

# Producing and Delivering a MOOC on Pattern-Oriented Software Architecture for Concurrent and Networked Software

Douglas C. Schmidt and Zach McCormick

Vanderbilt University, Institute for Software Integrated Systems, Nashville, Tennessee  
{d.schmidt,zach.mccormick}@vanderbilt.edu

## Abstract

A massive open online course (MOOC) is a web-based class environment aimed at large-scale global participation and open access via the Internet. MOOCs are also a disruptive trend changing how education is delivered and funded throughout the world. In the spring of 2013, we developed and taught Vanderbilt's first MOOC, entitled "Pattern-Oriented Software Architecture for Concurrent and Networked Software" (known as the POSA MOOC). This ten-week MOOC was an amalgamation of several courses on software design and programming taken by ~600 undergraduate and graduate students at Vanderbilt during the past decade. Enrollment in our POSA MOOC was more than 50 times (31,000+) that number, consisting of students with a wide range of background, interests, and expertise from scores of countries around the world.

This paper describes our experiences producing and delivering the POSA MOOC. Where possible, we ground our observations in data from statistics collected via Coursera, which was the delivery platform we used for the POSA MOOC. We also discuss the broader implications of MOOCs on life-long learning and the role they play in improving the quality and productivity of software professionals in academia and industry.

**Categories and Subject Descriptors** K.3.1 [Computer Uses in Education]: Distance learning; K.3.1 [Computer Uses in Education]: Collaborative learning

**Keywords** MOOCs; Coursera; pattern-oriented software architectures and frameworks; object-oriented design and programming

## 1. Introduction

Vanderbilt has been a respected institution of higher education since 1873, graduating over one hundred thousand students during the past 140 years. In a span of the past 8 months, however, Vanderbilt has more than doubled the number of students taught by its faculty. The source of this surge in exposure stems from the "Massive Open Online Courses" (MOOCs) that Vanderbilt began offering through Coursera in March of 2013 (see [www.coursera.org/vanderbilt](http://www.coursera.org/vanderbilt) for the list of Vanderbilt MOOCs).

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

SPLASH '13, October 26–31, 2013, Indianapolis, Indiana, USA.  
Copyright © 2013 ACM 978-1-4503-1995-9/13/10...\$15.00.  
<http://dx.doi.org/10.1145/2508075.2508465>

The MOOC we taught on the Coursera platform was called "Pattern-Oriented Software Architecture for Concurrent and Networked Software," which we refer to as the POSA MOOC (see [www.coursera.org/course/posa](http://www.coursera.org/course/posa) for access to this material). This MOOC showed by example how applying object-oriented patterns and frameworks can help to alleviate many accidental and inherent complexities associated with developing and deploying concurrent and network software. The patterns and frameworks covered in this MOOC have been used successfully in many domains, including telecom and datacom, mobile devices, electronic medical imaging, network management, aerospace aviation and automation, as well as online gaming, web services, and financial systems.

By the time the POSA MOOC launched on March 4th, 2013, 31,000+ students were enrolled, hailing from a wide range of countries, as shown on the heat map in Figure 1. During the ten



Figure 1. Global POSA MOOC Student Locations

weeks of the POSA MOOC, a subset of these students accessed the on-line video lectures 464,498 times and attempted the on-line quizzes 37,817 times. Many students expended significant effort to submit 13,220 assignments—written in 6 different programming languages—and conduct 45,649 unique peer-graded assessments of these assignments. Moreover, conversations between students and course staff on the discussion forums numbered well over 7,000 unique posts, providing a highly interactive (albeit time-consuming to monitor and manage) virtual learning community.

Producing and delivering a MOOC at this scale was much different from the courses we've traditionally taught at Vanderbilt. Imagine teaching a course where the students could have the prerequisite background (or not), join the class (or not) at any time, listen to the lectures (or not) at any time, take the quizzes (or not) at any time, do the programming assignments (or not) at (almost) any time, read the archives of past discussions (or not — usually not by the way) prior to posting their questions, etc. This summary captures just part of what it's like teaching a MOOC. In addition to being a non-linear—often hectic—adventure, it's also a fasci-

nating experiment in the democratization of learning, as well as a harbinger of things to come in higher education.

Differences between MOOCs and traditional face-to-face classes at Vanderbilt profoundly affected the preparation, presentation, and assessment of the POSA MOOC material. For example, the students we taught in the POSA MOOC had much more diverse backgrounds, interests, and expertise than traditional Vanderbilt undergraduates, which impacted both our teaching style and student learning experiences. One example of this diversity in the POSA MOOC is shown in Figure 2, which depicts enrollment by age based on a voluntary survey completed by a subset of the enrolled students. These demographics look nothing like the typical

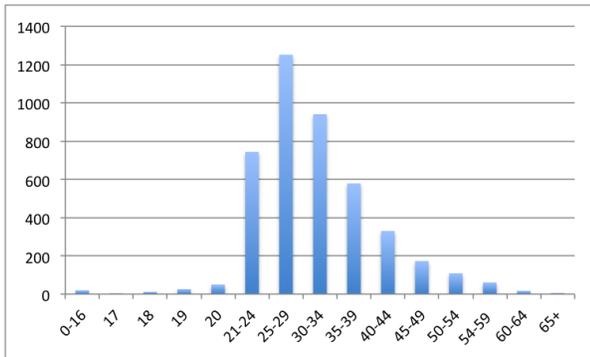


Figure 2. POSA MOOC Enrollment by Age

age profile at Vanderbilt, where the vast majority of students are 18-22 years old. Moreover, there were significant challenges associated with assessing student performance in “design-oriented” MOOCs (such as POSA) versus “fact-oriented” MOOCs (such as the Algebra or Pre-Calculus) on Coursera.

A common criticism [11] of conventional MOOCs is that they dehumanize education by neglecting or degrading interactions amongst students and teachers. There’s certainly no substitute for face-to-face engagement between motivated teachers and students. We applied several innovative techniques and social media tools in our POSA MOOC, however, that enabled meaningful dialogue between students and the course staff, which helped ameliorate some noted deficiencies with conventional MOOC offerings.

The remainder of this paper is organized as follows: Section 2 summarizes the contents and structure of the POSA MOOC, emphasizing the changes we made to our face-to-face classes to handle the lack of prerequisites in Coursera MOOC offerings; Section 3 discusses our observations and lessons learned creating and presenting the POSA MOOC; Section 4 discusses our experience with the impact—both pro and con—that MOOCs are having on traditional on-campus education; and Section 5 presents concluding remarks and outlines our future plans for the POSA MOOC.

## 2. Structure and Contents of the POSA MOOC

This section describes the POSA MOOC structure and contents.

### 2.1 Summary of the POSA MOOC

Our MOOC is motivated by the advent of multi-core and distributed core processors—coupled with ubiquitous wired and wireless connectivity—which is increasing the demand for researchers and practitioners who understand how to successfully develop and deploy concurrent and network software. Despite continuous improvements in processors and networks over the past four decades, however, developing quality concurrent and network software re-

mains hard. Moreover, developing quality *reusable* concurrent and network software is even harder.

