to support students in troubleshooting during their web application development practice.

In this paper, we report on the practice of this lecture and the students' evaluation of the lecture.

2 Overview of the course

The target course of this study is called "Design of Information Systems." [1] The course is offered for third-year undergraduate students majoring in informatics education (full quota of fifteen students). The students learn the basics of software engineering as well as simple web application development using the JSP/Servlet technologies and relational database management systems. To that end, our department offers courses on programming, databases, and so on in the previous (first-fourth) semesters before the Design of Information Systems course (fifth semester).

Thirty-one students took the course in 2020, and two teaching assistants supported the course. Our university requires all students to own a PC (BYOD: Bring your own device). Students use their PCs to build the required software engineering environment for this course. The software engineering environment consists of the Java programming language, MySQL RDBMS, and the Gradle build tool. In the following, we provide an overview of the implementation of this course in a face-to-face fashion as it has been conducted until recently. Then, we provide a brief explanation of our remote online education environment. We discuss some issues in conducting application development practice in a fully remote environment. This course uses their PC to build their software engineering environment.

2.1 An overview of practice in the previous years

Before the 2019 academic year, students built the required software engineering environment for this course according to the instructions provided by the instructor during class time. When troubles occurred, the students would ask colleagues or the teaching assistants (TAs) to support them, in a face-to-face fashion. After building the environment, the students studied RDBMS MySQL. Then, they created some Gradle projects using the source programs provided by the instructor and confirmed their execution. At this point, students would get in-person support from colleagues and the TA, should they encounter any issues. However, in the 2020 academic year, face-to-face classes are not allowed because of the COVID-19 pandemic. The class is held once a week; however, it is held via an online meeting system.

[1] The syllabus of "Design of Information Systems",
http://tgusyl.u-gakugei.ac.jp/ext_syllabus/referenceDirect.do?nologin=on&subjectID=237100002507&formatCD=1

2.2 A brief explanation of our remote online education environment

In the current situation, we have to give lectures via a remote meeting system. We decided to use Microsoft Teams[2](hereafter, we use the term Teams) to give lectures in a synchronous manner, because it is a platform used by our university. The users of Teams can hold meetings in a unit called a "team." They can create a chatroom called a "channel" per topic within a team. Teams provides a chat function that allows users to communicate with those outside a team. In a meeting or a chat, participants/users can share their screens. The teacher creates a team for this course, whose members are students who apply to this course, the TAs, and himself. Lectures are given within this team by presenting learning materials through the screensharing feature and explaining them via a microphone. Questions during class times should be submitted via text chat in Teams. The teacher also accepts questions via the chat function outside the class time.

We also distribute programs and various instructions as well as materials for lectures via a learning management system (LMS), as we have done thus far.

2.3 Issues regarding conducting application development practice in a fully remote environment

When conducting remote practice, it is especially important for the students to build their software engineering environment and for the instructor and TAs to support students' troubleshooting tasks during practice.

In addition, as described in the previous section, questions from students in remote education are started in a text-based way. However, the students have difficulty in properly expressing the problems they encounter, in the form of text, to the teacher and TAs. This can be mitigated by providing a mechanism to gather information on the problems encountered by students and to inform the teacher and TAs. We collected the version information of the software they use and the error log data that represent impasses to ascertain the situations of the students.

We have developed tools to address the above-mentioned situations and introduced them in practice. We addressed two issues we described in Section 1 by these tools: (1) building a software engineering environment on the students' PCs for the practice and (2) support for troubleshooting when students encounter troubles in application development.

3 Related Work

This section describes the related studies from two viewpoints: online software engineering education, and online engineering education in response to the COVID-19 pandemic.

3.1 Online software engineering education

Ellis presented the implementation of distributed software engineering education [2]. Before its implementation, face-to-face sessions were conducted. The target students were professional students. However, we must prepare a fully distributed online educational environment for novice university students.

Bourne et al. described the future practices of online engineering education in their 2005 article [1] : "Learning Anywhere, Anytime." It differs from our online, live education.

