7 Surprising Lessons Learned From Teaching iOS Programming To 30,000+ MOOC Students

Parham Aarabi
Parham@ecf.utoronto.ca
Narges Norouzi
nnorouzi@comm.utoronto.ca
Jack Wu
Jackwu95@gmail.com
Michael Spears
mike.spears@utoronto.ca
University of Toronto

Abstract - **In this paper, we experimentally explore the impact of different teaching paradigms on teaching a large-scale iOS programming MOOC consisting of 30,162 students. Our initial approach utilizes methods from our in-person lecturing experience. After launching the initial version of the course, we analyzed our performance and student feedback based on which we recreated and re-launched the entire course with a particular focus on clarity, video-quality, and packaging the topics in 1-2 minute micro-modules. Based on feedback from 650 students, we observed that the overall lecture positive feedback increased from 65% before our course adjustment to 83% after the adjustment. In this paper we will provide a detailed overview of the lessons learned and the impact of our new teaching methods on each section of the course.**
We also noticed a slight increase in course completion rates, from 5.4% before the adjustment to 5.8% after the adjustment.

Index Terms – MOOC, Swift Programming, e-Learning

1. INTRODUCTION

Massive Open Online Courses (MOOC), especially in the context of programming and app development, has been of increasing popularity in recent years [1,2,8,9]. Although MOOCs and regular university lectures have many elements in common, including similar goes, different studies have shown that preparing a MOOC is very different and sometimes harder than preparing an ordinary university lecture-based course [3,5]. Many different recent studies have focused on methods to enhance student experience in MOOCs, including the need to focus more than ordinary university lectures as well as the need to maximize the video quality [5,6]. One of the challenges mentioned by [6] was that the very broad background, language, and preparation levels of students increased the difficulty in delivering the MOOC effectively.

In July 2015, our team at the University of Toronto was selected to teach a 4-course specialization on iOS app development with the goal of teaching students from different age groups and background to learn to build iOS apps. Having limited prior experience in teaching a large-scale MOOC, our immediate approach was to base our courses on in-person university-lecture based courses that we had taught previously (with several of these lessons having been discussed in [7]). The major highlights of our initial methodology consisted of the following:

1. When discussing a topic, explain the story and rationale behind the topic to get the students interested.
2. When teaching students, make sure they understand how to solve problems rather than just giving them the answer.
3. Create content, jokes, and social references for a 19-21 audience.
4. Create content by assuming the attention span of a typical student is about 20-25 minutes.
5. Establish very difficult tests and quizzes to challenge and push students.
6. Assume that students have a high-bandwidth desktop computer at home, which is the primary device with which they watch online content.
7. Assume that our guide (and competition) for teaching is a textbook. Hence the pace for the lectures would be similar to following a book.

After our first course titled "Introduction to Swift Programming" was launched, we observed significant feedback about the content, especially about the videotaped lectures for the course. It became clear based on a detailed evaluation of the student feedback that our teaching methodologies that were based on years of teaching

University lectures were invalid for a large-scale MOOC. The following section illustrates the key lessons learned as a result of our self-evaluation, based on which the course was re-done (including all of the lecture content). In section 3 we will provide quantitative evaluation of the two courses, including a breakdown of feedback on individual lecture topics.

2. LESSONS LEARNED

As we learned after our initial course had launched, teaching large-scale MOOCs is very different than teaching an ordinary university-lecture based course. The primary differences arise from the different modality in which students listen to lectures, which limits their attention span, and places a more significant burden on the instructors to maximize content clarity and video quality.

2.1. Avoid Storytelling and Laser Focus On A Single Specific Learning Objective

In-person lectures are often enhanced by anecdotes about a particular topic, stories about how certain ideas came to be, and generally a show that centers on the subject matter but provides a thorough discussion of other related elements. This, in the context of our MOOC, did not work. What did work was a singularly focused presentation of a subject matter with little context or storytelling. In fact, anecdotal stories where almost unanimously disapproved by the students, who complained that their limited time was wasted by content which was not central to the topic at hand. In the second version of the course, every minute of each lecture was spent focusing on a single small learning objective, which worked well in the context of an iOS app development course. Successive modules would together combine to convey a larger message, but this was more a subtle effect rather than an obvious teaching goal. In practice, this worked very well, and was the primary reason for improved feedback from students.