The principles, methods and skills required to develop such software can be greatly enhanced by understanding how to create and apply *patterns* [3] and *frameworks* [7]. A pattern describes a reusable solution to a common problem that arises within a particular context. When related patterns are woven together, they form *pattern languages* [2] that provide a vocabulary and a process for the orderly resolution of software development problems.

A framework is an integrated set of software components that collaborate to provide a reusable architecture for a family of related applications. Frameworks can also be viewed as realizations of pattern languages that facilitate reuse of detailed design and source code. The POSA MOOC described how to apply patterns and frameworks to alleviate many accidental and inherent complexities associated with developing and deploying concurrent and network software in multiple domains, including mobile applications, web servers, object request brokers, and avionics control systems.

### 2.2 Summary of the Video Lectures

The students we teach in our courses on patterns and frameworks at Vanderbilt have the necessary background. For example, our *Intermediate Software Design* course ([www.dre.vanderbilt.edu/~schmidt/cs251](http://www.dre.vanderbilt.edu/~schmidt/cs251)) focuses on object-oriented design patterns and advanced object-oriented programming techniques with C++. Students taking this course must have successfully completed introductory courses on programming and data structures, so we know they are familiar with core object-oriented programming features (such as classes, inheritance, and dynamic binding) and object-oriented programming languages (such as Java and C++). In contrast, when developing and teaching the POSA MOOC we had no idea how well/poorly prepared the students would be since there are no prerequisites in the Coursera curriculum.

We dealt with the lack of prerequisites by structuring the POSA MOOC into multiple sections. While the overall focus of the MOOC was on patterns and frameworks for concurrent and networked software, we added several introductory sections that covered the background material needed to understand core concepts in concurrency, networking, patterns, and frameworks, as well as an optional section that covered core object-oriented design, programming, and pattern concepts. The following is a summary of the topics we covered in all these sections.

#### 2.2.1 Section Zero: Course Overview

This hour of introductory videos was designed to help students visualize the motivations for—and challenges of—concurrent and networked software. We also summarized how patterns and frameworks help to address key challenges of software, in general, as well as concurrent and networked software, in particular.

#### 2.2.2 Section One: An Introduction to Concurrent and Networked Software

This section contained 3.5 hours of videos that provided background information related to operating systems and middleware. We discussed key design dimensions of concurrent and networked software (such as principles for partitioning systems into multiple layers and services), as well as reviewed common UNIX and Windows operating system programming mechanisms and Android programming mechanisms (which figured prominently in Section Two of the POSA MOOC).

Section One emphasized concepts, so there wasn’t much discussion of patterns, frameworks, or code. Our goal was to summarize the material students needed to understand the topics covered in later sections. At Vanderbilt this material would have been covered

in earlier courses, but since the Coursera curriculum enforces no prerequisites we filmed these introductory videos.

### 2.2.3 Section Two: An Introduction to Patterns and Frameworks

This section had 6 hours of videos that delved deeper into the POSA MOOC's main emphasis: *patterns and frameworks for concurrent and networked software*. This section focused largely on design rather than programming, with many structural and behavior elements of patterns and frameworks conveyed via UML diagrams. Although there was some example code in Java, C++, and C, expertise in these programming languages wasn't needed to understand the material in this section.

We began with an overview of patterns and frameworks in general, emphasizing key concepts, such as codifying design experience, enabling systematic reuse, and combining groups of related patterns to define a process for the orderly resolution of software development problems in particular domains. We outlined several examples of common concurrent and networked programming patterns (such as Proxy, Broker, Observer, and Command Processor) and frameworks (such as Android [10], ACE [14], and TAO [15]) written in Java and C++.

We also discussed the pros and cons of patterns and frameworks, when to use them and avoid them, and what alternatives to consider if they don't work in particular contexts. In addition, we summarized additional reference material on patterns and frameworks to guide students interested in learning more about these topics than we covered in the POSA MOOC.

### 2.2.4 Section Three: Applying Patterns and Frameworks to Concurrent and Networked Software

This section contained 6 hours of videos and had the most technical depth of the POSA MOOC. It focused on how to develop concurrent and networked software by applying patterns and frameworks and grouping patterns into pattern languages. To make the examples in this section concrete and relevant, we chose a case study from the domain of high-performance web servers, based on the JAWS [6] open-source web server developed in C++ using many patterns and ACE framework components. There were numerous C++ code examples in this section, so students needed a solid grounding in C++ (or an equivalent object-oriented language like Java or C#) to understand the examples.

The patterns and frameworks covered in this section covered a range of concurrent and networked software capabilities, including service access and configuration, inter-process communication, synchronous and asynchronous event handling, concurrency, and synchronization. Most patterns in this section were based on the pattern language in the *Pattern Oriented Software Architecture Volume Two* book [13], which covers patterns for concurrent and distributed objects. We also discussed how patterns from the book *Design Patterns: Elements of Reusable Object-Oriented Software* (the so-called "Gang of Four" book) [3] help simplify certain design and programming aspects of concurrent and networked software.

### 2.2.5 Section Four: A Case Study of "Gang of Four" Patterns

This optional (*i.e.*, no quizzes or peer-graded assignments) section provided 3.5 hours of background videos on object-oriented design and patterns that weren't directly relevant to concurrent and networked software, but which are essential to becoming an effective developer of object-oriented programs. It was organized around a case study that applied over half of the 23 patterns in "Gang of Four" book to showcase a pattern- and object-oriented design and programming techniques using C++. This case study enabled students to learn and evaluate the limitations with alternative software development methods (such as algorithm decomposition) and

demonstrate by example how patterns and object-orientation help to alleviate these limitations.

## 2.3 Student Assessment Mechanisms

To obtain approval from Coursera and Vanderbilt to launch the POSA MOOC, we created several methods for assessing the student performance. Moreover, we offered the course at the following two levels of engagement in recognition of the fact that not all participants have the same learning objectives or available time:

- **Normal Track.** Students at this level received a *Statement of Accomplishment* that certified proficiency with the course concepts, which we assessed via weekly auto-graded quizzes. This track was designed for students who had time/interest in taking the auto-graded quizzes and final exam, but who did not have time/interest to complete the peer-graded short essay questions and peer-graded programming assignments.

Students could fulfill the Normal Track without joining when the MOOC started, as long as they completed all auto-graded quizzes and final exam by the time the POSA MOOC ended after ten weeks. The final grade for the Normal Track was based on the weekly quizzes (90% of the final grade) and a final exam (10% of the final grade). Students who obtained 70% or more of the maximum score received a *Statement of Accomplishment*.

- **Distinction Track.** Students at this level received a *Statement of Accomplishment with Distinction*. In addition to completing the auto-graded weekly quizzes and final exam from the Normal Track, students in the Distinction Track also completed peer-graded short essays and peer-graded programming assignments. The programming assignments involved writing concurrent and networked software in popular pattern-oriented software architecture frameworks written in a range of languages—including Java, C++ (and C++11), C#, Python, Ruby, and Scala—using a variety of production object-oriented frameworks—including Netty (Java); Twisted (Python); and Qt, Boost, and ACE (all C++). This track was designed for students willing to invest the time to achieve mastery of the course material and apply it in structured assignments.

