Adapting to Online Teaching in Software Engineering Courses

Simona Motogna
motogna@cs.ubbcluj.ro
Babeş–Bolyai University
Cluj-Napoca, Romania

Andrian Marcus
amarcus@utdallas.edu
University of Texas at Dallas
Dallas, USA

Arthur-Jozsef Molnar
arthur@cs.ubbcluj.ro
Babeş–Bolyai University
Cluj-Napoca, Romania

ABSTRACT

The COVID-19 worldwide pandemic caused sudden and unexpected changes in how we teach software engineering and other university courses. This paper presents an empirical study that aims to improve our understanding on how the assessment of student learning changed, in response to the transition from in-class to online courses. A questionnaire was distributed to instructors across the globe. The results indicate that the evaluation methodologies for most reported learning objectives have changed. Not surprising, in-class oral presentations and in-class exams are no longer used by the instructors for evaluations. We observed a trend of having fewer exams and more project-related evaluations after the transition. Not all instructors changed the way they evaluated student learning after the transition, however the majority reported their effort in student learning assessment increased after the transition, whether they made changes in methodologies or not.

CCS CONCEPTS

• **Social and professional topics → Student assessment**.

KEYWORDS

student learning assessment, Software Engineering education

ACM Reference Format:
Simona Motogna, Andrian Marcus, and Arthur-Jozsef Molnar. 2020. 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 (EASEAI '20), November 9, 2020, Virtual, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3412453.3423194

1 INTRODUCTION

Global education was hit by a major problem when the COVID-19 pandemic spread throughout the world. From pre-school to post-graduate level, in a very short time everyone had to adapt from a traditional education with face to face courses, project work, team assignments and evaluation to online activities. The transition raised several challenges to both instructors and students, impacting the learning experience, student engagement, and student learning assessment.

While some programs had online courses before the pandemic, most of the instructors in academia do not have much experience with or training in remote/online teaching. In addition, the transition happened very fast, typically within a couple of weeks.

As of July 15, 2020, the COVID-19 pandemic is far from over and many universities plan to have online or hybrid courses in the Fall semester. This paper presents the results of an empirical study aimed at investigating how instructors adapted their learning assessment methodologies during the transition. Specifically, we focused on how the evaluation of student learning changed in software engineering and related courses. From here on, for simplicity and readability, we use the term *course* for referring to an undergraduate or graduate-level software engineering or related course, unless otherwise specified. Likewise, we use the term *transition* for referring to the transition from in-person to online classes during the COVID-19 pandemic, unless otherwise specified.

The study relies on a survey, distributed to instructors who taught or are teaching software engineering courses in the past few months. The answers provide insights into which assessment methodologies are most impacted by the transition and which specific learning objectives. We also learned that the majority of the respondents experienced an increased effort in evaluating the student learning, after the transition, compared to before the transition. We hope that these insights will help instructors preparing for teaching in the Fall semester.

The paper is organized as follows. Section 2 describes the design of our empirical study, while Section 3 analyzes and discusses the results of the study. The main threats to the validity of our observations are discussed in Section 4 and Section 5 gives a brief overview of related work. Finally, conclusions and future work are presented in Section 6.

2 EMPIRICAL STUDY

We conducted an empirical study in order to understand how instructors responded to the unplanned transition from in-person to online software engineering classes in higher education. We aim at answering the following research question:

RQ: *How is the assessment of student learning influenced by the transition?* Based on the experience of the authors, we expected the transition to an online environment to have lead to changes in the assessment methods used to fulfill certain learning objectives, especially those relying on in-class presence and interactions.

In order to answer our research question, we designed and distributed a survey with questions about the learning objectives and student learning evaluation methods used in software engineering and related courses, before and after the transition.

