2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) | 978-1-7281-6942-2/20/$31.00 ©2020 IEEE | DOI: 10.1109/TALE48869.2020.9368334

Fully Online Implementation of Introductory Software Engineering Education Including Web Application Development Practice

Yuki Yamada
Graduate School of Education
Tokyo Gakugei University
Tokyo, JAPAN
m198132f@st.u-gakugei.ac.jp

Kiichi Furukawa
Graduate School of Education
Tokyo Gakugei University
Tokyo, JAPAN
m208125n@st.u-gakugei.ac.jp

Atsuo Hazeyama
Department of Information Science
Tokyo Gakugei University
Tokyo, JAPAN
hazeyama@u-gakugei.ac.jp

Abstract—**Owing to the COVID-19 pandemic, a number of educational institutes have been forced to execute online distributed education. In traditional courses conducting not only lectures but also practice, the instructor and teaching assistants walk around the class, check how the students are doing, find students who are having trouble, and guide them. However, in distributed education, it is difficult to do the same things as in a collocated environment; therefore, some measures are required. We have been conducting introductory software engineering education that applies web application development in addition to lectures. In the fully online implementation of this course, the following issues need to be considered: (1) the creation of a mechanism to mitigate problems regarding the building of a software engineering environment, and (2) the creation of a question and answer process during application development on a remote meeting system. This paper describes some solutions to the two issues.**

Keywords—COVID-19, fully online software engineering education, building of a software engineering environment, web application development, question and answer

I. Introduction

Owing to the COVID-19 pandemic, a number of educational institutes have been forced to execute online distributed education. Both students and faculty have been required to shift to online education in an emergent way and within an extremely short time period. Bao proposed six instructional strategies for online education by the COVID-19 pandemic [1]. In traditional courses conducting not only lectures but also actual practice, the instructor and teaching assistants walk around the class, check how the students are doing, find students who are having difficulties, and guide them. However, in distributed education, it is difficult to do the same things as in a collocated environment, and certain measures are therefore required.

We have been conducting an introductory software engineering education course. This course applies the practice of web application development in addition to lectures by an instructor. In the practice, students need to build their software engineering environment on their own personal computer (PC) and create projects using source programs provided by the instructor and confirm the execution. They then tackle an assignment. The development environment consists of plural open-source software. Some individual differences in the PCs used by the students can cause problems in terms of building their development environment. Bourne et al. pointed out the difficulty of building an environment for practice and coaching, including communication in online engineering education [2].

In the fully online implementation of this course, we need to consider the following two issues: (1) the creation of a mechanism to mitigate problems regarding the building of a software engineering environment, and (2) the creation of a question and answer process during the application development practice under fully distributed circumstances on a remote meeting system. This paper describes the solutions to the two issues mentioned above.

The remainder of this paper is organized as follows. Section II describes related work of this study. Section III gives an overview of our course. Section IV describes our approach we propose in this paper. Section V describes the current status of this course. Section VI concludes this paper.

II. Related Work

Ellis presented the implementation of distributed software engineering education [3]. Before the implementation of distributed education, a face-to-face session was conducted. The target students were professionals. We had to prepare a fully distributed online live educational environment (not an on-demand style) for novice university students.

Bourne et al. described the future practice of online engineering education in 2005 [2]. Their online education is a “learn anywhere, anytime” approach, which differs from our live online education method.

Massively open online courses (MOOCs) are a fully distributed online educational environment. Although Fassbinder et al. stated that the potential of software engineering education MOOCs remains under investigation, they considered how such MOOCs should be designed along with their challenges and development [4]. However, MOOCs are generally prerecorded and not live. Our course aims at live online education, similar to a collocated face-to-face approach.

III. Overview of the Course

The target course of this study is called “Design of Information Systems” for third-year undergraduate students majoring in informatics education. The students learn the basics of software engineering as well as simple web application development using JSP/Servlet technologies and relational database management systems. 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). Although the students have finished studying the constituent technologies used in software development, such as programming and database development, almost all of them are novices in software development at the university level. A total of 31 students took this course during the 2020 academic year, which was supported by two teaching assistants (TAs).

