

Analysis of the Transition to a Virtual Learning Semester in a College Software Testing Course

Chloe Smith
University of Virginia
Charlottesville, Virginia, United States
cas8ds@virginia.edu

Upsorn Praphamontripong
University of Virginia
Charlottesville, Virginia, United States
upsorn@virginia.edu

ABSTRACT

This report presents observations from two semesters of the same course, Software Testing, within the same department, in different school years. The first semester, Fall 2019, was the last semester completely in-person before the COVID-19 pandemic made it so that our courses changed to a virtual setting. The second semester, Spring 2021, is the most recent virtual semester at the time that this report is being written (in May of 2021). Though the course was modified to some extent for the purpose of it being online, we had a goal for it to remain largely the same. In both versions of the course, there was a heavy emphasis on interactive exercises as a means to increase engagement and performance within the class and in regards to software testing as a whole. The virtual semester gave us an opportunity to try out a variety of new avenues of learning and an exercise in trying to keep consistency from an in-person environment to a virtual one. This report will address added benefits of the virtual settings, consistencies that we managed to maintain throughout the transition, and lost exercises from the course along with ideas of how to mitigate the loss in any future virtual semesters. It will also address feedback from a survey given to students as we consider the results of the semesters.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **Human-centered computing** → Collaborative and social computing design and evaluation methods; • **Applied computing** → *Computer-assisted instruction*; *Interactive learning environments*; Collaborative learning; **Distance learning**; **E-learning**.

KEYWORDS

Virtual Learning, Software Testing, Adapting to Online Setting

ACM Reference Format:

Chloe Smith and Upsorn Praphamontripong. 2021. Analysis of the Transition to a Virtual Learning Semester in a College Software Testing Course. In *Proceedings of the 3rd International Workshop on Education through Advanced Software Engineering and Artificial Intelligence (EASEAI '21)*, August 23, 2021, Athens, Greece. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3472673.3473967>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
EASEAI '21, August 23, 2021, Athens, Greece

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8624-1/21/08...\$15.00
<https://doi.org/10.1145/3472673.3473967>

1 INTRODUCTION

Software testing is a critical step to expose information about the quality of one's code [1]. It aids in ensuring one's code is effective, safe, free from grave errors, conformant, compliant, and low-cost when it comes to future technical support. Despite this, software testing is often overlooked or neglected [2, 3, 6]. Many students within our department dislike testing their software and oftentimes consider software testing as simply a means to debugging a code until some threshold for a course grade is met. We have seen cases of testing merely through print statements, omission of testing as a whole, and very limited testing in a variety of courses. We have even seen situations where the test cases were changed rather than the code itself for the sake of the code "passing" the tests. This, of course, leads to low quality code that would be very expensive to maintain and may cause software failures in the future. We desire for the students in our department to gain a more robust view of software testing standards while here, so that as they enter more real-world coding scenarios, they may produce more effective and reliable code.

The Software Testing course offered within our department aims to further the testing capabilities and desires of our students with an emphasis on interactive learning activities [11]. As we have spent several semesters evaluating different methods of presenting Software Testing concepts and their effects on student interest and performance, we stumbled into a unique period of virtual learning due to the COVID-19 pandemic. As classes move online, not only do we need to maintain the course coverage, but we also need to ensure equity and inclusion in our online classroom settings [7]. Several studies and approaches have been introduced in response to an immediate shift of an online learning environment [4, 5, 12]. In our Software Testing course, we adopted several online platforms such as Zoom and Google Cloud, and developed a collection of digital materials to provide equitable and inclusive access to learning resources (including human resources and digital learning materials). We largely maintained our "hands-on" focus within the class, omitting only three out of thirty-three exercises due to the in-person nature of those three activities.

In this paper, we reflect on the way the two semesters compared in activities and their implementations. We discuss the execution of the course in these two semesters and the adjustment due to the virtual learning environment, describe the results of these executions (via observation within the class and surveys given before and after the Software Testing course in the virtual semester, Spring 2021), and suggest some preliminary insights into how this course may adapt whether in person or virtual in the future.