To fulfill the Distinction Track students needed to complete the various peer-graded assignments by their due dates (which were typically two or three weeks after the assignments were initially released on the Coursera platform). The final grade for the Distinction Track was based on the weekly quizzes (35% of the final grade), peer-graded short essays and peer-graded assignments (55% of the final grade) and a final exam (10% of the final grade). Students who obtained 70% or more of the maximum score received a *Statement of Accomplishment with Distinction*.

## 3. Observations and Lessons Learned

Although we were institutionally responsible for teaching thousands of students, by the time the POSA MOOC finished we felt like we'd learned at least as much as we'd taught. This section describes some of our observations and lessons learned while producing and delivering our POSA MOOC.

### 3.1 An Enormous Amount of Time was Needed to Prepare the Content Prior to MOOC Launch

Filming videos used by thousands of students as their primary exposure to the POSA MOOC content required considerably more preparation than we were accustomed to based on face-to-face courses we've taught at Vanderbilt and other universities. Below we discuss several reasons for the concentrated effort.

#### 3.1.1 Filming High Quality Video Lectures

After teaching courses on object-oriented software patterns and frameworks for two decades, it's become second nature to present

lively and inspiring lectures despite minimal rehearsal and *ad hoc* slides. This haphazard model of preparation *does not* work in a MOOC since there are no students to interact with while filming the videos in the studio. As a result, we needed to produce much tighter scripts and highly structured lecture material.

Each week of the POSA MOOC featured several hours of video lectures from one or more of the *sections* summarized in Section 2. Each section was similar to a volume in a multi-volume book series and was composed of multiple *modules*, which were akin to chapters in a book. Each module was composed of multiple *parts*, which were akin to a portion of a book chapter and were roughly 15–20 minutes long (the length Coursera recommends to keep the attention of students). Every five minutes or so, a multiple-choice “in-video” quiz popped up to check whether the student viewing the video understood the material covered thus far.

It took us two solid months of filming to produce 80+ individual videos that ran for ~20 hours. In contrast, ~40 hours are spent lecturing in a conventional semester-long Vanderbilt class (and the preparation time for these lectures is *much* lower since there’s more opportunity for improvisation). Many POSA MOOC videos were filmed using advanced *green screen* technology (see [en.wikipedia.org/wiki/Chroma\\_key](http://en.wikipedia.org/wiki/Chroma_key)), which provided maximal flexibility in rendering the instructor in front of various backgrounds, which were mostly Powerpoint presentation slides.

We prepared 1,200+ Powerpoint slides for the POSA MOOC videos. Many slides were new since we had to convey the material without the benefit of interactive dialogue from students in a face-to-face class, so we made the slides more explanatory than normal. Moreover, explicit permission was needed to use copyrighted images in MOOC videos, so we replaced copyrighted images in our original slides with Creative Commons licensed images that didn’t require explicit permission to use.

Even with 1,200+ slides and 20+ hours of videos, the POSA MOOC just scratched the surface of patterns and frameworks for concurrent and networked software, which are much broader and deeper topics than we could possibly hope to cover in a single MOOC. The Powerpoint presentations therefore also contained extensive URL cross-references at the bottom of many slides. These URLs pointed to resources, papers, documentation, and source code that we or other experts have produced on topics related to material covered in the POSA MOOC.

Presenting the POSA MOOC material on camera was also *much* harder than giving lectures in class since there were no students to ask questions or give visual cues indicating if they comprehended the material. We therefore had to master the art of presenting slides smoothly and at an even pace, as well as maintaining enthusiasm while staring into the steely gaze of a video camera for hours at a time. A significant amount of time was also spent learning and applying screen capture and video editing tools during the post-production process to fix various glitches that inevitably crept into the videos and Powerpoint presentations.

### 3.1.2 Creating Student Assessment Mechanisms

In addition to the time required to develop the videos, we needed considerable time to formulate in-video quizzes (which was a tedious process we called “quizzification”), weekly quizzes (which counted towards the final grade, whereas the in-video quizzes did not), and peer-graded essays and programming assignments. In traditional classrooms, open-ended questions enable students to synthesize their knowledge in a free-form way, with instructors interpreting student responses and providing immediate feedback. Due to the infeasibility of providing individual feedback to thousands of students in a MOOC, however, we needed much more effort up front to construct questions for quizzes and rubrics for peer grading in a clear and helpful manner. Section 3.2 describes how we

leveraged crowd-sourcing to improve the POSA MOOC assessment mechanisms after the class began.

Help from experts in other areas helped to temper the increased time commitment stemming from the large amount of work involved in creating the POSA MOOC. For example, we leveraged the skills of experienced project management and video production experts at Vanderbilt, who filmed the lectures and edited/rendered the video content, thereby increasing the productivity and quality of the course production process. Having knowledgeable members of Vanderbilt’s *Center for Teaching* ([cft.vanderbilt.edu](http://cft.vanderbilt.edu)) and *Associate Provost for Undergraduate Education* to (1) answer our numerous questions about using Coursera platform, (2) help create and review supplementary course material, and (3) assist in the review process was also invaluable. In addition, the Coursera course operations specialists did a masterful job of supporting our needs both before and during the POSA MOOC.

### 3.2 An Even Larger Amount of Time is Needed to Manage a MOOC After Launch

After finishing the rendering and quizzification of the final POSA MOOC videos we had a great sense of relief since we thought the hard part of the project was over. Little did we know that the most demanding aspects of our work was just beginning. After the POSA MOOC went live at 2pm GMT on March 4th, the online discussion forums were immediately inundated with questions from students around the world, who wanted answers to questions ranging from what programming languages were allowed to clarifications about how the MOOC grading policies affected their chances of receiving Statements of Accomplishment. We each spent 40+ hours per week attempting to answer every question posed by the students.

In addition to enjoying our interaction with motivated students on topics associated with patterns and frameworks for concurrent and networked software, we spent so much time in the discussion forums for the following reasons:

- **Accelerate and amplify the learning process.** We quickly realized that the discussion forums were essential to the learning process. In particular, these forums helped transform the course from passively watching “lecture-oriented” videos to actively engaging in “learning-oriented” dialogue between us and the students. It was fascinating to watch students evolve and deepen their understanding of the course material based on the types of questions and resulting conversations that occurred in the discussion forums.

- **Dispel common misconceptions.** Our heavy involvement in the discussion forums was essential in dispelling common misconceptions resulting from “folk lore” that’s accrued over time in the software community. For example, many students initially thought that (1) patterns were only applicable to object-oriented programming languages, (2) the only purpose of patterns was to compensate for deficiencies in mainstream languages (particularly C++ and Java), and (3) object-oriented frameworks incurred too much time/space overhead for use in resource-constrained systems. These types of misconceptions rarely arise in face-to-face courses at Vanderbilt since few undergraduates have sufficient experience with alternative programming languages and design methods to raise these issues. In the MOOC environment, however, there were many spirited debates on these topics.

- **Build good will.** Maintaining a continuous course staff presence in the discussion forums also built up good will that paid off in various ways throughout the POSA MOOC. For example, many students contributed back to the MOOC by crowd-sourcing the programming assignment specifications beyond C++ to different programming languages (including Java, C#, Python, Ruby, and Scala) and different frameworks beyond ACE (including Netty, Twisted, Qt, and Boost). Students also crowd-sourced the entire contents of POSA MOOC wiki site ([share.coursera.org/wiki/index](http://share.coursera.org/wiki/index)).

php/posa:Main), which ultimately provided a detailed glossary of technical terms used in the videos, as well as a comprehensive list of all URL cross-references to other technical literature we embedded in the Powerpoint slides.

