Volume 14, No. 5
September 2016
ISSN: 1545-679X

INFORMATION SYSTEMS EDUCATION JOURNAL

In this issue:

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published reviewed published by **ISCAP,** Information Systems and Computing Academic Professionals. The first year of publication was 2003.

ISEDJ is published online (http://isedj.org). Our sister publication, the Proceedings of EDSIGCon (http://www.edsigcon.org) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the 1st Edition of Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to members of AITP-EDSIG who perform the editorial and review processes for ISEDJ.

Full Flip, Half Flip and No Flip: Evaluation of Flipping an Introductory Programming Course

Meg Fryling
mfryling@siena.edu

Robert Yoder
ryoder@siena.edu

Eric Breimer
ebreimer@siena.edu

Computer Science Department
Siena College
Loudonville, New York 12211, USA

Abstract

While some research has suggested that video lectures are just as effective as in-person lectures to convey basic information to students, not everyone agrees that the flipped classroom model is an effective way of educating students. This research explores traditional, semi-flipped and fully-flipped classroom models by comparing three sections of an Introduction to Programming (Java) course that were taught at the same institution in Spring 2015 by three different instructors using three different paradigms. The data and observations collected suggests that incorporating in-class activities improves student satisfaction but a semi-flipped classroom, including in-class activities, some outside-class lecture videos, and some in-class lectures, may generally provide the best overall experience for the students. However, while students may be more satisfied and get more programming practice in a flipped paradigm, overall student performance did not appear to be greatly impacted.

Keywords: Flipped Classroom, Inverted Classroom, Java, Introduction to Programming, Information Systems Education

1. BACKGROUND

The flipped classroom is an alternate teaching methodology being used in some high schools and college classrooms as a means of increasing student engagement and academic performance (Clark, 2015; Danker, 2015; Gunyou, 2015; Strayer, 2012; Vaughan, 2014). The flipped method operates by altering the traditional model of conducting lectures in the classroom, to one where lecture materials are distributed to students for their study outside of the regularly scheduled class meeting times, leaving in-class time for activities such as homework assignments. Lage, et al. define the flipped/inverted classroom as one in which the "…events that have traditionally taken place inside the classroom now take place outside the classroom and vice versa" (2000, p. 32). Scheduled class times then involve students working individually or in groups, with computer and video technology on activities and assignments pertaining to the course subject matter (Bishop & Verleger, 2013; Clark, 2015; Gaughan, 2014; Herreid & Schiller, 2013; Vaughan, 2014). Findlay-Thompson & Mombourquette (2014, p. 64) argue that "a flipped classroom is most commonly described as a reverse teaching model where the teacher uses various forms of technology such as videos to record the normal classroom lectures and students are required to view these recorded lectures outside regularly scheduled classroom time. This allows for the homework portion, or other interactive activities, to be completed within the classroom setting." Similarly Bishop & Verleger (2013, p. 4), suggest that the flipped classroom can be considered "an educational technique that consists of two parts: interactive group learning activities inside the classroom, and direct computer-based individual instruction outside the classroom."

Prior research has suggested that video lectures are just as effective as in-person lectures to convey basic information to students (Zhang, Zhou, Briggs, & Nunamaker, 2006). Therefore, some believe that using class time to reiterate textbook material is not the most effective use of class time and may only encourage students to skip the reading entirely. There is an increasing number of faculty at various secondary schools and colleges that realize that the traditional lecture style of teaching has been ineffective in meeting the educational needs of students (Clark, 2015; Gunyou, 2015; Vaughan, 2014).

Researchers have reported a variety of positive aspects to the flipped model (Saulnier, 2015), especially in courses that can utilize lecture time for computer-based activities (Frydenberg, 2013). However, some research indicates that there are some potential negative aspects (Strayer, 2012) where students reported feeling lost and "…were more likely to disengage with the material sooner than students in the traditional class-room" (2012, p. 189).

2. INTRODUCTION

This research explores the impact of three different teaching models (traditional, semi-flipped and fully-flipped classroom) in an Introduction to Java programming course. This course is required in our Computer Science major as well as our Computer Science and Information Systems minors. Each course section meets for 3 hours per week in a lecture room (32 students max) and smaller groups (16 students max) meet for 2 hours per week in computer labs to complete programming activities. This research focuses on changing the lecture format. The labs were identical for all sections.