2 ADDED ABILITIES FROM VIRTUAL SETTING

2.1 How Are You Feeling?

Since our course was hosted by Zoom, we utilized the Zoom share-screen, annotation, and chat features allowing students to interact with a "How are you feeling?" web page [9]. The main purpose was to promote student well-being during the pandemic, and to check on the emotions and events of the students each day [?]. Students were encouraged to address anything that they desired to before the content for the day began. They were able to draw on the board using the "annotate" feature on Zoom, circling emojis that matched their emotions that day and drawing/writing about their days up to the start of class. This made it easier for the instructor to read the students without being in the same room as them. The shared screen was simpler than having students walk up to a whiteboard in person, and the nature of the Zoom share screen allowed students to remain largely anonymous if desired. We feel as though this created a safe outlet for students to address their state day-to-day; and hence promoted a welcoming and inclusive learning environment. There were even times when a student would direct message personal hardships to the instructor during this time. This gave the instructor a chance to intervene and follow up with an email after class in support of the student in the midst of their difficulties.

If this were translated to an in-person class, it would likely still involve technology in the form of a survey or website used before class begins.

2.2 Guest Lecture

Our virtual course had a guest speaker, Kinga Dobolyi (Ph.D., Data Scientist for In-Q-Tel Labs). Having a guest speaker in this class is not atypical for Software Testing, even during in-person semesters, since we desire to show students a window into the Software Testing field throughout the course, and guests are a practical way to demonstrate software testing as a career. Although guest lectures are, of course, possible in person, the virtual setting makes it much easier to actualize, as transportation costs are taken away. Dobolyi was able to speak to our class from Arlington. We see a lot of potential in this for future semesters that may be virtual (or even in person), as we can host guest speakers who are unable to travel to our campus. As we have felt the shift to a "new normal" from COVID-19, virtual guest lectures seem to be a positive change that opens a lot of doors for who we are able to host in this capacity.

2.3 Class Recordings

This virtual semester began a standard of taking class recordings that were posted to a class web page. These videos could be watched back at anytime. We heard from some students that the recordings were very helpful for review. Students were able to revisit parts of the lecture that may have confused them at first. Students who live in different time zones or overseas were able to view the lecture, making it possible for us to provide equal access to the course lectures and materials. These videos were useful also for students who had a valid reason to miss class, as they could see exactly what they had missed, rather than piecing together what they missed

from write-ups and/or the instructor repeating an appended version of the lecture in office hours.

2.4 Office Hours

Our Office Hours, like our lectures, were located on Zoom. This added a number of benefits to our office hours. First, both the student and the Teaching Assistant (TA)/instructor were able to see the same content on their respective screens, rather than hunching over the same computer in person. This made it easier for both parties to focus on what the student wanted to look at. Second, the annotate feature on Zoom came in handy once again, as TAs/instructor were able to write and/or draw helpful supplements on the screen of the student. Third, the TAs/instructor were able to share a whiteboard screen in the case that they wanted to draw a diagram to aid the student. This is possible with whiteboards or papers in person, but we found that this Zoom share-screen version felt more seamless and efficient in many cases. Lastly, similar to the case of guest lectures, the Zoom office hours made it much easier for students to attend office hours, even if they only had free time less than the hypothetical amount of time that it would have taken to walk from their prior engagement to the office hours location and then from the office hours location to their next engagement during an in-person semester. We noticed an increase in office hours attendance this semester, and we believe that the increase was due to the added convenience of hosting office hours on Zoom.

3 CONSISTENCY FROM IN-PERSON TO VIRTUAL

3.1 Course Structure

Software Testing online vs. in-person had very similar structures. The timeline of Software Development topics throughout the classes Fall 2019[8] and Spring 2021[10] did not vary much, if at all. The general outlines of the lectures did not waver. The course structure and grading scheme were almost identical.

3.2 The "Duh-Huh" Game for TDD, to Demonstrate Minor Exercise Adjustments

In both versions, most every lecture is paired with some activities. These activities are meant to help students understand the concepts, apply and reinforce the concepts to real-world scenarios, as well as help students to discover the importance of the topics covered in lecture in the real-world in the first place.

One hands-on activity that we used in both semesters is called the "Duh-Huh" game. Two main purposes of this activity are to introduce the Test-Driven Development (TDD) and to practice test automation using JUnit. Students who are playing will take turns counting up from one, one number per turn, in sequence. If the number is divisible by 3, they will say "Duh," if it is divisible by 5, they will say "Huh," and if it is divisible by both 3 and 5, they will say "Duh-Huh." It is then suggested that making a program to print what the players should say each turn would be helpful for the flow of the game. The students follow the TDD process as discussed earlier in lecture: first by writing a user story, then creating requirements out of the user story, transforming the requirements into tests, writing the JUnit test code, and then iteratively writing program to

solve the Duh-Huh game and running the tests until the program passes the tests. Through this activity, students learn the TDD process, understand the importance of test harnesses, and recognize the benefit of using tests as guidelines to develop software. More information on the exercise can be found at <http://www.cs.virginia.edu/~up3f/cs3250/inclass/potd01-TDD-get-started.html>, and this exercise takes about 15-20 minutes.

This "Duh-Huh" game translated fairly well to the virtual environment, with some minor adjustments adapting it smoothly. Whereas the Fall 2019 semester involved students positioned in a circle with a natural ordering around the circle, the Spring 2021 semester had to have order assigned over Zoom. Though this change was made, we found that the "Duh-Huh" game seemed to have a similar impact as before and similar enthusiasm among participants as well. Thus, we deem this exercise as useful in both contexts, according to our initial observations within the classrooms.

This exercise is a good demonstration of how many of our exercises translated to online. No actual rules or requirements changed, but the implementation was slightly altered to match the Zoom environment. The vast majority of our exercises changed in this way. Many did not even need alterations apart from it being on Zoom rather than in person. While the news of a virtual semester sounded intimidating at first, we were pleased to see how well many aspects of our class translated to virtual learning.

4 LIMITATIONS FROM VIRTUAL SETTING, AND THOUGHTS ON HOW TO LESSEN THEM

With the class going virtual, some larger changes were made throughout the course of the online semester, as compared to the earlier in-person semester. There were several exercises that we chose to omit from the course once it became virtual, though the topics covered by these exercises were still included. We maintain that this was the right decision during the rapid transitions to online learning, but we also consider ways to translate these omitted activities to the virtual setting in the future.

4.1 Agile Airplane Testing

The agile airplane testing activity relies on using test harnesses as guidelines for students to produce and test paper airplanes. Due to its nature where developers build paper airplanes and testers evaluate the quality of the airplanes by throwing them, the activity was difficult to implement in an online setting. As the virtual environment made it so that students were located in various places, the build-test team dynamic was impossible to achieve in the same manner. It was scratched accordingly.

A potential way to make this exercise possible if there were to be another virtual semester would be to have a riddle or code on a website for the students (who act as developers) to try and solve (perhaps a bingo board with the words shuffled for the first player and the "correct" board hidden from both players). Another student (who acts as a tester) could assess and provide recommendations by entering the developer's solution ideas. Subsequently, the website could give some sort of feedback for the tester to relay back to the developers. The developers could then use the suggestions in the subsequent development to improve the quality of their solution.

4.2 Candy-Coverage

A candy-coverage testing activity was difficult to implement online. Originally, the activity included each group being given two bags of candy to test. The activity included several steps: creating coverage criterion, designing tests to satisfy the coverage criterion, executing the tests on one candy bag, exchanging the tests with another group, executing that other group's tests on the second bag of candy, and then analyzing and comparing the quality of both test sets. As the virtual environment made it difficult to provide teams who were not in the same room as us with candy bags, a problem exasperated by the fact that teams were not in the same location as each other either, this exercise was omitted in the virtual semester.

To make this exercise exist online, a webpage could be made that randomizes a "candy bag" on the screen. Partners could design and develop tests in breakout rooms, with the instructor and/or TAs checking in on the rooms. This would have a less tasty reward at the end of the exercise, but it would accomplish the same goal.

4.3 Cross-Course Collaboration

Lastly, in the in-person semester there was an extra credit assignment which involved collaboration with the Data Structures and Algorithms 1 course in our department. In this assignment, Software Testing students would create tests for the BSTs and AVL assignment of the Data Structures and Algorithms 1 course in order to mimick a more realistic, real-world coding and testing dynamic. This collaboration faltered in the rush of the Spring 2021 semester. Thus, even though it was an optional assignment in the first place, it ceased to be an option in the hectic Spring 2021 semester. This would be easy to pick back up, since the collaboration was mostly virtual in the first place.

5 SURVEY RESULTS

We conducted a pre-survey and a post-survey from both semesters. The participants in the surveys were students from the course. These surveys were anonymous and optional. They had no bearing on course grades. In fact, the course instructor was excluded from the survey processes and had no access to the survey results until after grades were finalized and the semesters were concluded in order to combat any potential bias in survey responses.

5.1 Summary of Post-Survey Results

For both the virtual and the in-person semesters, students seemed more confident and interested at the end of the semester than at the start. This indicates that the class is useful in increasing student intrigue and confidence in their abilities.

Table 1 shows the difference between pre-survey and post-survey responses to statements on the topic of interest in software testing during the virtual semester. The results are fairly in-determinant, as despite there being fewer neutral responses to all statements, the disagreement and agreement to the phrases remain fairly balanced. The two clearest outcomes are that more students like testing their software by the end of the semester and that more students disagree that Software Testing is an intellectually rewarding field of study. There is also more strong agreement that software testing is exciting. These results indicate that the students may desire to test their own

Table 1: Differences Between Virtual Pre-survey and Post-survey on Student's Interest in Software Testing (by percent).

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I like testing my software	0	-9.51	-7.5	21.99	-4.96
I find software testing interesting	0	2.01	-4.86	5.6	-2.75
Detecting hard-to-find problems in software is rewarding	0	2.22	-2.54	1.26	-0.95
Software testing is exciting	0	-0.42	-0.74	-0.85	2.01
Software testing is an intellectually rewarding field of study	0	6.66	-5.08	-1.16	-0.42

Table 2: Differences Between Virtual Pre-survey and Post-survey on Student's Expectation of Success in Software Testing (by percent).

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I am confident in my ability to develop high-quality tests	4.87	-11.18	-10.96	14.62	2.66
I am confident in my ability to excel in my software testing coursework	-2.33	2.66	-1.77	5.54	-4.1
I believe I can learn the necessary skills to become an effective software test engineer	0	2.43	-9.08	17.61	-10.96
I have the necessary skills to become an effective software test engineer	-4.65	-4.31	-4.21	7.98	5.21

code more frequently, but on the most part do not desire software testing to be the focus of their careers.

Table 2 shows the difference between pre-survey and post-survey responses to statements on the topic of student expectation of success in Software Testing during the virtual semester. Although there were some mixed results, student confidence increased overall. The statement regarding confidence in their ability to excel in coursework shows students becoming slightly more moderate, with a shift from "Strongly Disagree" to "Disagree," and a similar shift from "Strongly Agree" to "Agree." Students indicated a greater confidence in their possession of necessary skills most clearly. There also seems to be an increase in confidence in regards to high-quality test development. Overall, it seems that more confidence was gained throughout the virtual semester by students, but that there was also a shift to stating these expectations less strongly in general.

As for the in-person semester [11], it had slightly more agreement in most categories, but since the virtual learning environment was so new and at least some of our students had additional stressors during the difficult semester, this result is not surprising. In fact, the shifts were fairly small.

With all of our collected data, it is important to keep in mind that these observations were based on solely two semesters: one being in-person, and one being virtual. Our sample size is small, and so it would be apt for us to keep collecting responses in future semesters for more generalization.