- **Reward constructive student participation.** Based on our involvement in other MOOCs, we noticed that the tone and content of discussion forum postings often tended to devolve into frustration and *ad hominem* attacks without the consistent presence of course staff. In general, the conversations that occurred in the POSA MOOC discussion forums were civil and technically focused. The most active participants were also quite knowledgeable and thoughtful in their postings, which we explicitly encouraged through our heavy engagement in the forums.

Although it was time consuming to engage with all students posting on the discussion forums, this investment yielded significant improvements in course material, better understanding for course staff and students, and a greater camaraderie in the POSA MOOC “virtual classroom.” Ideally, the time commitment required for future offerings of the POSA MOOC will be reduced by codifying our answers to frequently asked questions into a “FAQ” database and leveraging the assistance of “Community TAs” drawn from the pool of students who did well in the initial offering.

### 3.3 The POSA MOOC Student Diversity was Both Challenging and Rewarding

Students in our Vanderbilt courses come from diverse socioeconomic and national backgrounds, but are uniformly smart (their average SAT scores asymptotically approach perfection) and generally well-prepared (having successfully completed the prerequisites). In contrast, student background, expertise, and interests in the POSA MOOC were *much* more diverse, as discussed in Section 1, which yielded the following challenges:

- **Much greater level of skepticism from experienced students.** Undergraduates at Vanderbilt rarely have much programming experience when they arrive. In contrast, many POSA MOOC students—especially those posting frequently in discussion forums—had 20+ years of experience as software professionals. They manifested this experience by stating much stronger preferences for particular programming languages, runtime platforms, software tools, and development methods. They also asked *much* harder and more probing/skeptical questions about the pros and cons of different technical approaches relative to typical students in undergraduate courses at Vanderbilt.

Instructors benefit from being challenged to communicate clearly and justify their positions, just like students. We therefore found it was tremendously educational (and exhilarating) to engage in discussion forum debates on many topics in the POSA MOOC. Moreover, most students eventually came to appreciate the benefits of patterns and frameworks as a result of our discussions. We spent substantially more time, however, motivating and justifying the topics and techniques covered in the lectures than we’d expected based on our prior experience at Vanderbilt.

- **Increased workload to fill knowledge gaps.** At the other end of the spectrum were POSA MOOC students who lacked the necessary background since Coursera enforces no prerequisites, as noted in Section 2.2. We therefore filmed over eight hours (*i.e.*, nearly half the videos) of supplemental material to prepare less experienced students for the key topics of the POSA MOOC. Likewise, we spent a great deal of time in the discussion forums explaining basic concepts of object-oriented design and programming to inexperienced students. In contrast, our Vanderbilt students would have had these prerequisites, which greatly reduces the time required for course preparation and delivery.

In hindsight, a better solution than filming supplemental material would have been to create a group of related MOOCs that

students could take as a sequence. As with the curricula offered in conventional on-campus Computer Science and Computer Engineering programs, the goal of sequenced MOOCs is to introduce students to the material in the right order and at the right level. Section 5 describes the sequenced MOOCs we are creating together with the University of Maryland for the spring of 2014.

- **Generalizing from limited perspectives impedes learning.** Vanderbilt students are generally well-rounded, *e.g.*, they are trained—and predisposed, given the rigorous admissions process that emphasizes a range of scholastic and extracurricular activities—to think critically from multiple (holistic) perspectives and are equally facile at verbal, written, and quantitative skills. Moreover, the Computer Science and Computer Engineering programs at Vanderbilt expose undergraduates to a wide range of programming languages, development environments, operating systems, middleware, development methods, and application domains.

In contrast, judging from the comments and questions on the discussion forums, many of the POSA MOOC students tended to express themselves from much more narrow (and reductionist) perspectives. For example, if they developed one-off end-user apps they thought everyone just developed one-off end-user apps (and thus had no use for developing frameworks that encouraged systematic reuse). Likewise, if they worked in an organization that didn’t appreciate the value of software design they assumed everyone worked in an environment that doesn’t appreciate software design (and thus there was no point in learning about design patterns). Helping to broaden student perspectives was an essential role we played in the POSA MOOC discussion forums.

- **Overly narrow focus on programming.** A point related to the limited perspectives discussed above was how students reacted to different types of assignments. It’s not unusual for Computer Science and Computer Engineering courses at Vanderbilt to have short-essay questions, either as homework assignments or quizzes, since instructors (and most students) recognize the strategic value of communicating effectively via technical and persuasive writing. As shown in Figure 2, however, most POSA MOOC students did not fit the profile of university students who’ve recently taken courses in writing or other non-technical subjects. Many of these students were highly averse to the short-essay questions we assigned in the early part of the MOOC.

We originally intended these short-essay questions as a means to assess the students’ ability to codify/convey their understanding of the software *design*, which formed the bulk of the Section Two videos (outlined in Section 2.2.3). Design knowledge isn’t adequately assessed by having students write programs that are assessed for functional correctness. Although functional correctness is necessary, it’s not sufficient to demonstrate the application of good design practices and patterns.

Based on feedback we got on the discussion forums, however, we quickly realized that the bulk of the POSA MOOC students wanted to write programs, not essays. We therefore adapted our assignments after the first several weeks and updated the rubrics used to assess the essays to downplay the weighting of grammar and correctness based on concerns voiced by non-native English speakers from the wide range of countries shown in Figure 3. We also rapidly switched to programming assignments instead of short essays for the remainder of the MOOC.

Rewarding aspects of the POSA MOOC diversity included:

- **Highly stimulating discussions with expert software developers.** As noted above (and in Section 3.2) we engaged in many informative conversations (and intense debates) on common misconceptions about patterns and object-oriented frameworks in the POSA MOOC discussion forums. These discussions rarely occur in our classes at Vanderbilt, where most students lack the experience of expert practitioners, and thus the same (few) students typ-

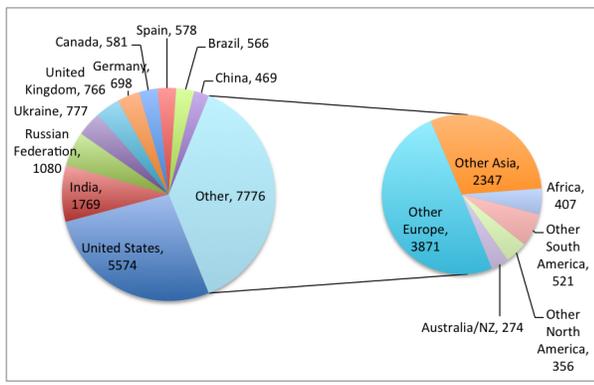


Figure 3. POSA MOOC Enrollment by Country

ically answer most questions. Getting other students to engage is often hard, which makes the class less interesting for both instructors and students. In contrast, we had extensive student engagement in the POSA MOOC. Although the relative anonymity of a MOOC certainly helped encourage greater student participation, it was also clear that expert practitioners were eager to share their experience.

- **Greatly improved course structure and content.** Another benefit of the diversity in the POSA MOOC student population was the vast amount of feedback given by the students on the discussion forums. Students in our on-campus courses rarely provide detailed feedback on the course contents. Again, this lack of feedback likely stems from the fact that 18-22 year-olds have limited experience with developing production software, so they aren't attuned to subtle problems with the material.

In contrast, many students in the POSA MOOC had decades of experience developing software and were eager to share their knowledge in the discussion forums. Moreover, they quickly found and reported mistakes or ambiguities in the POSA video slides/lectures, quizzes, and peer-graded assignments. This crowd-sourcing feedback enabled us to support multiple programming languages in assignments, as well as helped us improve course material, e.g., fixing typos in slides, repairing broken links, and improving peer-graded assignment rubrics.

Such a rapid, iterative course improvement process is rare in conventional Vanderbilt classes, which are much smaller (typically 10-20 students), so there's less critical mass for crowd-sourcing. There are also fewer opportunities for students to provide structured feedback on course material. For example, by the time this feedback is solicited in a post-course survey, students may have forgotten their earlier concerns.

In addition to improving POSA MOOC content, student feedback also improved MOOC structure and logistics. For example, we released videos over weekends so students could watch them outside of work. We also moved all quiz deadlines to the end of the MOOC, so students could finish these auto-graded quizzes at their own pace. In general, we found that being responsive to diverse student needs helped us build a more effective learning environment for the POSA MOOC.

### 3.4 Assessing Student Performance in a “Design-oriented” MOOC is Harder than in “Fact-oriented” MOOCs

In our experience, the POSA MOOC students wanted meaningful ways to assess their progress, i.e., they didn't just want to watch videos, they want to actually apply what they'd learned into practice. For example, after we started releasing peer-graded programming assignments there was an explosion of participation in the discussion forums, with students posting their questions/solutions

and generating many interesting analyses and comments on each others solutions. Our challenge was to assess their solutions in a realistic and scalable manner.

Assessing student performance in the software design and programming courses we teach at Vanderbilt involves a significant amount of manual scrutiny. For example, we personally review and comment upon every line of software written by our students. While that level of personalized scrutiny may work for a small class in a private (and expensive) university, we couldn't afford to assess thousands of solutions from MOOC students (who generally pay nothing to the course staff).

Ideally, scaling up POSA MOOC assessments would apply auto-grading tools. Despite being an active field of research, however, the auto-grading tools available to assess students in a design-oriented MOOCs (such as POSA) aren't as mature as in a “fact-based” MOOC (such as Algebra or Pre-Calculus). The problem stems from the lack of useful tools for auto-grading software *designs* in terms of (relatively subjective) quality attributes, such as reusability, understandability, and evolvability.

We evaluated these quality attributes in the POSA MOOC via Coursera's two-phase peer-assessment feature. In phase one, students uploaded their submissions to the Coursera platform by the deadline. In phase two, students automatically received an anonymized set of submissions from other students. The Coursera web interface guided students through a grading rubric and they scored each submission and entered free-form comments. They had to finish grading their assigned submissions (we had them grade four submissions each) before the grading deadline or they received a 20% penalty for their own submission). After this deadline passed, students could see their final grade, which was an average of the grades they were given by their four peer graders.

Although Coursera's peer-assessment feature is scalable, the wide range of student abilities, motivations, and time constraints made peer assessment problematic. In the POSA MOOC, we started with simple rubrics that gave students significant freedom in assigning grades. We met with resistance almost immediately since some students were lax graders and others were strict, so we switched to providing more detailed rubrics.

Although we had better success with detailed rubrics, the lack of expert judgement was evident, regardless of the specificity of the prompts and rubrics. In our future MOOC offerings, therefore, we plan to use more systematic assessments by calibrating the peer review process [9] with videos that walk through our solutions and compare them against the rubrics.

The limitations with peer-assessment and auto-grading in MOOCs underscore the invaluable role that expert judgment plays in fostering critical thinking by—and experienced evaluation of—students in on-campus courses at immersive educational institutions like Vanderbilt. In other words, you get what you pay for.

### 3.5 The Coursera Platform is a Work-in-Progress

Bugs, kludges, workarounds, and missing features are realities in all the existing MOOC platforms. While none of the problems we encountered with the Coursera platform used for the POSA MOOC were show-stoppers, we did encounter the following limitations, some from our perspective and some from the students' viewpoint.

- **Lack of an integrated view of their progress.** We were initially overwhelmed by students' obsessions with their scores on various quizzes and peer-graded assignments. These obsessions were manifested by the volume of questions about how grades were calculated and complaints from students who didn't know their current grades. Moreover, there was simply no single place in the Coursera platform for students to get a sense of completion status or due dates for assignments.

We repeatedly answered the same questions on the discussion forum to help students attain a bird's-eye view of their status since the Coursera platform did not support this capability. Obviously, this could be very different for different courses, so a one-size-fits-all solution would not work, but we managed to adapt one case at a time. Eventually, we created a semi-automated grading calculator (discussed in Section 3.6.3), but what's really needed is a standard solution provided by the Coursera platform itself.

- **Information overload on the discussion forums.** In addition to repetitive questions from students about their grades on the discussion forums, many students didn't look for existing answers before posting their questions. After answering the same questions several times, we decided to keep a running list of links to common questions that we'd answered before, so that we could efficiently guide them to the information they needed. While this practice wasn't too time-consuming for us, it indicates something that Coursera should emphasize in future platform releases to help students find the resources they need quickly, *e.g.*, some type of automated FAQ capability.

- **Lack of scalability as a MOOC progresses.** Many POSA MOOC students requested the programming assignment specifications be provided in languages beyond Java and C++. We agreed to do this as long as students themselves crowd-sourced the appropriate changes to the specifications. While this enhanced the breadth of the POSA MOOC by allowing students to program in the language of their choice, it introduced quite some confusion due to limitations with the Coursera platform.

For example, there was no way to put different peer-graded assignments into subfolders under "Peer Assessments." Since each assessment was a unique combination of programming language, (optional) framework, and assignment number, the number of assignments on the page skyrocketed as the MOOC progressed. Moreover, Coursera provided us with no automated means to help students who submitted to an incorrect link (*e.g.*, submitted a Java solution to the Python version of the assignment) after the submission period closed. There was also no easy way for students who submitted assignments in multiple programming languages to see their grades since they had to open each submission and calculate the highest score from all their submissions (which became their final grade for that assignment).

- **Limited hardware/software support for the Coursera platform.** Since MOOCs are intended for students around the world, MOOC delivery platforms should ideally be compatible with a wide range of browsers and devices. When we taught the POSA MOOC in the spring of 2013, however, the Coursera platform primarily supported Chrome, Firefox, and Internet Explorer. Other common hardware/software combinations, such as Android and iPhone/iPad, weren't well supported.

When our students encountered issues with different phones, tablets, and browsers they complained to us via the discussion forums. Occasionally, we could suggest a quick fix, but more often our only recourse was to point them to the official Coursera technical support portal. To their credit, the Coursera technical support staff quickly fixed many problems we reported.

For example, it's now possible for the course staff to edit typos and mistakes in video subtitles. Originally, course staff could not perform these edits, which meant that non-native English speaking students were often confused when they tried to read the nonsensical subtitles (mis)transcriptions. Likewise, Coursera added support for "permalinks" (see [en.wikipedia.org/wiki/Permalink](http://en.wikipedia.org/wiki/Permalink)) to discussion forum postings, which allowed us to post links to common questions that we'd answered before.

In general, we were pleased with the scalability of the Coursera platform and the responsiveness of the Coursera operations staff.

### 3.6 Innovations Helped Make the POSA MOOC More Like a "Real" Course

Despite the limitations with the Coursera platform discussed in Section 3.5, our goal was to make the POSA MOOC seem like a "real" course. The following is a summary of the innovations we devised for the POSA MOOC that went beyond conventional usage scenarios for the Coursera platform. Some of these innovations have become standard practice in other MOOCs.

#### 3.6.1 Virtual Office Hours

Learning involves much more than watching videos—it requires meaningful dialogue between students and teachers to clarify doubts and deepen collective understanding of the material. A common criticism [11] of conventional MOOCs is that they dehumanize the learning experience by neglecting or degrading interactions between students and teachers. To address this limitation, we used two social media tools—discussion forums (discussed in Section 3.2 above and webcasting (discussed in the following paragraph)—to engage in continuous interactive dialogue with our POSA MOOC students.

We used Google Hangout and a YouTube channel (see [www.youtube.com/user/vuposa](http://www.youtube.com/user/vuposa)) to hold weekly "virtual office hours," where students asked questions about assignments and videos via instant messaging and we broadcast answers to them live. Google Hangout automatically recorded the office hours, which we uploaded to the POSA Coursera website so students could view them offline. Roughly 70–100 students participated live for virtual office hours each week, but well over ten times that number viewed the archived videos of the virtual office hours (nearly as many as those who watched the higher-quality videos lectures).

These viewing statistics are particularly noteworthy since we recorded the virtual office hours on a laptop webcam connected to a YouTube channel. This technology was *much* less sophisticated than the green screen technologies we used in the video lectures, which indicates MOOC students are attracted to more than flashy visuals. As social media technology matures it should become feasible (albeit potentially time consuming) for MOOC professors to communicate with students in ways similar in quality and quantity to those found in large lecture courses at many universities.

#### 3.6.2 Crowd-sourced Programming Assignments

Our use of crowd-sourced programming assignment specifications was particularly effective at broadening the scope of the course and engaging more students. Although the videos largely focus on C++ and Java (*i.e.*, the programming languages with which we are most fluent), the peer-graded programming assignment descriptions were crowd-sourced to include C#, Ruby, Python, and Scala. Incidentally, the total number of programming assignment submissions for each programming language was Java (2,205), C++ (869), C# (450), Python (382), Ruby (100), and Scala (75).

Supporting this diversity of programming languages is impractical in a course at Vanderbilt, due to the effort needed to understand and assess assignments written in many different languages. The Coursera peer assessment system, however, enabled students to complete assignments in programming languages that were foreign to the course staff. Students could therefore tailor the course to meet their needs/interests, while still enabling us to teach common architecture and design structures and behaviors via patterns.

#### 3.6.3 Grading Calculator

The Statements of Accomplishment discussed in Section 2.3 conferred no course credit at Vanderbilt. Despite this lack of college credit, many students in the POSA MOOC were quite concerned with the grading policies. For example, many non-native English speakers were frustrated by the short-essay assignments because

they feared they would be penalized for poor grammar or language skills, as discussed in Section 3.3. We didn't anticipate this much concern with grading—in fact, students in the MOOC seemed as concerned with grades as undergraduates in our credit-bearing courses at Vanderbilt.

Not surprisingly, therefore, many students requested an easy-to-use, bird's-eye view of their progress in the course relative to the criteria needed for attaining a "Statement of Accomplishment". In response to these requests, We built a simple calculator in JavaScript and attached it to a wiki page. Our solution, however, still required students to navigate to several different pages to figure out their individual assignments scores to feed into the calculator. This capability therefore should really be part of the Coursera platform so it's available to all students in all MOOCs.

### 3.7 Interpret MOOC Enrollment Statistics Carefully

Although 31,000+ students enrolled in the POSA MOOC, only ~20,000 ultimately ever logged in and participated in some way, such as by watching videos or reading/posting in the discussion forums (Figure 3 provides a breakdown of the countries these ~20,000 students hailed from). Moreover, only ~1,600 of these participants actually received some form of Statement of Accomplishment, as shown in Figures 4 and 5)

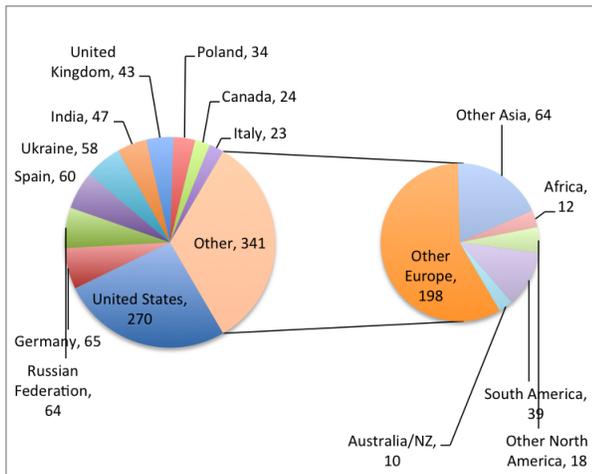


Figure 4. Normal Track Completions by Country

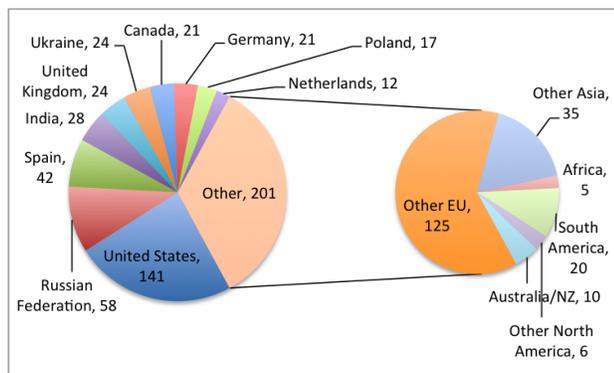


Figure 5. Distinction Track Completions by Country

A completion rate of 5-10% (which is consistent with other published studies [5]) may seem less impressive than the original

31,000+ enrollment figure suggests. Even having ~1,600 students complete the POSA MOOC is notable, however, since it would take us 20+ years to teach that many undergraduates at Vanderbilt. Moreover, it's clear from numerous conversations in the discussion forum and virtual office hours that students learned a great deal from the POSA MOOC, even if they didn't have time to obtain a Statement of Accomplishment.

Some intriguing findings are suggested by the statistics collected from the Coursera platform. For example, the 4.2% completion rate for students from India (1,769 started and 75 finished) versus the 11.3% completion rate for students from the Russian Federation (1,080 started and 122 finished). We have not yet identified the cause for such differences, but our future work will examine the correlations between completion rate and various diversity related factors (such as broadband penetration, English proficiency, age, and types of software jobs available in various countries) discussed in Section 3.3.

### 3.8 MOOCs Can Enhance Student-centered Learning Opportunities

As educational researchers and teachers have preached for years, every person learns differently, and MOOCs open up new opportunities for learning to happen in a more student-centric way, especially at scale. The remainder of this section describes several ways we predict that MOOC platforms will impact student-centered learning based on our experience with the POSA MOOC.