Three sections of the Introduction to Programming course were taught at the same institution in Spring 2015 by three different instructors; hereafter referred to as instructor T (Traditional course), instructor S (Semi-flipped course), and instructor F (Fully-flipped course). The formats were selected by the instructors strictly based on preference. Instructor F had never taught this particular course in a traditional format but had previously taught the course in a fully flipped format and had already developed lecture videos. Therefore, it was logical that instructor F teach the course in a flipped format again. Instructors T and S had both taught the course previously in a traditional format but instructor S was interested in trying the flipped format in combination with some traditional lecturing. This led us to a unique situation in which the three course sections in the same semester were offered in three different teaching styles.

Since instructor F had taught the course most recently, it was agreed that all sections would use instructor F's course schedule so that exams, lab practical exams, and homework assignments could be similar across the sections. Unlike previous offerings of the course, students were allowed to select any lab/lecture combination. This meant they did not necessarily have the same lecture instructor as they did for lab. Tests were given during lab time to give students enough (and equal) time to complete the test (2 hours), since two sections had lectures twice a week (85 minutes each class session) and one (instructor T) met three times each week (60 minutes each class session). This made it critically important that all sections of the course stay on the same schedule, covering the same material at the same time.

The exams and lab practical exams were developed collectively among the instructors and were all very similar but not identical since the students were not all taking them at the same time. The five homework assignments and the final exam were also developed collectively and were identical for all sections of the course. The first homework assignment was similar to the in-class activities, including several small exercises to practice basic skills. The last four homework assignments were larger projects that allowed students to use the knowledge they gained to develop working games.

Students were unaware of the format differences prior to the start of the class so there was no self-selection for a particular teaching style during registration. Additionally, no students changed sections after the classes began.

3. METHODOLOGY & FINDINGS

Traditional Classroom (T)
Instructor T planned on delivering traditional PowerPoint and whiteboard-based lectures, with some class discussion and quizzes each week based on the text readings. Since no in-class activities were planned for this section of the course, Instructor T was not assigned any teaching assistants (TA's). Instructor T's section included 13 students with 8 CS majors, 1 CS minor and 1 Physics/Computational Science major.

About three weeks into the course, a major problem emerged: since most of the class time was spent on lecture, material was covered much faster than the fully- and semi-flipped lecture sections. Instructor T was already a chapter and a half ahead of the other sections and it was impossible to proceed at this pace since all instructors agreed ahead of time to give our students the same exams and the same labs.

Initially, the students seemed to be comfortable with the pace of the traditional course – two of the 13 students had a Java class in high school. However, the quiz results were disappointing – probably because students were not doing the reading assignments.

Before this semester, instructor T felt that students were getting enough practice programming through the labs and the five programming assignments given throughout the semester. Instructor T was compelled to shorten the lecture component and add a programming activity to each class session – this is what enabled the course to synchronize with the other course sections. While every student owned a laptop, sometimes they forgot to bring them or their laptop was being repaired. Fortunately, the department has five "loaner" laptops readily available.

The first in-class programming activity was end-of-chapter exercises as sometimes employed by the other instructors, but instructor T designed the rest to match the important points presented during the shortened lecture. Some activities built upon previous activities. All of the students immediately liked the programming activity, (student survey indicated 9.75 out of 10 – *strongly agree* that "in-class programming exercises improved my comprehension of the course material") even staying after class to complete them, and instructor T could give immediate help to struggling students. Students indicated that these should have counted for more of their grade, and instructor T agrees.

Another successful class activity was to play "The Good, the Bad, and the Ugly." The instructor selected excellent programs from labs or programming projects (the good), programs that did not work or had major problems (the bad), or mostly worked but were inefficient or hard to follow (the ugly). Student's names were removed from the code. The programs were projected on the screen, and students were invited to choose what category each example was from. The class would discuss why it was in that category, and explore ways to improve the code. Students actually requested this activity several times.

In previous semesters, there were a few students that failed to hand in programming assignments, usually resulting in lack of engagement and poor grades. To remedy this problem, instructor T would allow students to start their projects in class in place of the regular programming activity. This enabled instructor T to answer questions about the project that students encountered as they began coding. This resulted in less panicked students coming to office hours the day before the project was due, resulting in less stress for all involved.