Massively Open Online Courses(MOOCs) are conducted in a fully distributed online educational environment. While Fassbinder et al. state that "their (MOOCs') potential in Software Engineering Education remains underinvestigated," they consider how MOOCs for software engineering education should be designed and their challenges, in addition to the development of a case [4]. MOOCs users assume that learning environments are online. On the other hand, our students are experiencing a forced change in their learning environment, from a face-to-face to a synchronous and fully online fashion. Furthermore, we must redesign our course in a very short time.

3.2 Online education in response to the COVID-19 pandemic

We introduce some studies regarding the practices of online engineering education in response to the COVID-19 pandemic.

Suryaman et al. reported the results of their questionnaire with the goal to evaluate whether online education is properly conducted for students and faculty in civil engineering [5]. They concluded that it is important to provide learning materials so that students do not feel bored, and that easy access is necessary, since they do not always want to attend online education classes.

[2]Microsoft Teams, https://microsoft.com/teams/

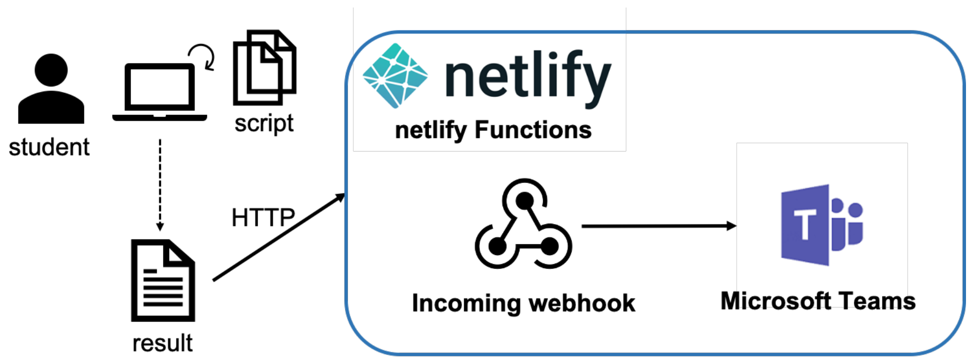

Figure 1: Overview of the support tool for development practice

Chen et al. investigated user satisfaction for some online education platforms and proposed improvements during the COVID-19 pandemic [3]. We need to evaluate the user satisfaction of our proposed platform.

4 A support tool for development practice

As described in Section 2.3, we have developed a tool to deal with the issues of (1) building a software engineering environment on the students' PCs for practice and (2) troubleshooting when students encounter troubles during application development. To deal with (1), we tried to automate the building of the software development environment. To address (2), we collected and checked the students' software and error log information and send a notification to Teams (an educational platform used in this course.) We have developed a tool to perform these tasks.

This tool consists of two scripts and one application that runs on Function as a Service (FaaS). One script is to build the software engineering environment, and the other is to support troubleshooting tasks during web application development. The application notifies the teaching staff of the results of the script. Figure 1 shows an overview of the support tool for development practice. When a student executes a script, the result is sent to the application running on the FaaS environment. Then, the application notifies Teams.

In the following subsection, we describe the details of the tool.

4.1 The application to notify the educational platform, Teams

We implemented a mechanism for the teaching staff to ascertain the results of the students' script executions, which sends a notification to Teams. The mechanism is implemented using Netlify[3] FaaS environment. FaaS is an event-driven service, and we can describe various processing by a web request with the HTTP communication protocol as a trigger.

When the application receives the web request including the text file that is returned from the script execution, it verifies the text file, extracts the contents from the text file, and notifies Teams using the Incoming Webhooks. Even when the Webhook request fails, the results are stored in a database to ensure that the confirmation of the execution results is not skipped. This mechanism allows the teaching staff to ascertain the status of the students on Teams.

4.2 The script to build the software engineering environment

The preparation of a software engineering environment for all students is crucial, because a slight difference in software engineering environment (e.g., differences in programming language versions) cause problems in application development. We suppose that the students use Windows or macOS on their PCs. To address the differences in the operating systems (OS) they use, we prepared a PowerShell script for Windows users and a bash script for macOS users. The script needs to install a set of software to be used in this course, namely Java, MySQL, and Gradle, on the students' PCs. The software was installed using package management software (e.g., Chocolatey[4], Scoop[5], and Homebrew[6]). If the package software was not already installed on the students' PCs,

[3]Netlify, https://www.netlify.com/
[4]Chocolatey, https://chocolatey.org/
[5]Scoop, https://scoop.sh/
[6]Homebrew, https://brew.sh/

Figure 2: A screenshot of executing the script to build a software engineering environment.

the script also installed the package management software.

The teaching staff must ascertain whether an installation on a student's computer succeeds or not through the execution of the script. The script checks for the occurrence of installation errors for each software, verifies the version of each software, and outputs these results in a text file. The script sends the text file to the application described in the previous section via the HTTP communication protocol so that the teaching staff can ascertain the installation status.

Figure 2 shows a screenshot of executing the script. The script requests the students to enter their ID to identify them in the log file. After entering the ID, the script is executed.

4.3 The script for acquiring application execution data

It is very important for students to be able to pinpoint the causes of errors encountered during web application development and find out relevant solutions by themselves. However, as they are novice developers, they may not know how to deal with these errors. Therefore, experienced TAs must support students in troubleshooting. To do so in an appropriate and prompt manner mitigates the students' burden. However, it is not easy for the TAs to confirm the status of the students in a fully remote environment as they could in a collocated classroom. Therefore, we propose a mechanism for the TAs to grasp the status of students so that they can support their troubleshooting activities. We have identified a requirement for supporting troubleshooting during application development practice as follows. The teaching staff should easily ascertain students' statuses during application development. To meet this requirement, we collected log data during the students' application development process. The students developed a web application in a project that introduced the Tomcat plugin of Gradle. We developed a mechanism by which the teaching staff could confirm error logs that a software engineering environment generates, to support students' troubleshooting activities. The script invokes Tomcat that executes an application. At this time, the script extracts and stores the logs generated by the Gradle logger in the form of a text file. The text file is sent to the application. This allows the teaching staff to see in Teams the errors that students encounter.

4.4 Execution method of the tool

The tool can be used by executing the script. It is necessary to devise an execution method for the script. An approach is to distribute the script via LMS-like learning materials, and the students can download and execute it. The execution of the script depends on the character set of the text file, and the script is written. Most LMS do not support the specifications of the character set of the text file. To prevent bugs caused by the character set, we did not use LMS for the distribution of the script. Instead, we adopted an execution method in which the script is allocated to GitHub, which is a source-code hosting service; each PC (each student) gets the script code and executes it. The method seems complicated; however, it is useful for both the students and the teaching staff. The students can simply invoke one-line commands without downloading the script on their PCs. Further, the teaching staff can not only attend to the problem of addressing various character sets but also save maintenance costs (incurred by bug fixes and version updates) by modifying only the source code on GitHub without having to re-distribute the script to all students.

Figure 3: Notification message by the script to build the software engineering environment

Table 1: Question items

No	Items of question
1	On building of a software engineering environment by the script
2	On the time taken to build a software engineering environment by the script
3	On the instruction for execution method of the script

After the execution of the script, the message shown in Figure 3 is sent to a Teams channel in which the teaching staff participate. This message allows us to grasp the installation status of the students.

5 Evaluation of the practice

Our evaluation involves our proposal to the two issues raised in Section 1: (1) the process of building a software development environment for course exercises using the script we developed and (2) the ease of asking questions in the remote implementation of the course compared to that in the face-to-face implementation.

(1) Using the script to build a software development environment to complete the exercises

We based our evaluations on the results of the script to build the required software engineering environment, as obtained by the teaching staff; and the results of an online questionnaire survey completed by the students.

A total of 26 students ran the script, and we confirmed they all succeeded in installing the necessary software. Further, we conducted an online questionnaire for the students to evaluate their experience, after installation. Table 1 shows the questionnaire items. We adopted a four-point Likert scale ("strongly satisfied," " satisfied," "unsatisfied," and "strongly unsatisfied") for all items. We received 19 responses to the questionnaire. Figure 4 shows the results for the questionnaire item 1, item 2 and item 3.