Our university requires all students to own their own PCs. This course requires the students to use their PCs to build their software engineering environment. The software engineering environment consists of Java programming language, an integrated environment of Eclipse, MySQL RDBMS, and the Gradle build tool. Such an environment is generally used in Web application development. However, differences in versions regarding the OS and software installed on the students' PCs need to be taken into consideration.

Prior to the 2020 academic year, the students built their own software engineering environment according to the instructions provided by the instructor during class time. When problems occurred, the students asked their fellow students or the TAs to support them. After building their environment, they studied RDBMS MySQL. They then created some Gradle projects on Eclipse using source programs provided by the instructor and confirmed their execution (the application includes a registration function and a browsing function, and the total number of lines of code is approximately 450). At this time, the students also asked their fellow students or the TAs to support them when encountering difficulties. However, in the 2020 academic year, face-to-face classes were not allowed because of the COVID-19 pandemic. Class was held once a week, but only through an online meeting system.

IV. Preparation toward Full Implementation

As mentioned in the previous section, it is necessary for all students to prepare their own software engineering environment. Because the troubleshooting required by the students when building their software engineering environment on their PC will likely be difficult for the TAs to handle through a remote meeting system, to implement the course design in a fully remote environment, as mentioned in Section III, we require two areas of preparation. The first is to support the building of a software engineering environment for all students. The second is to create a mechanism to support troubleshooting by the students during their web application development.

Under the current situation, we have to give lectures through a remote meeting system. We decided to use Microsoft Teams[i] (hereafter, referred to as Teams) to give lectures in a synchronous manner because this is the platform used by our university. In addition, we distributed programs and various instructions as well as materials for lectures through a learning management system (LMS), as previously used.

In the following subsection, we describe our preparations.

A. *Automatic Creation of Students' Software Engineering Environment*

The preparation of a software engineering environment for all students is an extremely important issue. This is because a delicate difference between a software engineering environment (such as differences in the version of the programming language and/or OS used) causes problems in application development.

Until the 2019 academic year, we had asked the students to build a software engineering environment by themselves in the classroom. Despite the learning effectiveness, some students faced problems in building their software engineering environment. A fully remote learning environment has made the situation worse. Building a software engineering environment completely on their own is not the main goal of this course. Therefore, we decided to prepare a script to allow automatically building a software engineering environment and to grasp the status of the students.

1) Requirements and design of the software engineering environment building script

We supposed that the students would use Windows or macOS on their PC. To address the differences in OS used, we prepared a PowerShell script for Windows users and a bash script for macOS users.

The script needs a set of software to be installed on the students' PC, namely, Java, MySQL, and Gradle. Such software programs were installed using package management software (e.g., Chocolaty , Scoop , and Homebrew). When the package software was not already installed on the PCs of the students, the script installed the software automatically.

To conduct remote education, the instructor and TAs need to ascertain the status regarding whether the installation by the students is successful based on the script execution. The script checks the occurrence of installation errors for each software, verifies the software version, and outputs the results in a text file. The script sends the text file to Teams through the HTTP communication protocol, and thus the instructor and TAs can ascertain the status regarding whether an installation by a student is successful. We describe the mechanism in the next subsection.

2) A method for ascertaining the building status of a software engineering environment by the students

We implemented a mechanism for the instructor and TAs to ascertain the status of building a software engineering environment by the students by notifying the status of Teams. The mechanism is implemented using Netlify Functions, which is a Function-as-a-Service (FaaS) environment provided by Netlify[ii]. FaaS is an event-driven service, and we can describe various types of processing using a web request with the HTTP communication protocol as a trigger. Netlify was adopted as a platform for the COVID-19 Task Force website of Tokyo[iii]. Therefore, our selection was valid.

We will now present the processing using the Netlify functions in more detail. When the script receives the web request including the text file described in Section IV.A.1, it verifies the text file, extracts the contents from the text file, and notifies the status of Teams using the Incoming Webhook provided by Teams.

Even when a Webhook request fails, the results are stored in a database such that confirmation of the execution results is not skipped. Figure 1 shows workflow of the software engineering environment building script. This mechanism allows the instructor and TAs to ascertain the building status of the software engineering environment by the students using Teams.

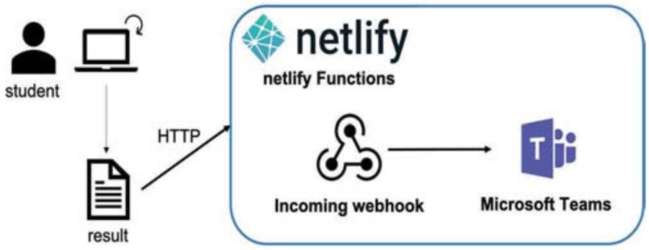
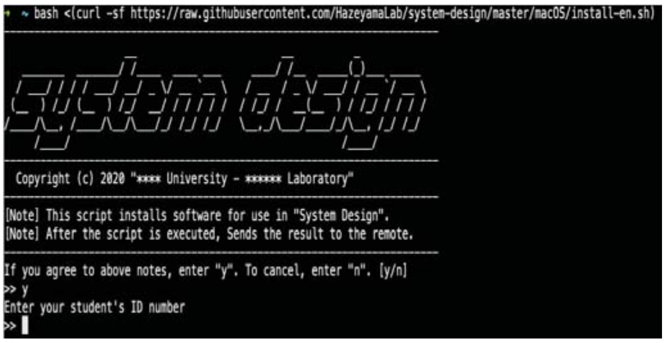


Fig. 1. Workflow of the software engineering environment building script.

3) Execution method of software engineering environment building script

It is necessary to devise an execution method for the script. One approach is to distribute the script through LMS like learning materials, which the students download and execute. The execution of the script depends on the character set of the text file. Most LMSs do not support the specifications of the character set of the text file. Therefore, we do not use an LMS for distribution of the script but adopt a method of execution in which the software engineering environment building script is allocated on GitHub[iv], which is a source code hosting service allowing each student to obtain the script code and execute it on their PC. As a benefit of this method, the students do not need to download the script on their PC and can execute the script regardless of the differences in the character set. The method seems to be complicated, although both the students and the teaching staff (the instructor and the TAs) have certain advantages. The students can invoke only one line command without any downloads. The teaching staff can not only address the problem of addressing various character sets but also save maintenance costs (bug fixes and updates of future versions) by modifying only a source code on GitHub without a re-distribution of the script for all students.

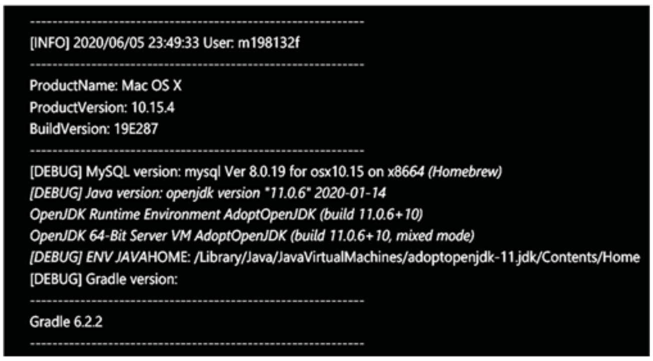
Figure 2 shows a screenshot of executing the script. The script requests the users to enter their ID to identify users in the log file. After entering the ID, the script described in Section IV.A 1 is executed. After the execution of the script, the message shown in Figure 3 is notified of the Teams channels of the instructor and TAs. This message allows the instructor and TAs determining the state of installation of each student.



```
→ ~ bash <(curl -sf https://raw.githubusercontent.com/HazeyamaLab/system-design/master/macOS/install-en.sh)
------------------------------------------------------------------------

------------------------------------------------------------------------
 Copyright (c) 2020 "**** University - ****** Laboratory"
------------------------------------------------------------------------
[Note] This script installs software for use in "System Design".
[Note] After the script is executed, Sends the result to the remote.
------------------------------------------------------------------------
If you agree to above notes, enter "y". To cancel, enter "n". [y/n]
>> y
Enter your student's ID number
>> ▌
```