2.2. Answer Key Questions Instead of Teaching Problem Solving

One of the things we always emphasize in our regular university lectures is the need for developing problem solving skills instead of just jumping to the solution. Problem solving skills remain with a student (hopefully) for life, while solutions to particular questions become less relevant when questions (i.e. programming languages) change. An example of teaching problem solving that we sometimes employ during regular lectures is to start with a program with errors, and sequentially find out the errors and explain the reason behind the errors and the correct solution. It was quite surprising to find out that our MOOC students almost unanimously disliked the same. Upon exploring this further, we realized that with online courses students care far more about the specific solution than about how to get there. Although we understand why this would be the case (i.e. if you are busy, with limited time, you care less about methodology than you do about getting what you need), we still fundamentally believe that it is the methodology that should be taught.
However, to address the needs of the students while balancing our own beliefs, we abandoned the find-the-errors method of teaching programming and instead focused on well defined learning objectives (about 1 minute of content), with a very short and focused discussion on the methodology behind the topics discussed (about 30 seconds), followed by an optional (i.e. ungraded) expanded discussion of the methodology and background.

2.3. Create Content For An 18-60 Group, Not Just 20 Year Olds

One of the things that we underestimated was the difficulty in teaching a very broad audience with different backgrounds and different age groups. This realization is similar to that of [6]. In our regular university lectures, our students are often the same age (i.e. 18-25), they think about similar problems (i.e. getting good grades), have similar commitments (i.e. other courses), and are generally on the same wavelength when it comes to content, entertainment, goals, etc. However, when the audience becomes broader, what is interesting to one group can often be boring to another group. As a result, you either end up alienating one group (i.e. if they don't get a joke), or you end up removing any content that would not have a broad target. For our MOOC, over 20% of the students enrolled were over 45 years old, and over 6% were over 55. This created a very different dynamic than what we were used to.
In the second iteration of our course, most of the jokes, cultural references, and other content meant to be entertaining were removed as a result of direct student feedback. It is interesting to note that the exact same content that was negatively viewed by the MOOC students was often positively viewed by our regular university students. Although the removal of this content reduced the “fun” and entertaining aspect of our lectures, the more concise and clearer discussions did result in a significantly higher positive feedback from students.

2.4. Take The Attention Span Of Typical University Students, Then Divide It By 20

One of our observations over the years is that the typical university student has an attention span of about 20-25 minutes. In other words, you can discuss a topic for that amount of time before you start to lose the interest of the audience, as a result, each discussion topic needs to come to a conclusion and finale within that timespan. For an online course, it is now our view that the attention span is about 1-2 minutes. Essentially, content needs to be broken down into short 1-2 minute learning micro-modules. Each micro-module is essentially a self-contained online video that students can independently listen to and learn without requiring any previous content.

One of the key changes in our second iteration of the course was our focus on these 1-2 minute micro-modules, which based on the feedback from the students, helped in increasing the digestibility of the course for an online audience.

2.5. Make Sure To Get Validation Just Right (i.e. Goldilocks Validation)

In our regular university courses, tests, quizzes, and other validation mechanisms are used as a tool to motivate students to learn the topics discussed and to push them to use all of their intelligence to solve problems (including topics and problem solving skills that were not discussed in our specific course). Having very difficult and/or unorthodox tests has proven to be a useful means for improving student performance, especially for the top-tier students.

Testing in the context of an online course serves an entirely different purpose. Students look to these quizzes as a means of actually validating to themselves that they have mastered a particular subject matter. Tests and quizzes that are extremely difficult make many of the students (including some good students) believe that they are unable to learn a subject matter. Whereas in a regular university course students are forced to be along for the ride, our observation is that online students tend to be quicker to decide that a subject matter is not right for them. On the other hand, tests and quizzes that are too easy have the opposite effect and make unfit students incorrectly believe that they are a good match for the course. The latter may be good for increasing course revenues, but ultimately is a failure of the instructors if we cannot help students gauge their performance in a realistic fashion.

As a result, it is very important for MOOC instructors to set the validation elements of the course at a correct level to appropriately convey to the students their ability to learn a subject matter. One last note regarding validation is that in university lectures the question that we as instructors ask ourselves is "are these students good enough to get a degree from our university". In a MOOC the question is "are these students good enough to learn a particular topic". The answer to these questions may not necessarily be identical, and this impacts the type and difficulty of the validation that is used in each type of course. In this paper, we left out the impact of validation and have focused solely on the lecture feedback of the students.

2.6. Remember That Your Lecture is Watched On Small Mobile Devices, With Limited Bandwidth, In Loud Environments

When creating the videos for our initial course, we focused on a user working on a desktop computer watching our video while trying the discussed ideas in Xcode (the iOS development platform) at the same time. While this did happen with some of our students, there was a significant number (perhaps even the majority) who watched our videos on mobile devices. As a result, there were a significant number of students who could not make out the detailed programming notes and steps on a small mobile screen.

In the second iteration of our course we focused primarily on our mobile users, using extremely large fonts, split-screens, and other video focusing methods to maximize use of a small mobile display. Similarly, while our audio recordings were okay for a quiet desktop environment, they did not initially have the clarity for someone listening on a bus or in a busy coffee shop - which again was the setting for a large portion of our students. In the second iteration of the course, the audio was to maximize the volume and clarity.

2.7. You Are Competing With Google – So Be Efficient

Giving a regular in-person university lecture in many ways is a competition between the lecturer and a textbook. What the lecturer says repeats what the book covers, perhaps with less detail and examples, but with more focus. As a result, we as lecturers have taught ourselves to gauge our lecturing performance, pace, and direction usually by aligning it to a particular textbook.

When students take a university degree their goal is be enlightened with new experiences and knowledge that will guide them throughout their careers. When people search for keywords online, they simply want the most accurate and direct answer to a question. As university lecturers, our role is clear in the physical university context just as the role of a search engine is clear in the context of an online search. Online courses fall somewhere in-between these two disparate contexts. Students here want to learn, but they want to learn something very focused with little care about the contexts or stories related to the subject matter.

In this regards, our competition is really Google or similar search engines. Students want to know specific items, which they could just as easily search for online. The speed and efficiency of search engines creates a different level of expectation from online MOOCs than that of regular in-person university lectures.

2.8. Putting It All Together

Although the above lessons were interesting for a variety of different reasons, we did not obey these lessons blindly during the creation of the second version of our course. Instead, we considered these items more carefully during each lecture. For example, context around a particular programming idea was reduced, but not necessarily removed entirely. Or, as another example, students were provided specific explanations about a learning objective but there were further readings and auxiliary lectures that, if and when they had time, would teach them about the problem solving and programming methodology behind the specific learning objective.

When all of these items were put together, our substantially improved second iteration of the course was launched. The following section examines the differences in quantitative student feedback between the two courses.

3. Student Feedback

Both iterations of our course consisted of 16 lectures which covered various topics on Swift-based iOS programming. The first version of the course was based only on our university lecturing experience, and the second version included all of the feedback and lessons that were explained above.

Figure 1 demonstrates the ratio of positive user ratings for each of the lectures in the two versions of the course. As shown, the ratio of positives votes is almost always higher in the second course. Based on a total of 650 student votes in both of the MOOC courses, we observed that the overall lecture feedback has increased from 65% positive ratings before for version 1 of the course to 83% positive ratings for the second version.