6 CONCLUSION AND REMARKS ON MOVING FORWARD

Technology aided us greatly as we moved our Software Testing course online. Although the virtual class seemed daunting, we were able to keep the class structure very similar to the in-person class. The few exercises we did omit are adaptable for online settings in future semesters. We discovered new capabilities in regards to Office Hour flow and checking in with students' emotions. We see greater accommodations for students within class recordings. Such discoveries can shape in-person semesters as well. We experienced the first fruits of virtual guest lectures. Overall, we saw continued

success in improving student interest and confidence in Software Testing, despite the challenging semester being made virtual.

REFERENCES

- [1] Paul Ammann and Jeff Offutt. 2016. *Introduction to Software Testing*. Cambridge University Press, Cambridge, UK.
- [2] Moritz Beller, Georgios Gousios, Annibale Panichella, and Andy Zaidman. 2015. When, How, and Why Developers (Do Not) Test in Their IDEs. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (Bergamo, Italy) (ESEC/FSE 2015)*. ACM, New York, NY, USA, 179–190. <https://doi.org/10.1145/2786805.2786843>
- [3] Maria A. S. Brito, João L. Rossi, Simone do Rocio Senger de Souza, and Rosana T. V. Braga. 2012. An Experience on Applying Software Testing for Teaching Introductory Programming Courses. *CLEI Electron. J.* 15 (2012).
- [4] Lia M. Daniels, Lauren D. Goegan, and Patti C. Parker. 2021. The impact of COVID-19 triggered changes to instruction and assessment on university students' self-reported motivation, engagement and perceptions. *Journal of Social Psychology of Education* 24 (2021), 299–318. <https://doi.org/10.1007/s11218-021-09612-3>
- [5] Thurston Domina, Linda Renzulli, Brittany Murray, Alma Nidia Garza, and Lysandra Perez. 2021. Remote or Removed: Predicting Successful Engagement with Online Learning during COVID-19. *Sociological Research for a Dynamic World* 7 (2021), 1–15. <https://doi.org/doi/pdf/10.1177/2378023120988200>
- [6] Sebastian Elbaum, Suzette Person, Jon Dokulil, and Matt Jorde. 2007. Bug Hunt: Making Early Software Testing Lessons Engaging and Affordable. In *Proceedings of the 29th International Conference on Software Engineering (ICSE '07)*. IEEE Computer Society, Washington, DC, USA, 688–697. <https://doi.org/10.1109/ICSE.2007.23>
- [7] The Organisation for Economic Co-operation and Development (OECD). 20 (accessed May 25, 2021). The impact of COVID-19 on student equity and inclusion: Supporting vulnerable students during school closures and school re-openings. <https://www.oecd.org/coronavirus/policy-responses/the-impact-of-covid-19-on-student-equity-and-inclusion-supporting-vulnerable-students-during-school-closures-and-school-re-openings-d593b5c8/>
- [8] Upsorn Praphamontripong. [n.d.]. Fall 2019 Software Testing Schedule. <http://www.cs.virginia.edu/~up3f/cs3250/schedule-F19.html>
- [9] Upsorn Praphamontripong. [n.d.]. How Are You Feeling? <http://www.cs.virginia.edu/~up3f/cs3250/how-are-you-feeling.html>
- [10] Upsorn Praphamontripong. [n.d.]. Spring 2021 Software Testing Schedule. <http://www.cs.virginia.edu/~up3f/cs3250/schedule.html>
- [11] Upsorn Praphamontripong, Mark Floryan, and Ryan Ritzo. 2020. A Preliminary Report on Hands-On and Cross-Course Activities in a College Software Testing Course. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 445–451. <https://doi.org/10.1109/ICSTW50294.2020.00080>
- [12] Michael Prince, Richard Felder, and Rebecca Brent. 20 (accessed May 25, 2021). The impact of COVID-19 on student equity and inclusion: Supporting vulnerable students during school closures and school re-openings. <https://www.engr.ncsu.edu/wp-content/uploads/drive/1PGLZxoVvKctmiyvXTXtbw5ICLwZLDxah/2020-AEE-COVID-19-Felder.pdf>