Figure 4 item 1 shows the results for the first questionnaire item. All responses are positive (13 are strongly satisfied, and 6 are satisfied). The students expressed high satisfaction regarding their experience of building a software engineering environment using the script.

Figure 4 item 2 shows the results for the second questionnaire item. All responses are positive (10 are strongly satisfied, and 9 are satisfied). These results show that the time required to build the environment by using the script is not costly for the students.

Figure 4 item 3 shows the results for the third questionnaire item. All responses are positive (15 are strongly satisfied, and 4 are satisfied). The results reveal that students were particularly satisfied with the instructions for the execution of the script. This shows that our approach is not difficult for students.

(2) Ease of asking questions in the remote versus the face-to-face implementation

After the students completed the course, they responded to a questionnaire on the ease of asking questions in the remote versus face-to-face implementation of the course. The questionnaire items are shown in Table 2. The

Table 2: Questionnaire on ease of asking questions in the remote versus face-to-face implementation of the course

No	Items of question
1	Ease of questioning in the class time
2	The reason why item 1 is evaluated
3	Ease of questioning outside the class time
4	The reason why item 3 is evaluated

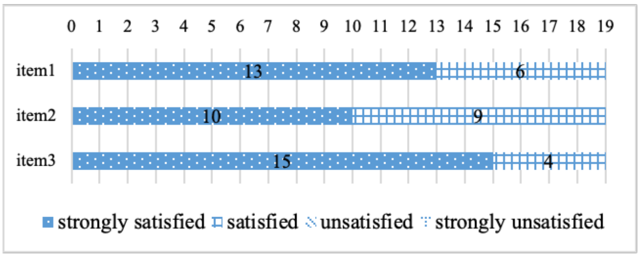

Figure 4: Results for questionnaire item 1, item 2 and item 3

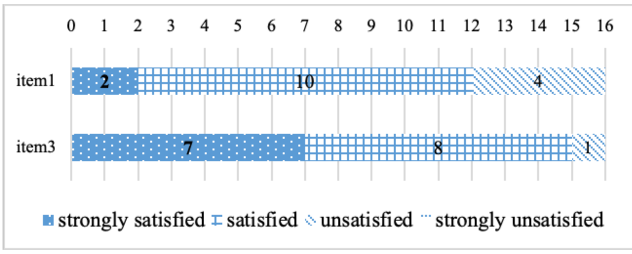

Figure 5: Results for questionnaire item 1 and item 3

first and third items were scored on a four-point Likert scale ("strongly satisfied," " satisfied," "unsatisfied," and "strongly unsatisfied"). The responses to other questionnaire items comprised free descriptions. We obtained 16 responses.

Figure 5 shows the results for questionnaire item 1, and item 3. As Figure 5 item 1 shows, most students are satisfied with the ease of asking questions during class. However, some students feel unsatisfied. The reasons for that were reported as follows: "It was difficult for me to ask a question, because all participants could hear my utterances in the team," "The instructor did not provide enough time for questions and answers," "It is difficult for me to ascertain the atmosphere around me in a remote meeting system. I think it is only me who does not understand the course contents," and "It is difficult for me to convey the contents of my questions in text chat." On the other hand, as Figure 5 item 3 shows, most students are more satisfied with the ease of asking questions outside the class time than that during class. The reason for such dissatisfaction is reported as "I feel that I have only a few opportunities for asking questions." Many students pointed out the effectiveness of the chat function.

6 Discussions

We discuss whether our tool contributes to supporting students' in building the software engineering environment on their PCs and their troubleshooting the encountered errors, that is, ensuring that the students successfully build the environment and monitoring students' application development practices.

6.1 Effectiveness of the tools

All participating students who executed the script succeeded in building the software engineering environment. Their degree of satisfaction with the process was very high. The teaching staff could ascertain the results of students' executions on Teams and ascertain the students' situations to help them to troubleshoot in a remote environment.

The acquisition script for application execution data was executed once. We suppose this script was used to support troubleshooting the errors that the students encountered during class (the TAs ascertained the situations of the students encountering trouble). However, in this practice, the students asked for help outside the class time, and the TAs used the phone call and screensharing functions provided by Teams.

We will reevaluate the tool under the supposed environment in future work.

6.2 Ease of communication during troubleshooting

The questionnaire results (Figure 5) reveal that the degree of satisfaction for the ease of asking questions in class time varies. Some students had no issues, while others found text-based communication to be problematic. Almost all students felt satisfied outside the class time, because they used the one-to-one chat function provided by Teams. They felt satisfied especially with phone calls and screensharing. The questions caused burdens for the teaching staff.

Table 3: Submissions of web application development tasks

Year	No. of students who take the final examination	No. of students who submitted a task of web application	Rate of submission (%)
2020	27	23	85
2019	24	22	92
2018	23	22	96

6.3 Learning effectiveness of our remote education practice

It is important to discuss the effectiveness of remote education. This paper discusses learning effectiveness from the viewpoint of the submission of web application tasks and the passing rate of the course.

Table 3 shows the submissions of web application development tasks during the past three years, including this year. This year's rate of submission was 85%, the lowest in the past three years. We think the reasons are that students are not allowed to learn with their friends on campus and that they have barriers to asking questions to the TAs online.

On the other hand, the passing rate of the course is determined by the mark of the final examination and the quality of the submitted web application tasks (documents and the application itself). It was 67% this year, almost the same as in the past two years (the average rate of the past two years was 62%). This means that the outcomes of remote education are not lower than those of face-to-face education.

In light of these results, we believe that the learning effectiveness in remote education is not lower than that in a collocated environment.

7 Conclusions

This paper has proposed a fully online implementation of introductory software engineering education, including web application development practice. To implement this course, we have developed a tool to automatically build a software engineering environment and obtain information on the status of the installation operation on students' PCs. We have also developed a mechanism to support students in troubleshooting their web application development issues. Although the technology that the script uses is not novel, this proposal could not have been implemented five years ago, because the required technology was not available at the time. However, currently, it is feasible thanks to the availability of Teams (including Webhook) and FaaS. Our approach is cost-effective and generally applicable to similar courses.

We applied the tools in an actual course. All students who executed the script succeeded in building the required software engineering environment, and their degree of satisfaction with the process was very high. However, the acquisition script for application execution data was rarely used. The students asked questions to the TAs outside the class time, and they preferred to communicate using phone calls and screensharing through the one-to-one chat function provided by Teams.

As for learning effectiveness, it is not lower in remote education than in a collocated environment.

We will conduct a longitudinal study to show effectiveness of our endeavor.

Acknowledgments

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.

References

[1] John Bourne, Dale Harris, and Frank Mayadas, Online Engineering Education: Learning Anywhere, Anytime, Journal of Engineering Education, Vol. 94, No. 1, pp. 131-146, 2005.

[2] Heidi J. C. Ellis, An evaluation of learning in an online project-based web application design and development course, Journal of Computing Sciences in Colleges, 21(6), pp. 217-227, 2006.

[3] T. Chen, L. Peng, X. Yin, J. Rong, J. Yang, and G. Cong, Analysis of User Satisfaction with Online Education Platforms in China during the COVID-19 Pandemic, Healthcare, Vol. 8, No. 3, 26 pages, Multidisciplinary Digital Publishing Institute, 2020.

[4] Aracele Garcia de Oliveira Fassbinder, Marcelo Fassbinder, Ellen Francine Barbosa, and George D. Magoulas, Massive open online courses in software engineering education, Proceedings of the 2017 IEEE Frontiers in Education Conference (FIE2017), pp. 1-9, IEEE, 2017.

[5] Heri Suryaman, Kusnan, and Husni Mubarok, Profile of Online Learning in Building Engineering Education Study Program During the COVID-19 Pandemic, International Journal of Recent Educational Education (IJORER), Vol. 1, No. 2, pp. 63-77, July 2020.