A downside to having to readjust lecture time for programming activity was that the lectures were not as organized as they could have been; as recorded lectures for flipped classes tend to be carefully structured. However, it was liberating to be able to (mostly) abandon the slides and concentrate on key concepts.

In the future, instructor T plans to add in-class programming activities to other programming courses, along with fill-in worksheets to guide note taking during lectures.

While the traditional section did include the activity-based learning associated with a flipped lecture, this section did not include the video lectures of the fully- and semi-flipped sections, which enabled instructors S and F to devote even more lecture time to activities. Table 1 compares student evaluation scores for the last time instructor T taught the course (i.e. Fall 2005) versus the recent (i.e. Spring 2015) course. Instructor T did not teach this course between those two semesters (a 10 year gap) and thus, did not have the opportunity to refine or improve the course.

Measure	S 2015	S 2005
Classes helpful	7.8	6.8
Classes interesting	7.5	6.7
Course rating	8.1	6.8

Table 1. Instructor T Student Evaluation Scores

Semi-Flipped Classroom (S)

In 2010, instructor S was not satisfied with his student evaluations from this particular course and wished to improve student engagement. However, instructor S could not devote the time necessary to entirely flip the course and sought to create one video and in-class activity each week.

Instructor S's section included 33 students with 24 CS majors, 3 CS minors and 6 IS minors. This section's lecture was scheduled on Tuesdays and Thursdays for 85-minute sessions.

Instructor S narrated and produced videos using Camtasia that included the presentation of lecture slides and code demonstrations related to the assigned textbook reading for the week. While the videos were designed to represent what might happen in a single 85-minute lecture session, editing out pauses and the lack of student interaction allowed the information to be presented more efficiently in videos that ranged from 24 minutes to 50 minutes (45 minutes on average). Typically, the videos would be posted to the course website the Friday before a Tuesday activity session.

The instructor was able to create 9 videos and 10 in-class activities. During 10 of the 13 Tuesday lecture sessions, students were required to bring a laptop computer to take an online quiz using the Blackboard LMS and then complete a programming activity based on the reading assignment and video posted on Friday (see Appendix A for sample quiz questions and activity instructions). Students were asked prior to the first class if they had a laptop available to bring to class. All students reported that they did.

Each quiz consisted of 10 questions (mostly multiple choice). Students were given 10 minutes to complete the quiz then immediately begin a programming activity designed to fill the remaining 75 minutes of class time. The activity included a deliverable (typically a small working program) submitted to the LMS. The instructor was assigned two undergraduate teaching assistants with Java programming experience to help answer students' questions. During the semester, instructor S deployed 9 videos and administered 10 quizzes and 10 activities. During one week, the quiz and activity was based on the reading, as the instructor could not deploy a video in time. Three Tuesday lectures were used to review for exams or to give students special help with homework.

On Tuesday activity sessions, the instructor would use the podium computer and the LMS to monitor students' quiz grades as they were completed. Two to three quiz questions were designed to be trivially easy for those who watched the video. If a student got all the trivial questions wrong, the instructor would dismiss the student to a nearby lounge to review the reading and re-watch the video. Three students were dismissed twice in the first three weeks of the course. By the fourth week, these three students were getting the trivial questions correct. Two additional students were dismissed just once early in the semester. In general, the quiz setup proved valuable in immediately identifying students who were not watching the videos and helped correct this deficiency in a positive way. Students who did poorly on the quizzes were dismissed and lost valuable lecture activity time to get help from the instructor and student assistants. However, as long as students returned in a timely fashion, worked until the end of the period and submitted the activity by the next lecture period (Thursday), no penalties were given.

Thursday sessions were used for traditional lecturing, which instructor S used exclusively when teaching the course in Fall 2010. Traditional lecturing includes the presentation of lecture slides and programming demonstrations. By replacing 10 out of 25 (40%) lecture sessions with activities, instructor S yielded improvements in student evaluations. Table 2 compares student evaluation scores for the last time instructor S taught the course (i.e. Fall 2010), which was in a traditional format, versus the recent (i.e. Spring 2015) semi-flipped course. Instructor S did not teach this course between those two semesters. This change in student evaluation scores is the most significant improvement instructor S has achieved in 12 years of teaching. It is unlikely that such improvement is the result of simple instructor maturation.

Measure	S 2015	F 2010
Classes helpful	9.1	8.3
Classes interesting	8.9	7.5
Course rating	9.4	7.4

Table 2. