Software Security Education at Scale

Christopher Theisen
North Carolina State
Department of Computer Science
Raleigh, NC 27606
crtheise@ncsu.edu

Laurie Williams
North Carolina State
Department of Computer Science
Raleigh, NC 27606
lawilli3@ncsu.edu

Kevin Oliver
North Carolina State
Friday Institute Research Fellow
Raleigh, NC 27606
kevin_oliver@ncsu.edu

Emerson Murphy-Hill
North Carolina State
Department of Computer Science
Raleigh, NC 27606
emurph3@ncsu.edu

ABSTRACT
Massively Open Online Courses (MOOCs) provide a unique opportunity to reach out to students who would not normally be reached by alleviating the need to be physically present in the a classroom. However, teaching software security coursework outside of a classroom setting can be challenging. What are the challenges when converting security material from an on-campus course to the MOOC format? The goal of this research is *to assist educators in constructing software security coursework by providing a comparison of classroom courses and MOOCs.* In this work, we compare demographic information, student motivations, and student results from an on-campus software security course and a MOOC version of the same course. We found that the two populations of students differed, with the MOOC reaching a more diverse set of students than the on-campus course. We found that students in the on-campus course had higher quiz scores, on average, than students in the MOOC. Finally, we document our experience running the courses and what we would do differently to assist future educators constructing similar MOOC's.

General Terms
Security and Human Factors, Software Engineering

Keywords
Security, Education, MOOCs, Online Learning, Software Engineering

1. INTRODUCTION
Science, Technology, Engineering and Math education (STEM) is of increasing importance around the world as technology becomes more ubiquitous in our lives. However, there is a lack of skilled workers in these areas. For instance, the computer industry is facing a severe shortage of security professionals[1,2]. In January 2014, the 2014 Cisco Annual Security Report estimated a potential shortfall of a million security professionals globally [19]. A shortfall of security professionals has serious implications for the current state of security worldwide. How are security professionals supposed to meet the growing number of threats in cyberspace while being critically short-staffed?

To keep up with the demand for skilled staff, new ways to train current software engineers are needed. By providing ways for engineers to transition to security roles, education can help bridge the gap in the supply and demand needs of the industry. Software security can be thought of as software engineering practices associated with building secure software, so current software engineers are good candidates for training on security issues. Such outreach efforts are difficult, but are required if the needs of the security community are to be met. Massively Open Online Courses (MOOCs) have emerged in recent years to help educate people in a variety of areas and are one possible avenue to bridge the outreach gap [1][2].

However, transitioning educational material designed for a classroom setting to an online format is not a simple process. A MOOC literature review by Liyanagunawardena et al. [22] explored the current research into MOOCs and some of limitations of the platform. These limitations include the lack of direct access to instructors for students due to scaling issues, high dropout rates of students, and how (or if) instructors should provide recognition for completing a MOOC.

Determining what gaps exist between current university classes and MOOCs for training software engineers in security is important for the administration of future MOOCs. Future MOOCs can use this information to improve the quality of courses in software security. The goal of this research is *to assist educators in constructing software security coursework by providing a comparison of classroom courses and MOOCs.* We ran an on-campus university course and a MOOC on software security using the same teaching materials. We then analyzed the effectiveness of each platform in teaching software security. We present information regarding the demographics, motivations and the performance of students in on-campus classes versus MOOCs to serve as a starting point for future educators to better design coursework.

We explore the following research questions:

RQ1: Why did software engineers sign up for the MOOC?

RQ2: How do software engineers in the MOOC perform on quiz and test questions relative to university students being taught in an on-campus setting?

RQ3: How well does the MOOC format work for software engineering professionals? What could be improved on for future courses?

To answer these questions, we compare two different class offerings for the same software security course. The first class offered was a graduate-level software security course taught by the second author and taught on-campus at North Carolina State University. The on-campus course was a "flipped" course, meaning the students had to listen to a podcast and a short video lecture prior to attending class, and the class period consisted of discussion and exercises. The second offering was a MOOC using Google's CourseBuilder[3] platform using the same course

[1] http://www.rand.org/news/press/2014/06/18.html

[2] http://www.zdnet.com/cybersecuritys-hiring-crisis-a-troubling-trajectory-70000032923/

[3] https://code.google.com/p/course-builder/

materials as the on-campus course. Before, during, and after these courses were run, we explored the research questions above.

In this paper, we make the following contributions:

- A comparison of demographic outreach in the on-campus course versus the online course.
- A comparison of student performance in the on-campus course versus the online course.
- An analysis of the outreach of the online course as compared to the class offered at the university.

The rest of the paper is organized as follows. Section 2 discusses the related work in the area of both software security education and transitions to online learning in general. Section 3 discusses the methodology used to construct both our course and our study. Section 4 presents the demographic information of the students in both courses. Section 5 discusses why students signed up for the course. Section 6 presents the quiz and test results across common questions in the two courses. Section 7 discusses student responses to the course. Section 8 presents the lessons learned by the instructors of the course. Section 9 discusses the limitations of this work. Section 10 presents the future work that can be done in this area.

2. RELATED WORK

In this section, we describe the learning platforms compared in this research, including a discussion on the current effectiveness of MOOCs deployed in other subjects. We also explore several recent works in software security MOOCs and their suggestions for MOOC construction. Finally, we look at prior research that has compared online learning options to university classes.

2.1 On-Campus Courses

For this research, we refer to an "on-campus" course as one taught in a university classroom or lecture hall by an instructor. Active learning is recommended by Felder et al. [15] as a good approach to engage students in engineering. Prince [14] describes active learning as containing two elements: student activity and engagement. "Flipping" courses, or having students interact with the instructor and each other during class time instead of having traditional lectures, has been shown to increase student engagement [16]. Online coursework could be better for student engagement, as students can take in a recorded lecture on their own time at their own pace. The classroom version of the course offered to students leverages these principles as in the class is "flipped" and the materials are posted online to foster student engagement.

2.2 Massively Open Online Courses (MOOC)

A MOOC is a course available to the general public on the Internet. Some MOOCs are offered for free to the public, while others have fees associated with taking the course or receiving a certificate of completion. MOOCs are generally offered by academic institutions in partnership with several companies [1], such as Coursera[4] or Udacity[5]. The first course considered a MOOC was offered by Stanford University in 2011. About 450,000 students signed up for three classes [1]. One of the benefits of MOOCs is their ability to reach students who would otherwise be unable to access educational materials in a variety of subjects. Hyman showed that MOOCs can help reach new audiences, with participants drawn in from countries across the globe [2]. However, MOOCs have also traditionally suffered from exceptionally high dropout rates, with up to 97% of registered students dropping out by the end of the course [17].

The MOOC version of the course is realized through the Google course builder platform, using the same course materials as the classroom version of the course. The current state of the art on MOOC research also covers how MOOCs are structured, and what has worked for previous courses. We next explore specific suggestions for MOOC construction in the following three sub-sections.

2.2.1 Structure of Content

Significant research has been done on the topic of the content structure in MOOCs for effective learning. Aiken et al. [3] suggest that lectures incorporated into MOOCs should be broken into manageable chunks for students, typically 5-15 minutes each. This breakdown is more manageable for students and prevents their minds from wandering as often. Kay et al. [22] echoes this sentiment, adding that short video lectures should be immediately followed by quizzes. In the Kay case study, four of the six MOOCs host their video lectures on YouTube instead of an internal solution. Kop [5] advises that students should be able to choose how they get new information during the course. Students who are more proactive in finding new information learn more effectively. Part of the role of the instructor is to teach students how to seek out new information. Kop also suggests that cultivating a community around the course benefits learning as students feel they can rely on one another, and that the instructors care about them and value their learning [7]. Pardos et al. [12] suggests that including multiple sources of information, such as wikis, books, videos, and discussion boards, can help students find their own way through the material. According to Cabiria [18], MOOC content should be a jumping off point instead of a terminal point for learning. One goal of MOOCs should be to motivate students to explore further on their own.

2.2.2 Interactions and Discussions

Belanger et al. suggest online discussion forums as an avenue for student interaction, with a moderating presence from the instructors and teaching assistants. Online forums can help generate the sense of community mentioned in section 2.2.1. Based on Vanderbilt University's first experiences with MOOCs, Bruff suggests that discussion forums be seeded with open-ended questions. These open-ending questions encourage student engagement from different perspectives [8]. Fournier et al. [9] recommend tool use as being key to the learning process in MOOCs. Introduction of tools to students not only assists in learning for the lessons they are required for, but the tools can also be used by students for self-learning opportunities long after the course is complete. de Waard et al. [13] discuss the opportunity for network building during a MOOC, and how these courses could be used to build a network of like-minded individuals for a field. This idea has important implications for the security space, as communication between organizations is key for quickly identifying and mitigating security threats. Kay et al. [21] recommends letting students consume the course in their own way. Some students may be uninterested in the quizzes and assignments and therefore "audit" the course by only watching the video lectures. These students should not be constantly reminded about the quizzes and assignments if those activities don't meet their needs.

[4] http://www.coursera.com/

[5] http://www.udacity.com/

2.2.3 Assignments and Assessments

The official Coursera course guide [10] suggests seeding questions during lectures. A few minutes of lecturing should be interrupted by questions periodically. Littlejohn [11] suggested that external artifacts, such as podcasts, articles, and reports from the relevant field could help tie course materials to the real world and emphasize their importance to students. For a security course, showing this real world importance of security information is key to motivate students. Pardos et al. [12] emphasize the effectiveness of weaving assignments in and out of instruction time. While mixing assignments and instruction during class is somewhat difficult to do in the classroom, they point out the ease of doing this in an online environment. Bruff suggests peer review as an avenue to provide more in-depth projects and assignments for students without overwhelming the course staff, as non-automated assignments are unfeasible once courses hit large numbers of enrolled students [8].

3. COURSE OFFERINGS

In this section, we describe the course offering of the on-campus class and the MOOC class.

3.1 Course Content

Both the on-campus and MOOC classes taught Software Security. Course modules focused on a vulnerability types and techniques for preventing or removing vulnerabilities. The stated goals of both course offerings are as follows:

- **Security risk management.** Students shall be able to assess the security risk of a system under development. Risk management will include the development of formal and informal misuse case and threat models. Risk management will also involve the utilization of security metrics.
- **Security testing.** Students shall be able to perform all types of security testing, including fuzz testing at each of these levels: white box, grey box, and black box/penetration testing.
- **Secure coding techniques**. Students shall understand secure coding practices to prevent common vulnerabilities from being injected into software.
- **Security requirements, validation, and verification**. Students shall be able to write security requirements (which include privacy requirements). They will be able to validate these requirements and to perform additional verification practices of static analysis and security inspection

Much of the course content focused on educating students on the OWASP Top 10[6] most critical web application security flaws and the IEEE Center for Secure Design Top 10 Design Flaws[7].

3.2 On-campus Course

The Fall 2014 semester was the fifth offering of the Software Security course at North Carolina State University. The course was run on a 15-week semester, with two class meetings per week. The instructional content consisted of a series of 5-15 minute lecture videos. Prior to each class period, students were required to listen to one or two videos and to read one news article on a security breach that had occurred that week. Students then took a quiz on the material to motivate the students to actually listen to the material.

Additionally, prior to each class period, students listened to a Silver Bullet Podcast[8], hosted by Gary McGraw and presented by Cigital. The purpose of including the podcast in the course is to keep students informed about the outlook of the current industry leaders in security. Students took a quiz on the podcast prior to class.

The on-campus course offering was a "flipped" course. "Flipping" a course refers to an instructional strategy of delivering instructional content outside of the classroom often via video lectures, podcasts and readings. Activities, including those that may have traditionally been considered homework and group work, are brought into the classroom. The on-campus offering of the course was the first time it was flipped. Previously, it was traditionally lecture-based.

Class attendance was mandatory. When students entered the classroom, they had to find their name on an index card and put the card in an "I'm here" pile. At the start of class, students were given some time to discuss the highlights on the podcast with one or two students sitting near them. Then, the instructor quieted the small group conversations and began class. The instructor randomly chose a card from the "I'm here" pile, and asked the student to share a highlight of the podcast. Students would be called until the instructor felt most of the important points of the podcast were discussed. At that point, she would fill in any remaining points she felt should also be highlighted.

The class would next be provided an exercise based upon the content of the video lecture. Students were invited to work in small groups on the exercise. Approximately 15 minutes prior to the end of class, cards would be drawn from the "I'm here" pile for students to share the results of the exercise. Sometimes students verbally shared their results sitting in their seats, sometimes they brought their laptops to the front of the room to share their work via a display projector, and sometimes students might show their handwritten work on the whiteboard via a document camera.

The students also had a semester-long course project. The project involved performing a security audit on the codebase of an open-source project using the knowledge they gained throughout the course. Students were free to pick an open source project on GitHub or Sourceforge. The goal of the project was to get students acquainted with the security challenges of a real, complex, messy project. Students performed several activities over the course of the project: a domain analysis to determine the types of threats their software system could face, an analysis of the design of the project, an inspection of the code itself, and a final report on the three portions of the project.

Course materials, including videos, lecture slides, and exercises are available on the course website[9].

[6] https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

[7] http://cybersecurity.ieee.org/center-for-secure-design.html

[8] https://www.cigital.com/podcast/

[9] https://sites.google.com/a/ncsu.edu/csc515-software-security/

3.3 MOOC Course

The course modules of the MOOC were similar to that of the on-campus course. The MOOC ran for 10 weeks. The students were provided each week:

- One introductory video of the two course instructors and the teaching assistant (TA) casually talking about an overview of the weekly content, any questions that had arisen the prior week on the message board or email questions, and anything notable on the assignments for the weekly.
- One video briefly reflecting on 1-3 security breaches that had occurred in the past week.
- Three 10-15 minute lectures on course content that covered the same material as the on-campus course.
- One Silver Bullet podcast

The Google CourseBuilder[10] platform was chosen for running the MOOC version of the course. CourseBuilder is an open source education platform provided by Google so educators can run their own courses on the subjects of their choosing. CourseBuilder was chosen because of previous successful classes run on the platform both at North Carolina State University and other educational institutions.

Students took a quiz on the material presented that week. In some weeks, students were provided interactive exercises related to the material. We had hoped to initiate a peer-evaluation scheme whereby students provided each other feedback on the exercises. However, the peer evaluation software did not work as planned. As a result, the students could opt to email their assignments to the TA for feedback. However, most students did not send assignment emails to TA. Students were not forced to keep up with the material presented each week and were allowed to go at their own pace until the end of the course. Students who completed all course materials by one month past the official end date of the class were provided a completion certificate. A "subreddit" was set up on the Reddit platform in order to facilitate communication between students while the course was running.

3.4 Assessment

The quizzes completed each class period/week allow for a comparison of student understanding in the two courses right after lecture material is presented. In addition to quizzes, a longer midterm and final exam are given to both sets of students. These exams are designed to check the student's overall understanding at the midpoint and endpoint of the course.

Students in both courses are exposed to security-related exercises in a variety of formats. While these exercises are not graded, the qualitative responses to the exercises are recorded for both sets of students in order to determine how the students felt about them, including what they learned, whether they felt it was useful, and any other interesting observations. By structuring the two courses such that they are closely related to one another, we can draw conclusions of the effectiveness of the learning avenue.

3.5 Data Collection

Before collecting demographic and academic performance statistics as found in the following sections, we asked students in the on-campus course and the MOOC for their permission to use their personal identifiable information as part of our research study. Students were not given any additional credit or preferential treatment for participating in the study, and the way students responded was kept from the graders to prevent bias when grading a student participating in the study versus a student not participating in the study. An approval from the Institutional Review Board for the Protection of Human Subjects in Research (IRB) at North Carolina State University was acquired[11] under before collecting student information in the graduate level course at the university.

Students were given two surveys during the course. The first survey was a pre-course survey, which collected demographic information and student expectations for the course. The second survey was a post-course survey, which asked students how their perspective changed on software security. Both the pre-course survey and post-course survey were issued through the Qualtrics[12] platform.

3.6 Limitations

While every effort was made to make the courses are similar as possible, limitations of the MOOC format meant that there were some differences between the courses. Table 1 describes the similarities and differences between the two courses. For quizzes in the MOOC, technical issues with Google CourseBuilder forced us to use Google Forms to generate our quiz content. While Google Forms provided immediate feedback to students on their answers, MOOC students were not able to track their overall progress. As we mentioned previously, the MOOC originally had a peer review project element, but peer review was scrapped because of technical issues. MOOC student discussions took place on the Reddit platform, with a course subreddit created to facilitate discussion. While the subreddit was originally designed to facilitate discussion, it was used by students to point out errors or ask questions about the assignments themselves.

Course Elements	*On-Campus*	*MOOC*
face-to-face lectures	Yes	No
online, video-based lectures	Yes	Yes
other online content (e.g.., podcasts)	Yes	Yes
online quizzes with automated feedback	Yes	Yes
group projects	Yes	No
hands-on assignments, exercises	Yes	Yes
peer review of assignments	Yes	No
face-to-face discussions, question answering	Yes	No
online discussions, question answering	No	Yes
relative self-pacing	No	Yes

Table 1: Course elements by section, detailing key areas where the courses differed from one another.

[10] https://code.google.com/p/course-builder/

[11] IRB number 3443

[12] http://www.qualtrics.com/

4. STUDENT POPULATION

We surveyed students both offerings of the course. We present the results of the surveys in this section.

4.1 Demographics

For the on-campus course at North Carolina State University, we had 116 graduate students registered, with 114 completing the course. At the start of the MOOC, we had 266 students registered, with 86 students taking the first quiz and 59 students finishing the entire course.

Table 2 contains the percentage of students in both sections of the course, by percentage in age brackets. The graduate course at NCSU is mostly students in their twenties, as one might expect for a graduate level university course. The MOOC had a much wider range of ages represented, with students from 22-59 represented heavily in the student population.

Table 3 contains gender information for both courses. Males were the larger percentage in both courses, but the on-campus course had a higher percentage of females than the MOOC.

Table 4 and Table 5 contain the percentage of registrants by race and the first language of those registrants. The on-campus course was almost entirely Asian (mostly of Indian descent), while the MOOC had a greater spread of representation, with whites as the largest group represented, followed by Asians. For language, Hindi and English were the most represented first languages for the on-campus course, while English was the most represented language in the online course.

Table 6 represents the educational level of registrants of both courses. As a graduate level course at a university, we assume 100% completion of an undergraduate degree (or its equivalent) before registration in the course. The MOOC had a higher spread of education, with about 80% of the participants having 4-year degrees or Master's degrees.

Table 7 is the percentage of registrants by employment status. Over 95% of the students in the on-campus course were either not working or working part time to support their schoolwork. By contrast, over 80% of the MOOC participants have full time jobs, mostly in computer science.

To further explore what types of working professionals are being reached by our MOOC, we asked two additional questions about the type of employment held by the students. First, we asked what type of organization they work for. The result is presented in Table 8. About 60% of participants work for a private company, with educational and governmental work following. Table 9 explores the workplace further, asking what subfield the student works in. The question allowed for multiple responses. Software Engineering was the most frequently cited work area, with computer security following it.

Age Categories	*On-Campus Class*	*MOOC*
Below 18	0.0	1.1
18-21	0.0	1.5
22-29	97.7	21.1
30-39	2.3	22.2
40-49	0.0	31.2
50-59	0.0	20.3
60+	0.0	2.6

Table 2: Percentage of Registrants by Age

Gender Categories	*On-Campus Class*	*MOOC*
Male	58.3	78.9
Female	41.7	21.1

Table 3: Percentage of Registrants by Gender

Racial Categories	*On-Campus*	*MOOC*
American Indian	0.0%	0.0%
Asian	97.7%	28.6%
Black	0.0%	7.5%
White	1.2%	55.6%
Hispanic	0.0%	4.1%
Multiracial	0.0%	3.0%
Other	1.2%	1.1%

Table 4: Percentage of Registrants by Race

First Language	*On-Campus*	*MOOC*
English	41.9%	77.4%
Chinese	3.5%	3.4%
Spanish	0.0%	1.9%
Hindi	44.2%	8.3%
German	0.0%	0.4%
French	0.0%	0.8%
Italian	0.0%	0.4%
Other	10.5%	7.5%

Table 5: Percentage of Registrants by First Language

Highest Ed Level	*On-Campus Class*	*MOOC*
High School	0.0%	5.6%
2-Year College	0.0%	5.3%
4-Year College	(assumed) 100.0%	46.2%
Master's	0.0%	33.5%
Doctoral	0.0%	3.8%
Professional	0.0%	4.1%
None	0.0%	1.5%

Table 6: Percentage of Registrants by Highest Educational Level Completed

Employment Status Categories	*On-Campus*	*MOOC*
Student Not Working	77.9%	9.8%
Student Working Part-Time	20.9%	0.8%
Student Working Full-Time	1.2%	0.0%
Working Part-Time	0.0%	4.1%
Working Full-Time	0.0%	82.7%
Not Employed	0.0%	2.6%

Table 7: Percentage of Registrants by Employment Status

4.2 Discussion

Based upon the demographic questions asked, we can say that we reached two different populations with these two courses. While the on-campus course had a demographic makeup that one would expect for a graduate level course at a university, the MOOC reached a large variety of computer science professionals that differed from the group reached by the on-campus course. Our result indicates that if the community wants to reach out to current professionals to educate them on best practices in software security and other advanced technical topics, MOOCs might be an effective approach. With the flexibility MOOCS offer versus traditional classroom settings, they fit better into the schedules of busy professionals. In addition, the MOOC participants skewed older than the on-campus students. Therefore, there is a higher likelihood that they have families with their own schedules to consider as well. All of these factors combine to make MOOCs an attractive platform for the working professional.

5. REASONS FOR ENROLLMENT (RQ1)

In this section, we explore the reasons software engineers enrolled in the MOOC. Questions include where the students heard about the class, and why the student signed up (the student's goals).

Employment Type Categories	*MOOC*
Not Employed	9.0
Self Employed	1.1
Non-Profit	3.4
Governmental	7.9
Private	59.4
Educational	13.5
Other	2.6
Multiple Types	3.0

Table 8: Percentage of Registrants by Employment Type

Employment Responsibility Areas	*MOOC*
Not Employed	6.0
Artificial Intelligence	2.0
Computer Architecture & Engineering	17.0
Computer Performance Analysis	8.0
Computer Graphics & Visualization	4.0
Computer Security & Cryptography	24.0
Computational Science	6.0
Computer Networks	16.0
Concurrent, Parallel, Distributed Systems	8.0
Databases	18.0
Health Informatics	4.0
Information Science	14.0
Software Engineering	50.0
Other	14.0
Not Applicable, Not Employed in CS	3.0

Table 9: Percentage of Registrants Working in Different Computer Science Employment Responsibility Areas

5.1 Signups

The first question we asked is where students heard about the course. These results are detailed in Table 10. Most students (over 60%) heard about the course via news releases the instructors made about the course's availability, such as online articles about the course offering or email listserv mailings. Other responses included professional associations advertising the course, social media postings about the course, and a colleague telling them about the course. The colleague category also includes word of mouth from students themselves.

We also asked students what general motivations they had for taking the course. These results are presented in Table 11, and students had the option of choosing as many categories for their motivation as they liked. The categories "General Interest in Topic", "For Personal Growth and Enrichment", and "Relevant to Job" all scored above 90% for students in the MOOC. "Relevant to Job" is interesting because it indicates we had a high percentage of software engineers enrolled in the course. Several of the categories were tangential to software security but still

Sources of Knowledge About the MOOC	*MOOC*
News Outlet	60.9
Professional Association	16.2
Social Media	14.3
Colleague	7.5
Other	1.1

Table 10: Percentage of registrants who heard about the software security MOOC from different sources.

Motivations for Enrolling in the MOOC	*MOOC*
General Interest in Topic	97.4
For Personal Growth and Enrichment	94.0
Relevant to Job	90.2
For Fun and Challenge	78.6
Share What I Learn with Colleagues	75.6
For Resources Applicable to My Practice of Software Security	71.4
Become a Better Coach or Mentor to Colleagues	63.2
Connect/Network with Other Software Security Professionals	56.4
Course Offered by Prestigious University/Professor(s)	55.3
Earn a Certificate/Statement of Accomplishment	51.1
Experience an Online Course	48.1
Take with Colleagues/Friends	35.7
Relevant to School or Degree	30.1
Relevant to Academic Research	21.8
For Career Change	21.8
For Incentives from Employer or Other Source (e.g., time off, financial, promotional)	18.8
To Improve My English Skills	10.5

Table 11: Motivation factors for MOOC Registrants

received a significant response rate. About half of the participants said taking an online course would be a valuable experience, while 10% of the students wanted the opportunity to work on their English skills.

5.2 Course Goals

In addition to looking at the student's motivations for signing up for the course, we also explored what the student was hoping to learn in the course. We presented the students with four pre-established learning goals for the course, as discussed in Section 3.1. We asked them how their goals shifted from what they expected at the beginning of the course. We asked students to rate the importance of these goals in order at the beginning before the class started (pre-course order) and after the class was completed (post-course order). The students' pre-course order and post-course order are found in Table 13. Both the on-campus students and MOOC students rated all of the goals in the same order before the course started and after it ended.

5.3 Discussion

For MOOC participants, we see a variety of reasons why students might enroll in the course. A diverse student body means diverse self-motivation goals for the course. While overall ordering of the four course-specific goals remained mostly the same before and after the course was run, individual students saw their goals shift as they took the course. Students in both the on-campus and MOOC versions of the course considered the goals to be at similar level of priorities. The first goal was the most important for both sets of students, while the second goal was the least important.

6. QUIZ AND TEST RESULTS (RQ2)

In this section, we discuss the results from the quizzes given in the on-campus course and the online course, and compare the performance of the two groups of students.

Table 13 shows all questions from the quizzes and tests that were common to both the on-campus course and the MOOC. There were 38 common questions to both courses that can be directly compared from a total of 216 questions asked across both courses. The questions were given to the on-campus students first, with the MOOC students getting the questions later. In some cases, there were issues of clarity or relevancy discovered after the question was given to the on-campus group. In those cases, the question was either reworded for the MOOC to make it clearer for students or completely rewritten for the MOOC. Questions that were reworded or completely rewritten were excluded from our analysis. Because the courses were not run in parallel, the current events questions in both courses were different. Finally, the on-campus course had attendance quizzes that were unnecessary for the MOOC. While the quizzes and tests had a direct effect on the on-campus students' grades, the MOOC students were graded on completion only, and were not receiving college credit for the course.

On-Campus	
Pre-Course Rank	*Post-Course Rank*
1. Security Risk Management	1. Security Risk Management
2. Security Requirements	2. Security Requirements
3. Secure Coding Techniques	3. Secure Coding Techniques
4. Security Testing	4. Security Testing
MOOC	
Pre-Course Rank	*Post-Course Rank*
1. Security Risk Management	1. Security Risk Management
2. Security Requirements	2. Security Requirements
3. Secure Coding Techniques	3. Secure Coding Techniques
4. Security Testing	4. Security Testing

Table 12: Pre-course order and post-course order of importance of goals for students in the on-campus course and the MOOC.

The mean of the difference between the on-campus results and the MOOC results is 20.7%, or what would be two letter grades at most universities. A t-test was performed on the two sets of correct answers. A 95% confidence interval was calculated for the difference in means at 11.9% to 29.6% difference. The p-value from the t-test was calculated as 0.0001, indicating that two groups are statistically significantly different.

7. MOOC STUDENT REFLECTIONS (RQ3)

Among the 59 MOOC students responding to the post-survey, 35.6% classified their participation as inactive, 35.6% as moderately active, and 28.8% as fully active. Students had the opportunity to expand on their answers in the survey in a short answer form. One student said: *"After I had signed up, my work pulled me to a different project and I didn't have time to complete the course at work, and my personal life was extremely busy then as well, so I didn't have time (or the energy) to complete it at home.*" The student responses highlights the importance of flexibility software engineers working through class material, as they may have additional demands on their time that university students do not.

In both classes, hands-on exercises with students performing exploits against dummy targets were well received by students. To quote one student in the MOOC, "*I really didn't know where to start when it came to performing exploits. Having a directed exercise where I could see the results of my actions really helped get me started.*" Additionally, after completion of the exercises, students in both sections reported that they had found vulnerabilities in websites they use every day. Some students felt overwhelmed by the exercises, while others felt the course would be improved with more complicated exercises.

For software engineers who only had time to work on the course on their work computer, exercises that were perceived to require administrator access on their computers was an issue. "*I shied away from the exercises that required I install stuff on my computer. Once employed and able to afford another computer and dedicate it to White Hat hacking exercises, I will be much more willing to download and do the exercises."* Exercises during the course made use of a Web Application Scanner Testing[13] site. Some exercises recommended students download a tool for use during the exercise, such as OWASP Zed Attack Proxy[14]. However, alternative ways to perform the exercises without the tool were provided.

Students in both courses were asked if the feedback provided by the course instructors was sufficient. The results of the agreement with these statements are presented in Table 14. The results are in the form of a 5-point Likert scale. Overall, students in the on-campus course thought that feedback was sufficient, while MOOC

[13] http://www.webscantest.com/

[14] https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

	On-Campus			MOOC		
Question	Answering	Correct	Incorrect	Answering	Correct	Incorrect
To practice a defense-in-depth strategy, where should input be checked?	111	98.2	1.8	111	97.3	2.7
An attacker can get rid of undesirable client side Javascript checks.	111	99.1	0.9	111	94.6	5.4
Which of these is the most desirable way to mitigate URL jumping vulnerabilities?	111	85.6	14.4	110	49.1	50.9
The preferred way to validate input to a program is through the use of a blacklist.	111	94.6	5.4	111	82.0	18.0
With a stored cross site scripting attack, the attacker-provided script is embedded in the web page generated by the server as an immediate response of an HTTP request. The client executes code in the context of the current user.	111	97.3	2.7	88	73.9	26.1
So many things to think about related to input filtering ... encoding, special characters, back slashes ... really, filtering with a white list is the way to go.	111	93.7	6.3	88	73.9	26.1
If you want to salt a hash, you can use an MD5 algorithm.	111	74.8	25.2	70	45.7	54.3
Adding https:// to the url will encrypt the connection and whatever page the url points to will be encrypted while in transit.	111	64.9	35.1	70	12.9	87.1
Google hacking is a means of hacking google's search engine.	111	99.1	0.9	73	86.3	13.7
It's important to have an index.html file in a directory.	111	90.1	9.9	73	32.9	67.1
Once authenticated, a user can be trusted.	111	100.0	0.0	57	98.2	1.8
Which of the following is an example of separation of privilege?	110	93.6	6.4	57	40.4	59.6
Historically, log files have been designed to enable forensics.	110	75.5	24.5	57	71.9	28.1
One means of preventing invalid redirects is to create a map between a value that can be used in a URL and a value in a mapping table.	109	100.0	0.0	60	91.7	8.3
What kind of attack are invalid indirects usually a part of?	109	75.2	24.8	60	33.3	66.7
Prudent strategies for protecting from vulnerable components include all of the following except which one?	108	100.0	0.0	62	88.7	11.3
What kind of access control model is in use for the following: "Only officers in the military can access top secret information."	105	91.4	8.6	55	43.6	56.4
Role-based access control is an example of one user, one role.	105	99.0	1.0	55	87.3	12.7
Which of the following is an example of a conflict in an access control model?	105	94.3	5.7	55	92.7	7.3
A missing function-level access attack can by done by manipulating a URL.	105	97.1	1.9	55	76.4	23.6
By default, no role should have access to any functionality. Functionality should be granted as needed.	105	98.1	1.9	55	89.1	10.9
One means of preventing a direct object reference is to use a mapping. An indirect reference map is a substitution of the internal (discoverable, especially sequential) reference with an alternate ID which can be safely exposed externally.	109	100.0	0.0	75	94.7	5.3
Static analysis tools can detect malvertising.	110	95.5	4.5	46	73.9	26.1
Security requirements should be integrated into the regular body of system requirements.	110	98.2	1.8	46	93.5	6.5
What type of security requirement is this: "All personally-identifiable information must be encrypted."	110	90.0	10.0	46	76.1	23.9
What type of security requirement is this: "Transactions that involve creating, reading, updating, or deleting personally-identifiable information must be logged."	110	96.4	3.6	46	56.5	43.5
Goal of an abuse case is to decide and document a priori how the software should react to illegitimate use	110	99.1	0.9	45	95.6	4.4
What relationship do you indicate on an arrow on use/abuse case diagram if you want to indicate the use case will make life harder for an attacker?	110	100.0	0.0	45	100.0	0.0
What relationship do you indicate on an arrow on use/abuse case diagram if the use case adds optional functionality to another use case but the other use case is complete without it?	110	100.0	0.0	44	97.7	2.3
What relationship do you indicate on an arrow on use/abuse case diagram if you want to indicate the use case will make life harder for a legitimate user?	110	97.3	2.7	45	86.7	13.3
A bank teller is able to change the value of his or her checking account balance. Which security property is not in question?	108	91.7	8.3	41	58.5	41.5
An intruder reads data in transit between two computers. This is a _____ threat on a _____.	108	99.1	0.9	41	87.8	12.2
Which of the follow diagram documents at least some mitigation strategies on the diagram itself?	108	90.7	9.3	41	29.3	70.7
A threat has been determined to be based a feature that is very easy to exploit and for which the impact of attack is very high. Which mitigation strategy is most appropriate, if possible?	108	73.1	26.9	41	36.6	63.4
The attack surface is all the ways an attacker can get into an application/program.	110	72.7	27.3	41	12.2	87.8
The main idea is ... for each new functionality added to a system to ask "will this new feature increase the attack surface" and to mitigate the risk, if possible.	110	100.0	0.0	41	97.6	2.4
A house with only one door has a smaller attack surface than one with two doors and many windows	110	100.0	0.0	41	97.6	2.4
Keeping the attack surface low is a risk mitigation strategy for zero-day attacks.	110	99.1	0.9	41	80.5	19.5
Average	109.3	92.7	7.2	60.5	72.0	28.0
Standard Deviation	1.9	9.6	9.6	21.7	26.1	26.1

Table 13: Number of responses and percentage of correct and incorrect responses to common questions in the on-campus and MOOC versions of the course

Prompts	*On-Campus* *Mean, Std Dev*	*MOOC* *Mean, Std Dev*	*Between Group Difference*
Feedback on activities is provided by the instructor(s) in the course	M=3.83, SD=.72	M=2.55, SD=1.06	t(39) = 6.298 p = .000*
The type of feedback participants can expect is clearly outline or explained	M=3.74, SD=.67	M=2.89, SD-.98	t(48) = 4.787 p = .000*
The overall feedback in the course was sufficient	M=3.80, SD=.81	M=3.05, SD=.97	t(133) = 4.549 p = .000

Table 14: Student response to statements on feedback received from course instructors (Likert scale)

students thought course feedback was insufficient. While weaker feedback for students in MOOCs may be expected considering the distributed nature of MOOCs and lack of personal communication, some of the result may be explained by technical difficulties with the course, which will be discussed in Section 8.

8. LESSONS LEARNED

In this section, we provide some of our lessons learned from our experiences offering a MOOC version of an on-campus course.

1. ***MOOC offerings are time consuming.*** We prepared to offer the MOOC by flipping the on-campus course. Nonetheless, the MOOC course still took a significant amount of time. The on-campus course ran for 15 weeks while the MOOC ran for 10 weeks. As a result, materials and quizzes had to be reorganized. Each week, we had to create a new, informal video in which we introduced the topics for the week and discussed current breaches in the press. Videos needed to be edited. Message boards needed to be monitored. Responding to individual questions from students on message boards and email was time consuming with limited impact for all students. Language barriers presented an issue in each form of communication and took time to resolve.

2. ***Peer evaluation is challenging.*** Due to the size of MOOCs, assessment of activities that could not be put in the form of multiple choice or true false is time prohibitive. As a result, many MOOCs utilize peer evaluation systems in which students grade other student's assignment. Reliable peer evaluation systems are hard to find. Another research group at North Carolina State University built the peer review system for both courses. We had numerous technical issues and resulted in a lot of frustration for students. We ended up scrapping the system. Better scalability testing for these types of systems is essential for MOOCs.

3. ***Multiple choice and true/false are limiting.*** Our only assessment vehicle was multiple choice and true/false questions. Not all course material fit well into that kind of assessment. Verification of students completing the exercises via the acquisition of a token or key may be one approach for verifiable exercises.

4. ***Support for the CourseBuilder platform in unreliable***. CourseBuilder turned off student tracking after we opened registration. Student tracking included both tracking student progress throughout the course and the scores from individual quizzes. After working with online support platforms on the problem, we discovered that we would have to launch the course again and have students reregister to turn tracking back on. Forcing students to register again was an unacceptable option to us, so we opted to look into alternatives to track student progress.

5. ***Google Forms quizzes had shortcomings.*** We used Google Forms for our quizzes and midterms. We were able to track student progress by setting up each quiz as a multiple response form and asking students to enter the email address they registered for the course on with each quiz. Using Google Forms had several consequences for course maintenance. One, the perceived quality of the course was lowered. Students disliked that they were not able to see their total scores. Second, Google Forms also caused a significant increase in work on the part of the instructors to manage course scores. Each quiz had to be manually parsed into an aggregate grades document.

6. ***Informal video messages from instructors are well received***. One well-received aspect of the course was the weekly security current events discussion we had each week. We felt they were especially useful in motivating software security in general, and we were fortunate in that news items happened to align with the topics we were teaching on occasion. Other courses – both MOOCs and in-person courses – could benefit from weekly current event discussions as well. Many STEM topics can be related to a recent news item.

7. ***Timely feedback on the message board is essential***. A member of the teaching stuff should be regularly checking the message board and responding to student questions and concerns. Some students in both sections commented that their learning could be aided by increased interactions between students and between students and instructors. Recommended student-instructor interaction strategies included more instructor participation in forums, synchronous webinars to discuss course topics, instructor availability via email, and enhanced feedback on quizzes.

8. ***Students will not dedicate much time***. The students who enrolled in the class indicated an interest in learning software security. We estimate that keeping up with the course might take 2-3 hours/week. Students dropped out of the course indicating they did not have the time available to complete the course each week.

9. ***Do not have strict time constraints***. We allowed students 6 weeks after the official course end to compete the course. Some students would not have finished if we had not provided flexibility.

9. LIMITATIONS

Results from the classroom setting are only from students who self-selected to provide results to the researchers. Students who opt to provide demographic information may represent a different population than the whole body of students.

MOOC students and classroom students may have different levels of motivation for correctly completing quizzes. Because the classroom students were in a computer science graduate program and the MOOC students are not, it would be reasonable to assume that the classroom students are more motivated to do well and receive good grades. To mitigate the grading issue, we told both sets of students that these quizzes would not be directly graded for credit, and were simply completion exercises. The MOOC

students were told the same thing. Quizzes are also not ideal for evaluating student understanding. While a consistent difference in scores is useful, it is not necessarily a direct indication of competency. Because of the format of the quizzes, students could have searched for answers, as we had no way of monitoring either group while the quiz was being taken.

10. FUTURE WORK

We identified several areas of improvement for future courses. Peer review, even before technical issues struck, represents a roadblock for participants in the course. Finding ways to mitigate peer review concerns will be important for more complicated activities that cannot be expressed in a quiz/midterm format. While we were not able to implement a Capture the Flag event for these courses, future iterations of courses in cybersecurity might consider having events like a Capture the Flag event to help students think like an adversary. As described in previous work, thinking like an adversary is important for professionals on the defensive side of security, so they know what their opposition is thinking and doing [20].

However, a full-scale Capture the Flag event may be infeasible for a MOOC because of the sheer number of students. To scale Capture the Flag to a MOOC, a simulated experience may be necessary. One example of a simulation of a Capture the Flag event would be to task students with securing a server hosted on a virtual machine. The server would then be "attacked" by a set of predetermined exploits generated by the instructors and teaching assistants. The simulation removes the need to pass around virtual machines to a wide range of students, though students who are taking the course at work may still struggle with the simulation approach. New hands-on activities are needed in order to provide real experience for students in work environments where individuals have to conform to the business information technology policy.

11. ACKNOWLEDGMENTS AND NOTES

Thanks to the CSC710 class at North Carolina State University for their feedback throughout the process of developing this research. Thanks to the students of both courses for signing up and participating in the class. Thanks to the Realsearch group for their feedback on this paper. The work in this paper was funded under National Science Foundation grant number 4900-1318428.

12. REFERENCES

[1] M. Y. Vardi, "Will MOOCs destroy academia?," *Communications of ACM,* vol. 55, no. 11, p. 5, November 2012

[2] P. Hyman, "In the Year of Disruptive Education," *Communications of ACM,* vol. 55, no. 12, pp. 20-22, December 2012.

[3] Aiken, J. M., Lin, S. Y., Schatz, M. F., & Caballero, M. D. (2013). The Initial State of Students Taking an Introductory Physics MOOC. *arXiv preprint arXiv:1307.2533*.

[4] Belanger, Y., & Thornton, J. (2013). Bioelectricity: A Quantitative Approach Duke University's First MOOC. "http://dukespace.lib.duke.edu/dspace/handle/10161/6216"

[5] Kop, Rita. "Information aggregation in networked learning: The huma

[6] n factor and serendipity." *8th Int. Conf. on Networked Learning, Maastricht, The Netherlands*. 2012.

[7] Kop, Rita. "The challenges to connectivist learning on open online networks: Learning experiences during a massive open online course." *The International Review of Research in Open and Distance Learning* 12.3 (2011): 19-38.

[8] Bruff, Derek. "Lessons Learned from Vanderbilt's First MOOCs." 2013. https://cft.vanderbilt.edu/2013/08/lessons-learned-from-vanderbilts-first-moocs/

[9] Fournier, Hélène, Rita Kop, and Hanan Sitlia. "The value of learning analytics to networked learning on a personal learning environment." In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge (LAK '11)*.

[10] Center for Teaching - Vanderbilt University, "The Coursera Resource Guide," 2012. https://cft.vanderbilt.edu/wp-content/uploads/sites/59/coursera.pdf

[11] Littlejohn, Allison, "Understanding Massive Open Online Courses," *CEMCA EdTech Notes*, 2013

[12] Pardos, Zachary A., and Emily Schneider. "First Annual Workshop on Massive Open Online Courses." *Artificial Intelligence in Education*. Springer Berlin Heidelberg, 2013.

[13] de Waard, Inge, et al. "Exploring the MOOC format as a pedagogical approach for mLearning." *Proceedings from mLearn* (2011).

[14] M. Prince, "Does Active Learning Work? A Review of the Research," *Journal of Engineering Education,* vol. 93, no. 3, pp. 223-231, 2004.

[15] R. M. Felder, D. R. Woods, J. E. Stice, and A. Rugarcia, "The Future of Engineering Education, Part 2: Teaching Methods that Work," *Chemical Engineering Education,* vol. 34, no. 1, pp. 28-39, 2000.

[16] Berrett, Dan. "How 'flipping' the classroom can improve the traditional lecture." *The chronicle of higher education* 12 (2012).

[17] Rivard, Ry. "Measuring the MOOC dropout rate." *Inside Higher Ed* 8 (2013).

[18] Cabiria, J. "Connectivist learning environments: Massive open online courses."The 2012 World Congress in Computer Science Computer Engineering and Applied Computing. 2012.

[19] "The Cisco 2014 Annual Security Report." 2014 - http://www.cisco.com/web/offers/lp/2014-annual-security-report/index.html

[20] M. Dark, and J. Mirkovic. "Evaluation Theory and Practice Applied to Cybersecurity Education." *IEEE Security & Privacy* 2 (2015): 75-80.

[21] Mirkovic, J., Dark, M., Du, W., Vigna, G., & Denning, T. (2015). Evaluating Cybersecurity Education Interventions: Three Case Studies. *Security & Privacy, IEEE*, *13*(3), 63-69.

[22] Kay, J., Reimann, P., Diebold, E., Kummerfeld, B., "MOOCs: So Many Learners, So Much Potential ...," in *Intelligent Systems, IEEE* , vol.28, no.3, pp.70-77, May-June 2013

[23] Liyanagunawardena, T. R., Adams, A. A., & Williams, S. A. (2013). MOOCs: A systematic study of the published literature 2008-2012. *The International Review of Research in Open and Distributed Learning*, *14*(3), 202-227.