Fig. 2. Screen shot of executing the script to build a software engineering environment.



```
------------------------------------------------------------
[INFO] 2020/06/05 23:49:33 User: m198132f
------------------------------------------------------------
ProductName: Mac OS X
ProductVersion: 10.15.4
BuildVersion: 19E287
------------------------------------------------------------
[DEBUG] MySQL version: mysql Ver 8.0.19 for osx10.15 on x8664 (Homebrew)
[DEBUG] Java version: openjdk version "11.0.6" 2020-01-14
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.6+10)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.6+10, mixed mode)
[DEBUG] ENV JAVAHOME: /Library/Java/JavaVirtualMachines/adoptopenjdk-11.jdk/Contents/Home
[DEBUG] Gradle version:
------------------------------------------------------------
Gradle 6.2.2
------------------------------------------------------------
```

Fig. 3. Notication message by the script to build the software engineering environment.

B. *A Mechanism to Support Troubleshooting during Web Application Development*

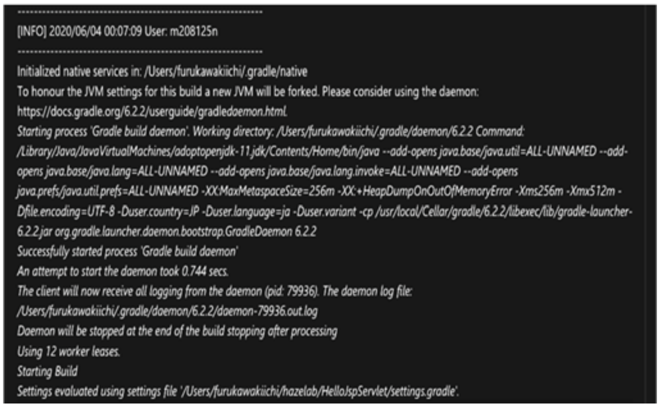
Allowing the students to pinpoint the reasons for errors encountered during web application development and find solutions to such errors on their own is extremely important. However, because they are novice developers, they may not know how to deal with these errors. Therefore, it is important for experienced TAs to support the troubleshooting of the students. Doing so in an appropriate and prompt manner mitigates the students' burden. However, it is not as easy for the TAs to confirm the status of the students in a fully remote environment as it is in a collocated classroom. We propose a mechanism to grasp the situation of the students and allow the TAs to support troubleshooting of the students using the technique we mentioned in Section IV.A.

1) Requirements and design of the application execution data acquisition script

The students develop a web application in a project introducing the Tomcat plugin of Gradle. We developed a mechanism by which TAs and instructors can confirm error logs that a software engineering environment generates to support troubleshooting for the students. The script for this invokes Tomcat, which executes an application. At this time, the script extracts and stores the logs generated by Gradle logger. The script also sends the logs to the remote Teams. The method is the same as that mentioned in Section IV.A.

2) Execution method of the application execution data acquisition script

We adopted a method of execution in which the application execution data acquisition script is allocated to GitHub, as mentioned in Section IV.A.3, allowing the students to obtain the script code and execute it on their PC. After execution of the script, the message shown in Figure 4 is sent to a channel of Teams where the instructor and TAs participate. This message allows the instructor and TAs to determine the execution of an application.



```
------------------------------------------------------------------
[INFO] 2020/06/04 00:07:09 User: m208125n
------------------------------------------------------------------
Initialized native services in: /Users/furukawakiichi/.gradle/native
To honour the JVM settings for this build a new JVM will be forked. Please consider using the daemon:
https://docs.gradle.org/6.2.2/userguide/gradledaemon.html.
Starting process 'Gradle build daemon'. Working directory: /Users/furukawakiichi/.gradle/daemon/6.2.2 Command:
/Library/Java/JavaVirtualMachines/adoptopenjdk-11.jdk/Contents/Home/bin/java --add-opens java.base/java.util=ALL-UNNAMED --add-
opens java.base/java.lang=ALL-UNNAMED --add-opens java.base/java.lang.invoke=ALL-UNNAMED --add-opens
java.prefs/java.util.prefs=ALL-UNNAMED -XX:MaxMetaspaceSize=256m -XX:+HeapDumpOnOutOfMemoryError -Xms256m -Xmx512m -
Dfile.encoding=UTF-8 -Duser.country=JP -Duser.language=ja -Duser.variant -cp /usr/local/Cellar/gradle/6.2.2/libexec/lib/gradle-launcher-
6.2.2.jar org.gradle.launcher.daemon.bootstrap.GradleDaemon 6.2.2
Successfully started process 'Gradle build daemon'
An attempt to start the daemon took 0.744 secs.
The client will now receive all logging from the daemon (pid: 79936). The daemon log file:
/Users/furukawakiichi/.gradle/daemon/6.2.2/daemon-79936.out.log
Daemon will be stopped at the end of the build stopping after processing
Using 12 worker leases.
Starting Build
Settings evaluated using settings file '/Users/furukawakiichi/hazelab/HelloJspServlet/settings.gradle'.
```

Fig. 4. Notification message by the application execution data acquisition script sent to the Teams.

V. Current Course Status

This course started on May 9, 2020. Based on a request of our university to avoid a system interruption through an overload of the communication channel, the first lecture was given after a lecture slide was uploaded onto the LMS by the instructor and the students had downloaded and studied the material by themselves. From the second class, we gave a live lecture using Teams, and to reduce the data size as requested we presented a lecture slide through a screen sharing function along with a vocal description. From the third lecture, we introduced a discussion as a face-to-face classroom. At this time, there were no problems in class operation or learning by the students. The two authors of this paper, excluding the developer, tested and confirmed both scripts. They successfully built their software engineering environment and confirmed the creation of a web application.

After the preparation, the instructor gave the students an instruction to build their own software engineering environment using the software engineering environment building script on June 5, 2020. On June 12, 2020, 15 out of 31 students executed the software engineering environment building script, and we confirmed that no problems had occurred.

VI. Conclusions

In this paper, a fully online implementation of an introductory software engineering course was proposed, including web application development practice. To implement this course, we developed a script to automatically build a software engineering environment and obtain the status information of the install operation. We also developed a mechanism to support troubleshooting by students during their web application development. We believe these tools will enable two issues, described in Section I, to be solved, that is, (1) the creation of a mechanism to mitigate issues regarding the building of a software engineering environment and (2) the creation of a question and answer process during the application development practice under a fully distribution of a remote meeting system.

This method used by the script is not novel. We can consider another advanced method that utilizes a cloud computing environment and prepares a virtual machine with a software engineering environment pre-built for the students. However, this method is too expensive, and the students have to study domain knowledge regarding cloud computing. The COVID-19 pandemic was an unexpected event, forcing us to prepare for a fully online implementation within an extremely short time period. In addition, although the method used by the script is not novel, such a proposal was probably not implementable five years ago. It is extremely practical and reasonable to do so now because the time has come to provide Teams (including Webhook) and FaaS. Our approach is extremely inexpensive, and similar courses are generally applicable.

We had not yet completed the application of the proposed method at the writing of this paper, and hope to report on the results such as satisfaction aspect of the students at a later date.

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.

References

[1] W. Bao, "COVID-19 and online teaching in higher education: A case study of Peking University," Human Behavior and Emerging Technologies, Vol. 2, No. 2, pp. 113-115, 2020.

[2] J. Bourne, D. Harris, and F. Mayadas, "Online engineering education: learning anywhere, anytime," Journal of Engineering Education, Vol. 94, No. 1, pp. 131-146, 2005.

[3] H. J. C. Ellis, "An evaluation of learning in an online project-based web application design and development course," Journal of Computing Sciences in Colleges, Vol. 21, No. 6, pp. 217-227, 2006.

[4] A. G. de O. Fassbinder, M. Fassbinder, E. F. Barbosa, and G. 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.

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

[ii] Netlify, https://www.netlify.com/

[iii] Tokyo COVID-19 Task Force website, https://stopcovid19.metro.tokyo.lg.jp/

[iv] GitHub, https://github.com/