#### 3.8.1 Increasing Asynchrony in Courses

A traditional university course might begin at 8am on Mondays, Wednesdays, and Fridays, with some students in the room wide-eyed and ready to learn, while others are barely awake. Likewise, many students may have avoided that particular course because it did not fit in their schedule. MOOCs can thus provide students—even those who live on a college campus—increased opportunities for taking courses they otherwise would not be able to take due to scheduling conflicts.

During the POSA MOOC, students could watch videos, take quizzes, and complete programming assignments at their own pace. Although students needed to finish certain parts at certain times, they were not restricted to completing their work by a certain day or time. This freedom granted by MOOC platforms allows students to tailor their learning to match their schedules.

#### 3.8.2 Location-agnostic Learning

We found that MOOCs enhanced the virtual classroom model used by other online learning systems in unique ways. For example, one of the POSA MOOC discussion forums was dedicated to students forming study groups, both online and offline. Some students simply exchanged e-mails throughout the course, or explored topics on the discussion forum. Other students got together in face-to-face meetup groups (both at Vanderbilt and elsewhere) to discuss various aspects of the MOOC. While this real-life interaction is not a guarantee, it's a benefit that some MOOC students realize compared with conventional online courses.

## 4. The Impact of MOOCs on Traditional On-campus Education

This section describes the impacts of MOOCs on traditional on-campus education based on our POSA MOOC experience.

### 4.1 Benefits of MOOCs

Some benefits of MOOCs were relevant to instructors and others were relevant to Vanderbilt, as discussed below.

#### 4.1.1 Benefits to Faculty

Several benefits that we observed as instructors for the POSA MOOC include the following:

- **Significantly better on-campus courses.** The lecture material for our on-campus courses is now much better than before due to (1) the months we spent preparing for the POSA MOOC (described in Section 3.1) and (2) the many contributions from students, who provided extensive feedback on the correctness, completeness, and clarity of the material. Eager students would often fix glitches in material the moment it was released, which helped all students who accessed the updated content later. These crowd-sourced improvements are a return on the investment spent preparing for a MOOC. As MOOCs grow in popularity, it is likely that several iterations of the same course will provide decreasing start-up costs with a constant rate of improvement, thus amortizing the initial time commitment, thereby yielding better on-campus courses *and* MOOCs.

Improvements in the POSA MOOC material not only benefited the students in the *Intermediate Software Design* course taught on-campus in the spring of 2013, but also helped ensure the consistency of future offerings across semesters and instructors. For example, the availability of the high-quality videos on the “Gang of Four” patterns produced for the POSA MOOC enable Vanderbilt students to personalize their learning, *e.g.*, they watch the videos before and/or after class at their own pace, read transcripts of the videos provided by Coursera, and learn from the in-depth conversations in the discussion forums. Moreover, the *Intermediate Software Design* course we’re offering in the fall of 2013 is applying “flipped” and “blended” classroom models [4], using POSA MOOC videos to shift some lecture content outside of class time. As a result, more class time is available to interact with and mentor our Vanderbilt students.

- **Fostering global life-long learning communities that connect students who possess a range of experience**, which helps compensate for the lack of critical mass in a local learning community. For example, there isn’t much demand in the Nashville IT community for pattern-oriented concurrent and networked software beyond our research group at the *Institute for Software Integrated Systems* ([www.isis.vanderbilt.edu](http://www.isis.vanderbilt.edu)). Teaching a conventional face-to-face continuing education course on these topics in Nashville would therefore be of limited value.

Beyond Nashville, however, there’s significant global interest in understanding patterns and frameworks for concurrent and networked software, as indicated by the POSA MOOC enrollment statistics discussed in Section 3.7. The MOOC format is ideally suited for disseminating this type of information and building a global life-long learning community around these topics. Moreover, Vanderbilt students benefit from being part of discussions with experienced software professionals from around the world, which provides them with a glimpse of both the global competitiveness and collaborative possibilities that they will face after graduation.

Related digital learning opportunities stemming from the POSA MOOC also help foster a global life-long learning community on patterns and frameworks for concurrent and networked software. For example, Professor Schmidt is creating a spinoff course on *Design Patterns in Java* for Pearson’s “LiveLessons” training series as a result of the POSA MOOC.

#### 4.1.2 Benefits to Vanderbilt

We observed several benefits to Vanderbilt that stemmed from our pilot project with the POSA MOOC, including the following:

- **Expanding the brand value of a Vanderbilt education.** The role that top-tier universities like Vanderbilt play in education goes well beyond lecturing to students in person or via videos. The clas-

sic method of determining the quality of education at a university is manifested in listings of degrees and honors conferred, grants awarded, and publications accepted by instructors and researchers. In a short span of time, MOOCs have doubled the number of students who have been taught by Vanderbilt professors, as discussed in Section 1. This wide-reaching visibility is helping Vanderbilt expand its brand value by providing a more transparent window into its classrooms, as well as demonstrating its interdisciplinary strengths in teaching, research, entrepreneurship, and innovation.

- **Better opportunities for engagement with alumni and prospective students.** The material produced for our MOOCs is being applied to better connect and (re)engage our alums with the intellectual life of Vanderbilt. Likewise, quality of a Vanderbilt education is now visible to thousands of students around the world, which encourages them to apply to Vanderbilt and partake in the immersive on-campus learning culture.

#### 4.2 Drawbacks of MOOCs

The following were some drawbacks we observed based on our POSA MOOC experience.

##### 4.2.1 Potential for “Deskilling” Education and Educators

A common concern with MOOCs [11] is that they will replace quality face-to-face education with impersonal delivery of information via digital media. While many of these concerns are being ameliorated via advances in MOOC technologies (such as the POSA MOOC innovations discussed in Section 3.6), it’s likely that administrators at some institutions will reduce costs by “deskilling” their faculty positions, *e.g.*, combining content from MOOCs from top universities with for-credit courses at their institutions that are proctored by untenured lecturers.

Based on the trends reported thus far [12], it appears that the institutions most impacted by MOOCs will be state universities and community colleges that (1) lack large endowments and research programs and (2) are committed to educating large numbers of students, despite cuts in funding from state legislatures. The positions affected at these institutions will most likely be tenured (or tenure-track) teaching faculty, who typically don’t receive significant external funding.

Universities with large endowments, robust research funding, and sufficient income from tuition and other sources (such as alumni donations) will continue (for now) to provide traditional immersive education. Even faculty at these institutions, however, may need to (and ideally *want* to) adapt their pedagogy to leverage digital learning methods and tools since it’s essential to justify the value of an immersive college education in today’s rapidly changing and globally competitive environment. In our experience, the effective use of digital learning techniques help amplify other value-added mentoring opportunities, such as undergraduate research, entrepreneurship, and collaborations with industry, in addition to enhancing traditional college learning experiences [1].

##### 4.2.2 MOOCs Require Substantial Institutional Investment and the Payoff Isn’t Clear (Yet)

Producing and delivering a high-quality MOOC requires a substantial commitment from course staff, as discussed in Section 3.1. Likewise, input from a plethora of campus administrators and staff is needed to review MOOC content, as well as address financial and legal concerns necessary to produce high-quality MOOCs that don’t infringe on intellectual property rights. Finding qualified people to conduct all these MOOC-related activities is hard, especially if there’s no clear economic benefit to the faculty and university.

Quantifying the return on investment (ROI) from MOOCs is tricky since their impact on the mission of most universities is not yet well understood. Top universities are rationalizing their in-

vestment in MOOCs primarily as outreach, *e.g.*, to help achieve the benefits mentioned in 4.1.2. It's not clear, however, the extent to which MOOCs actually improve on-campus course quality, strengthen relations with alumni, and help recruit the best and brightest students. Over time, high levels of institutional time commitment—along with money spent and resources used—will be hard to sustain without a clear ROI, even for a university like Vanderbilt with extensive human and financial resources.

### 4.2.3 Detecting and Dealing with Plagiarism is Tedious

Plagiarism is a chronic issue with online courses, including non-MOOC online courses. For example, it's easy to copy-and-paste answers found online into the Coursera submission form, which happened surprisingly often in our POSA MOOC, despite the fact that we weren't offering "official" Vanderbilt credit for successfully completing the MOOC. Since we didn't offer our MOOC for "real" college credit, our (admittedly limited) solution to plagiarism we encountered was simply to instruct students to cite their work and/or submit a comment with links if they posted their solutions in a blog post or online. These references enabled other students grading their work to do a cursory check to see if the work was plagiarized. With systems such as TurnItIn.com and Grammarly, we expect that Coursera and other MOOC platforms will soon integrate more effective and meaningful plagiarism detection.

## 5. Concluding Remarks

During a five month period in the winter and spring of 2013 we devoted hundreds of hours to producing and delivering a Coursera-based MOOC on patterns and frameworks for concurrent and networked software. Our experience was generally positive—and at times exhilarating—albeit exhausting. For a variety of reasons—not the least of which is that it would have taken us ~500 years to teach 31,000+ students at Vanderbilt—the POSA MOOC differed significantly from our previous experience teaching software design and programming courses to Vanderbilt undergraduates.

MOOCs are particularly relevant to software professionals in academia and industry because future researchers and practitioners will likely receive much of their education through MOOCs and associated digital learning methods and tools [8]. The ability of MOOCs to connect experienced software professionals with motivated novices is a potent pedagogical combination. MOOCs also have the potential (not yet fully realized) to help students personalize their learning experiences at a reasonable cost.

Based on our experiences with the POSA MOOC, we are expanding our Coursera offerings in the spring of 2014. The next POSA MOOC will be an intentionally-coordinated, trans-institution *sequence* of MOOCs that focus on patterns and frameworks for mobile device programming. These sequenced MOOCs will showcase how intentionally-coordinated MOOCs can create life-long learning communities that (1) cross-cut traditional institutional/disciplinary boundaries and (2) would not be feasible without the MOOC paradigm and MOOC platforms like Coursera.

Our sequenced MOOCs will be organized as follows:

- Professor Adam Porter at the University of Maryland, College Park will focus his MOOC on the GUI/client portions of the Android platform starting in January 2014 and
- Professors Doug Schmidt and Jules White at Vanderbilt will focus their MOOC on the patterns/frameworks for the server portions of Android and services in computing clouds in March 2014 at the conclusion of Porter's MOOC.

Coordinated programming assignments will span both MOOCs to integrate material covered in the videos. Students in Porter's MOOC will build GUI/client portions of an app using server modules provided as "blackboxes" by Schmidt and White. Likewise, students in Schmidt and White's MOOC will build server portions

of the app using client modules provided as "blackboxes" by Porter. These coordinated assignments will yield a complete solution that demonstrates the pattern-oriented integration of Android mobile devices with cloud computing platforms.

Despite our enthusiasm about the POSA MOOC, we recognize that MOOCs pose many social, economic, and technical challenges. For example, it's not clear yet how to prevent administrators at some cash-strapped institutions from diminishing the quality of higher education via wanton replacement of experienced faculty with MOOCs and inexpensive lecturers. Other discussions of our experiences with the POSA MOOC—and our observations about the MOOC paradigm—can be found at [www.dre.vanderbilt.edu/~schmidt/Coursera.html](http://www.dre.vanderbilt.edu/~schmidt/Coursera.html).

## Acknowledgments

Thanks to Dennis Mancl for his feedback on our initial drafts and to Vanderbilt for supporting our production and delivery of the POSA MOOC. In particular, Richard McCarty, Cynthia Cyrus, Derek Bruff, Michael Martin, Jeff Shoop, John Brassil, Doug Fisher, Melanie Moran, Jesse Badash, Katie McEwen, Sarah Collier, David Owens, Jay Clayton, and Jamie Pope provided invaluable assistance and helped position Vanderbilt for a leadership role in using MOOCs and other digital learning technologies effectively. Finally, we'd like to thank our Coursera liaisons—Ryan George and Pang Wei Koh—who patiently guided us through the mysteries of the Coursera platform. We look forward to seeing how they and the other MOOC innovators will enable affordable student-centric learning via digital learning methods and technologies.

## References

- [1] Bruff, D., D. Fisher, K. McEwen, B. Smith, "Wrapping a MOOC: Student Perceptions of an Experiment In Blended Learning," Accepted to the *Journal of Online Learning and Teaching*.
- [2] Buschmann, F., et al. *Pattern-Oriented Software Architecture, Volume 5: Patterns and Pattern Languages*, Wiley, 2007.
- [3] Gamma, E., et al. *Design Patterns - Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
- [4] Fisher, D., "Warming Up to MOOC's," *Chronicle of Higher Education*, November 6, 2012.
- [5] Hill, P., "The Most Thorough Summary (to date) of MOOC Completion Rates," *e-Literate*, February 26th, 2013.
- [6] Hu, J., and D. C. Schmidt. "JAWS: A Framework for High Performance Web Servers," *Domain-Specific Application Frameworks: Frameworks Experience by Industry* (1999).
- [7] Johnson, R. "Documenting Frameworks Using Patterns," Proceedings of the ACM OOPSLA Conference, Vancouver, British Columbia, October 1992.
- [8] Meister, J., "How MOOCs Will Revolutionize Corporate Learning And Development," *Forbes*, August 13, 2013.
- [9] Morrison, D., "Why and When Peer Grading is Effective for Open and Online Learning," *Online Learning Insights*, March 9, 2013.
- [10] Murphy, M. *Busy Coder's Guide to Android Development*, CommonWare, 2013.
- [11] Rees, J., "The MOOC Racket," *Slate*, July 25th, 2013.
- [12] Rosenberger, J., "John L. Hennessy on 'The Coming Tsunami in Educational Technology'," *Communications of the ACM Blog*, July 23, 2012.
- [13] Schmidt, D. C., et al. *Pattern-Oriented Software Architecture, Volume 2: Concurrent and Distributed Objects*, Wiley, 2000.
- [14] Schmidt, D. C., and S. Huston, *C++ Network Programming: Systematic Reuse with ACE and Frameworks*, Addison Wesley, 2002.
- [15] Schmidt, D. C., D. Levine, and S. Mungee, "The Design of the TAO Real-time Object Request Broker," *Computer Communications*, 21.4 (1998): 294-324.