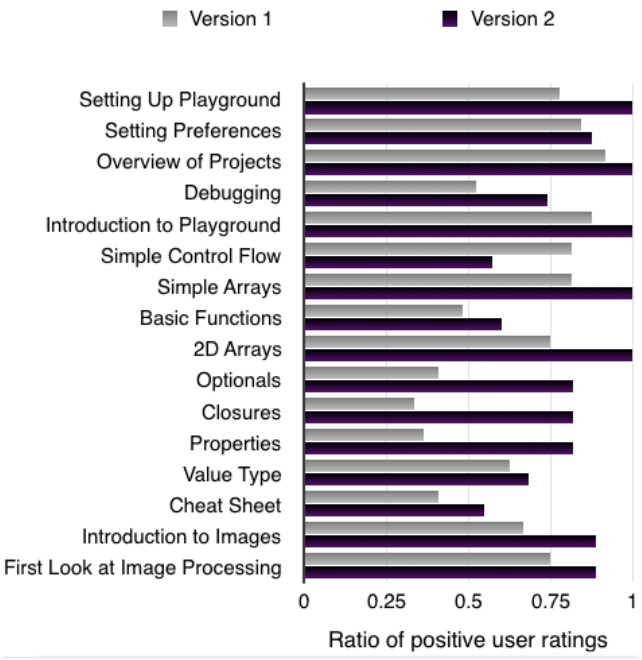

Figure 1. Ratio of positive user ratings in 16 different lectures in two versions of the MOOC

An interesting observation in the figure above is that in the first version of the course, as students would get to the more involved topics (such as Optionals, Closures, and Properties), the feedback becomes very negative. However, in the second version the positive feedback ratio remains fairly high at 80%. This is most likely attributed to the more focused micro-module method of content delivery, which appears to have worked well for the more complex topics.

3. Conclusion

In this paper, we experimentally explored the impact of different teaching paradigms on teaching a large-scale iOS programming MOOC consisting of 30,162 students. Our initial approach utilized methods from our in-person lecturing experience. After launching the initial version of the course, we analyzed our performance and the student data and recreated the entire course and launched a second, paying particular attention to clarity, video-quality, and packaging content in 1-2 minute micro-modules. Based on feedback from 650 students, we observed that the overall lecture positive feedback increased from 65% before our course adjustment to 83% after the adjustment.

References

[1] Kolowich, Steve. "The MOOC survivors." *Insider Higher Ed* (2012).

[2] Ardis, Mark A., and Peter B. Henderson. "Software engineering education (SEEd): is software engineering ready for MOOCs?." *ACM SIGSOFT Software Engineering Notes* 37.5 (2012): 14-14.

[3] Johnson, Don H. "Teaching a "MOOC:" Experiences from the front line."*Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), 2013 IEEE*. IEEE, 2013.

[4] Mazoue, James G. "The MOOC model: Challenging traditional education." (2014).

[5] Guo, Philip J., Juho Kim, and Rob Rubin. "How video production affects student engagement: An empirical study of mooc videos." *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 2014.

[6] DeBoer, Jennifer, et al. "Diversity in MOOC students' backgrounds and behaviors in relationship to performance in 6.002 x." *Proceedings of the Sixth Learning International Networks Consortium Conference*. 2013.

[7] Aarabi, P., The Art of Lecturing, Cambridge University Press.

[8] Haywood, Jeff, Amy Woodgate, and David Dewhurst. "REFLECTIONS OF AN EARLY MOOC PROVIDER." *MOOCs and Open Education Around the World* (2015): 89.

[9] Porter, David. "MOOC on Mobiles for Development Report." (2014).

[10] **https://www.coursera.org/learn/swift-programming/**

Author Information

Parham Aarabi, Department of Electrical and Computer Engineering, University of Toronto, Parham@ecf.utoronto.ca

Narges Norouzi, Department of Electrical and Computer Engineering, University of Toronto, nnorouzi@comm.utoronto.ca

Jack Wu, ModiFace Inc., Jackwu95@gmail.com

Michael Spears, University of Toronto, mailto:mike.spears@utoronto.ca