EASEAI '20, November 9, 2020, Virtual, USA

ACM ISBN 978-1-4503-8102-4/20/11…$15.00
https://doi.org/10.1145/3412453.3423194

Table 1: Survey Questions

No	Question text	Question type
Q1	When did the online transition happen?	Multiple choice
Q2	Please provide a link to the course syllabus, if available online. Feel free to remove any personally identifiable information	Open text
Q3	What is the level of the course you are teaching?	Multiple choice
Q4	What type of students are enrolled in the course?	Multiple choice
Q5	What is the title of the course?	Open text
Q6	In what country do you teach this course?	Open text
Q7	Which are the learning outcomes/course objectives for your course? (select all that apply)	Multiple choice & Open
Q8	Did you change the way you assess any of the learning outcomes during the transition?	Yes/No
Q9	Which of the learning outcomes are assessed differently during the transition? (check all that apply)	Multiple choice
Q10	Briefly describe how you assessed student learning, before the transition, for: .... *(each of the modified learning outcome)*	Open text
Q11	Briefly describe how you assessed student learning, after the transition, for: .... *(each of the modified learning outcome)*	Open text
Q12	How did the effort spent on student learning assessment change in the online teaching format, compared to the effort spent before the transition? If your semester is still ongoing, please provide your best estimate	Likert scale
Q13	If you have additional information about the changes in learning objectives assessment during the transition to online teaching,or any other comments, please write them below	Open text

2.1 Survey Design

The survey has a variable number of questions, depending on the answers to some of them (see Table 1), organized in four parts. **Part one** (questions Q1-Q6) is meant for collecting information about the course: title, location (country), and students level (undergraduate or graduate, Q4). The questions in part one are optional. Question Q2 is asking for the URL of the course syllabus, if available. The answer may reveal personally identifiable information, which we removed from any data under analysis, when present.

Part two (question Q7) is meant for collecting information about the learning outcomes of the courses. Respondents need to select or input the learning objectives of their course. The question from the second part is mandatory, meaning the respondents had to provide an answer before proceeding to the next questions.

Part three (questions Q8-Q11) is meant to collect information about the changes in the evaluation of the student learning objectives. Q8 asks whether there were any changes in the evaluation of the learning outcomes in response to the transition. If not, the subject is directed to the final question. A positive answer will generate Q9, which asks the subject to select the learning objectives for which the evaluation changed. Then for each selected learning objective a pair for questions is generated asking how the learning evaluation was done before the transition (Q10) and after the transition (Q11), each answered in an open text format.

Part four consists of Q12 and it is meant to be answered by all respondents, regardless of their previous answers regarding the changes in the evaluation of learning outcomes. The answers (on a Likert scale) reflect how the evaluation effort changed during the transition: significantly increased, somewhat increased, about the same, somewhat decreased, significantly decreased.

The **final question** (Q13) allows respondents to add any additional information in an open text format.

Question Q7 allows respondents to select the learning objectives of their course from a list of ten such objectives and/or to add up to five "other" learning objectives.

In order to establish the list of ten learning objectives, we used the expected student outcomes stated by the ACM Curriculum Recommendations for Software Engineering [1] to create an initial list. We then compared this list with the objectives listed in a set of 35 undergraduate and graduate SE courses taught at different universities worldwide, extracted from their online syllabi. Finally, we selected the ten most frequently appearing learning outcomes.

The survey was tested by the authors and piloted a few instructors from the authors' institutions, in order to collect feedback regarding the comprehensibility of the questions and the time needed to complete the survey.

2.2 Survey Distribution

We generated a public link to the survey, meaning that anyone with the link could complete the survey, and we sent it via email to 77 professional acquaintances of the authors. The email also asked them to forward it to other instructors who may complete the survey. We did not track how many of the original email recipients forwarded the email to others.

3 RESULTS AND ANALYSIS

We received 56 answers, from which 32 were complete. We eliminated three complete answers, as they concerned courses that we considered to be less related to software engineering (*i.e.* other, Algorithms, Compilers, Programming Languages), compared to the other answers. We established this information based on the answers to Q2, Q5, Q7 (see Table 1). All the results and analyses we present in this paper are based on the remaining 29 answers.

Among the remaining 29 complete surveys, 23 are about Software Engineering classes, while six are about related courses, namely:

Table 2: Learning outcomes before the transition and modified during transition

No	Learning outcome	Evaluated (% of 29)	Evaluation changed
1	Ability to understand software lifecycle development models	17 (58.62%)	5 (29.41%)
2	Ability to understand and apply software requirements engineering	17 (58.62%)	4 (23.53%)
3	Ability to understand and apply software design principles	23 (79.31%)	6 (26.09%)
4	Ability to understand and apply software testing techniques	19 (65.52%)	6 (31.58%)
5	Ability to understand the use of metrics in software engineering	9 (31.03%)	2 (22.22%)
6	Ability to understand formal methods in software development	5 (17.24%)	3 (60.00%)
7	Ability to establish and participate in an ethical software development	6 (20.69%)	2 (33.33%)
8	Ability to use CASE tools for software development	9 (31.03%)	2 (22.22%)
9	Ability to develop, maintain and evaluate large-scale software systems	11 (37.93%)	4 (36.36%)
10	Understanding of the role of project management including planning, scheduling, risk management	13 (44.83%)	3 (23.08%)
11	Other: Design and develop video games	1 (3.45%)	0 (0.00%)
12	Other: Software Evolution	1 (3.45%)	1 (100.00%)
13	Other: Plan, implement, deliver software system to a professional client	1 (3.45%)	0 (0.00%)
11	Other: Functional programming	1 (3.45%)	1 (100.00%)
11	Other: Language Theory	1 (3.45%)	1 (100.00%)

Telematic Applications and Services, Object Oriented Languages and Environments, Object Oriented Programming, Web programming, Videogames and Virtual Reality, and CS Capstone.

3.1 Background Information

While we did not target a representative sample for our study, we collected information to support the external validity of our findings. Note that all questions were optional, so some respondents did not answered all of them.

The geographical distribution of the respondents is Europe (17), USA (7), South America (3), China (1), and unspecified (1). This information was inferred from the answers to question Q6.

Regarding the level of the courses taught by the instructors responding to the survey (see Q3), 7 courses were introductory, 12 intermediate, and 10 were advanced. Based on the answers to Q4, we learned that 22 of them targeted undergraduate students, while 7 were targeted towards graduate students.

With the exception of China, most universities transitioned to online courses sometimes in March, 2020. This date corresponds to different parts in the timeline of a course, depending on the structure of the semester. For example, for most US institutions using the semester model, mid March corresponds roughly with the middle of the semester. In 15 instances the transition happened during the first third of the semester, in 9 during the middle third, while in 5 cases it occurred in the last third of the semester.

3.2 RQ - Learning Assessment Changes

The respondents selected a total of 15 distinct learning outcomes used in their courses, most of them from the given list of ten provided in question Q7, as shown in Table 2. The courses taught by the respondents have an approximate average of 5 (4.96) learning outcomes per course. Among the 15 learning objectives, the most frequently mentioned were: *ability to understand and apply software design principles* (23 instances); *ability to understand and apply software testing techniques* (19 instances); *ability to understand software lifecycle development models* (17 instances); and *ability to understand and apply software requirements engineering* (17 instances). See table 2 for information on all the 15 learning objectives.

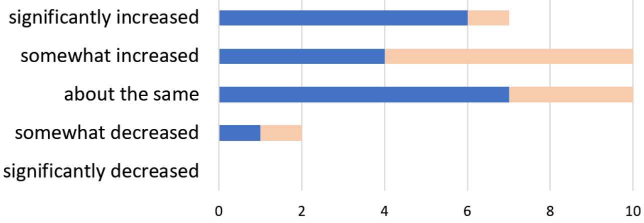

Figure 1: Effort required in student assessment after the transition, compared with before. Blue shows respondents who reported changes in the evaluation methods, while orange shows respondents who did not report changes.

Quantitative analysis. The answers to Q8 and Q13 (see Table 1) allow us to quantitatively assess the impact of the transition on the way learning objectives are evaluated in software engineering courses. Based on the answers to Q8, of the 29 respondents, 11 (37.93%) did not update the evaluation for any of the learning outcomes, while 15 modified the assessment methods for at least one learning objective and specified the changes. Three respondents stated that they changed the assessment methods, but did not specify for which learning objective.

The answers to Q12 (Figure 1) reveal that the impact of the transition affected in most of the cases the effort allocation for assessment. Of the 29 respondents, 17 (58.62%) reported that their effort in assessing the student learning increased significantly (7–24.13%) or somewhat (10–34.48%). Ten respondents (34.48%) reported no effect on the evaluation effort, while two of them (6.4%) reported somewhat less effort spent on the evaluation after the transition.

Table 3: Labels for learning assessments methods

Label	Definition	Example
Project (or specific deliverables)	Large assignment, involving software development, with set requirements, deadlines and milestones; over a medium or long period of time; individual or in teams; with one or more deliverables.	*"the application of a software lifecycle model to a project"*
Exam	Answering one or more questions, problems, tests, quizzes	*"oral exam, written exam, multiple choice test"*
Lab activity	Short time practical assignment, usually supposed to be done on site or in the lab.	*"design test cases in the lab"*
Homework	Assignment that is supposed to be solved at home, within a time limit.	*"finish the assignment at home"*
Oral presentation in class (OPC)	Oral communication, including Q&A and feedback.	*"I ask students in a team to come and demonstrate their work to the class"*
Oral presentation online (OPO)	Remote communication using some supporting platforms.	*”Each student had to share own screen and code”*
Recorded presentation (RecP)	Recorded presentation is submitted online (missing Q&A session).	*"video posted by students"*
Written document in class (WinC)	Paper based and supervised; evaluation is performed afterwards.	*"written exam"*
Written document online (at home live) (WOn)	Online with some form of supervision or constraints; evaluation is performed afterwards or instant.	*"online exam"*
Written document offline (at home not live) (Woff)	Completed at home with no supervision. Evaluation is done afterwards.	*"assignment for which the students had two days to respond; project documentation"*

No respondent indicated that their effort significantly decreased. It is interesting to observe that for seven (of 11) respondents who did not change their evaluation methods, the evaluation effort increased after the transition. Conversely, eight (of 18) respondents who changed their evaluated methods reported no increase in the evaluation effort.

For 13 of the 15 learning outcomes evaluated in the courses taught by the respondents, the evaluation changed after the transition (see Table 2). Most affected learning outcomes are: the *Ability to understand and apply software design principles* and the *Ability to understand and apply software testing techniques* (changed in 6−15.0% instances), followed closely by the one referring to *lifecycle development models* (changed in 5−12.50% instances each). However, these are also the most frequently evaluated learning objectives. When normalizing, the most frequently changed learning outcome is the *Ability to understand formal methods in software development*, as it changed in 3 out of 5 (60%) instances where it is evaluated.

Qualitative analysis. For each learning objective selected in Q8 ("*Which of the learning outcomes are assessed differently during the transition? (check all that apply)*"), the respondents had to describe in free text form the methodology they used for learning assessment before and after the transition.

Before analyzing the answers, we edited some of them in order to avoid confusion and improve readability. For example, for the response ”as for the previous case” we identified to which previous answer the respondent referred to and we replaced it with that answer. Overall, we edited the answers for five respondents. In two cases, respondents stated that they changed the assessment methods, but no learning objectives were specified for these changes. They specified the changes in the last question of the survey (Q13). We considered these answers valid, even if the changes were not linked to any specific learning objective. Two other particular cases needed special attention: one was missing the description of the assessment method before the transition and the other eliminated some of the assessment methods after the transition.

We employed *open coding* [5] for analyzing the answers to each instance of Q10 and Q11. The goal of the open coding analysis was to identify, within the free text, the evaluation methods used by the respondents in assessing the learning objectives before and after the transition.

Coding protocol. Based on the experience of the authors and consulting the syllabi from 35 software engineering courses we retrieved from the internet, we developed an initial set of labels. Each label corresponded to one assessment methodology (*e.g.*, written exam, oral presentation, *etc.*). We provided a short definition for each label, a list of synonyms, and examples. Two of the authors independently coded the answers, using the initial label set. The answers were coded with one or more labels. The process was iterative. When a new label was identified by one of the coders, it was communicated to the other coder and discussed. If the authors agreed that this is a valid label, then the coded answers were inspected again, together, to see whether the new label applies or not. In subsequent iterations, some labels were merged (*e,g,*, *practical assignment* was merged with *project*). Conflicts during labeling were resolved through discussion. Our methodology does not allow

Table 4: Distribution of assessment methods before and after the transition (acronyms defined in Table 3)

	Project					Exam				Lab	Homework
	OPC	OPO	RecP	Won	Woff	OPC	WinC	WOn	WOff	WinC	WOff
Before	4				1	1	8			1	
After		3	2	3	1			5	2		1

computation of Cohen's kappa coefficient for inter-rater reliability [4]).

In the end, we identified 10 labels that describe assessment methodologies. They are defined in Table 3. The labels are classified in two groups corresponding to the evaluation method (4 labels) and evaluation format (6 labels). Using one label from each category completely describes a particular assessment methodology used by instructors. For example, considering the answer *"project on an existing software tool. The oral discussion will be turned into a discussion on the project: problem, solutions and technologies used to accomplish it"*, then the answer is labeled with *project + oral presentation online (OPO)*. This means that the instructor evaluated each student's project through an oral presentation done online by the student, in the presence of the instructor.

Changes in learning assessment methodologies. Table 4 shows which assessment methodologies were used before and after the transition. For example, in 4 cases, the *Project* was evaluated via an *oral presentation in class (OPC), before* the transition. This type of evaluation was not used *after* the transition by any respondent. Likewise, in 3 cases, the *Project* was evaluated via an *oral presentation online (OPO), after* the transition, a method that was not used by any respondent *before* the transition.

Before the transition, the most used method of assessment was the *exam* (9 cases), while after transition *projects* were more widely used (9 cases), as shown in Table 4. Before the transition, in most of the cases (13 out of 15, see Table 4) *in class* evaluation either written, oral or as lab assignment was used. After the transition, the prevailing form of assessment was live online (13 out of 17 cases). In few cases, offline methods were used, such as, *recorded presentation* (2 cases) or *written documents* (4 cases–*e.g.*, project documentation or take at home exam).

The number of assessment methodologies, shown in Table 4, used before (row 3) and after (row 4) the transition is not the same because two respondents did not report the before methods only the after ones, while one learning outcome was no longer evaluated. Table 5 shows these cases.

The table also shows how did the changes counted in Table 4 changed. Analysing the data in Table 5 we conclude that, the method of evaluation (*i.e.*, project, exam, lab, homework) is preserved in most cases, with only three exceptions.

Table 5: Changes in assessment methods

Before	After	Count
Project OPC (Count 4)	Project OPO	1
	Project OPO & WOn	1
	Project RecP	1
	Exam WOn	1
Project WOff (Count 1)	Project WOn	1
Exam WinC (Count 8)	Exam WOn	4
	Exam WOff	2
	Project WOff	1
	eliminated	1
Exam OPC (Count 1)	Project OPO	1
Lab WinC (Count 1)	Homework WOff	1
unspecified (Count 2)	Project WOn	1
	Project RecP	1

(acronyms defined in Table 3)

4 THREATS TO VALIDITY

As with any empirical observations, the reported results and conclusions are subject to certain threats to validity. In order to mitigate most of them, we followed best practices for survey design [9, 11]. The followings are the major threats to the validity of our work and the ways we tried to mitigate them.

Internal validity concerns factors that could have influenced our results. The respondents had to choose the learning objectives of their courses from a given list of ten. We selected these based on the Software Engineering Curriculum Guidelines [1] and 35 online syllabi for software engineering courses. Also, we allowed respondents to add up to five learning objectives not included in the list. Fiour respondents added such objectives, one each.

The distribution of the survey was via an anonymous link, which means that it could have been filled by anybody with the link. To avoid such a situation, we distributed the survey via direct email only to individuals we know. We also asked them to only forward it to other instructors they know personally. The link to the survey was not made public. The authors did not complete the survey.

External validity concerns the generalization of our findings. We do not claim generality of our conclusions and we did not aim to define and target a representative sample of software engineering instructors. We collected and reported geographical information to gauge the distribution of the respondents. In addition, the responses covered 15 different learning objectives, which are common in software engineering courses.

Construct validity concerns the relationship between theory and observation. Our working hypothesis that the transition caused changes in the evaluation of student learning was based on the experience of the authors. However, we did not assume that every instructor made such changes, so we specifically asked that question, which revealed that 11 of 29 respondents did not make such changes. Also, we did not prescribe the answers to the changes, instead the respondents were asked to describe them in free text form. We measured the changes in evaluation methodologies by contrasting descriptions of the methodologies used before and after the transition.

Content validity refers to the extent to which a measure represents all facets of a given construct. We used open coding to infer the set of learning assessment methodologies reported by the respondents. Coding was done by two of the authors, to address reliability. The methodologies we identified are not the only ones that could be used in evaluating student learning. It is also possible that another set of coders would have identified a somewhat different list of methodologies, as in some cases, as reported before, the coders had to make inferences from the text.

5 RELATED WORK

The subject of this study is unique, given the context of the unexpected and sudden transition to an online teaching environment. Given that numerous institutions of higher learning already offered computer science or software engineering programs in an online environment, some studies have already brought contributions to the topic.

For example, the challenges faced in the creation of an online software engineering program at Arizona State University are detailed in [6], and some of them target student assessment. One of these is related to evaluating the outcomes such as design, teamwork and critical thinking which was ascertained to be difficult and after program implementation has been proved to be of low concern. Another issue was related to online interaction between students and academic staff, as well as between students themselves. Even if initially assessed as a medium concern, it has been proven to be one of the most important concerns during the transition to online. Academic integrity issues were reported as being solved, with proper assistance making the process easier to implement within an online environment. Our study did not address these related issues.

Several studies [3, 7, 10] compared offline and online exam grading. While [7, 10] argued that no significant difference exists in performance and learning outcomes for on-campus, respectively online students, the study presented in [3] investigates three main differentiators: (i) time efficiency; (ii) student experience: a survey identified that students strongly prefer online exams and grading; (iii) teaching staff opinions: online evaluation was more convenient and faster, but implied less communication and feedback.

Dedicated e-learning platforms for courses in computer science [2] can help the transition to online education. The platform developed by Alexandru *et al.* [2] was evaluated over two years in computer science courses and they reported reduced allocated time for student assessment and satisfying student feedback in terms of learning and evaluation experiences.

Ju *et al.* [8] reports an experience managing an online exam system for assessing programming skills, arguing that such a tool is more appropriate than pen and paper exams, and that software engineering courses can benefit by adopting such a tool. The study mentioned the issue of academic honesty as one hurdle that needed to be overcome. Our study did not focus on the challenges of the transition, but rather then form of the transition.

6 CONCLUSIONS AND FUTURE WORK

The answers to our survey revealed that the sudden transition to online teaching of software engineering and related courses, due to the COVID-19 pandemic, resulted in changes to the way instructors evaluate the learning objectives. We found that the evaluation methodologies for 13 of 15 reported learning objectives have changed. Not surprising, in-class oral presentations and in-class exams are no longer used by the instructors for evaluations. We observed a trend of having fewer exams and more project-related evaluations after the transition. Not all instructors changed the way they evaluated student learning after the transition (11 of 29). However, 17 of 29 respondents reported their effort in student learning assessment increased after the transition, whether they made changes in methodologies or not.

Our findings indicate that transitioning to online hybrid teaching for the entire semester will likely lead to even more changes, as the data we collected reflected changes made during the semester, when already some learning objective may have been evaluated.

Our survey focused on the changes in evaluation methodologies but did not address the rationale behind the changes nor the rationale for lack of changes. Likewise, we did not focus on the effectiveness of the changes. We expect that instructors will be able to address these issues after the 2020 Fall semester, when we plan to perform another, more in depth, study.

REFERENCES

[1] ACM. [n.d.]. *Software Engineering 2014 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering.* Retrieved June 22, 2020 from https://www.acm.org/binaries/content/assets/education/se2014.pdf

[2] Dan Alexandru, Adrian Iftene, and Daniela Gîfu. [n.d.]. Using New Technologies to Learn Programming Languages. In *Information Systems Development: Information Systems Beyond 2020, ISD 2019 Proceedings, Toulon, France, August 28-30, 2019*, Alena Siarheyeva, Chris Barry, Michael Lang, Henry Linger, and Christoph Schneider (Eds.). ISEN Yncréa Méditerranée.

[3] Yingjun Cao, Leo Porter, Soohyun Liao, and Rick Ord. 2019. Paper or Online?: A Comparison of Exam Grading Techniques. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '19)*. 99–104. https://doi.org/10.1145/3304221.3319739

[4] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educ Psychol Meas.* 20 (1960), 37–46. https://doi.org/10.1177/001316446002000104

[5] Uwe Flick. 2018. *An Introduction to Qualitative Research.* SAGE Publications.

[6] Kevin Gary, Sohum Sohoni, and Timothy Lindquist. 2017. It's Not What You Think: Lessons Learned Developing an Online Software Engineering Program. In *Proc. of 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*. 236–240. https://doi.org/10.1109/CSEET.2017.45.

[7] Courtney M. Holmes and Christine Reid. 2017. A Comparison Study of On-campus and Online Learning Outcomes for a Research Methods Course. *The Journal of Counselor Preparation and Supervision* 9, 2 (2017). http://dx.doi.org/10.7729/92.1182

[8] An Ju, Ben Mehne, Andrew Halle, and Armando Fox. 2018. In-Class Coding-Based Summative Assessments: Tools, Challenges, and Experience. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education.* 75–80. https://doi.org/10.1145/3197091.3197094

[9] Johan Linåker, Sardar Muhammad Sulaman, Rafael Maiani de Mello, and Martin Höst. 2015. *Guidelines for Conducting Surveys in Software Engineering.* Technical Report. Lund University.

[10] Jasmine Paul and Felicia Jefferson. 2019. A Comparative Analysis of Student Performance in an Online vs. Face-to-Face Environmental Science Course From 2009 to 2016. *Frontiers in Computer Science* 1 (2019), 7. https://doi.org/10.3389/fcomp.2019.00007

[11] Per Runeson and Martin Höst. 2008. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14 (2008), 131–164. https://doi.org/10.1007/s10664-008-9102-8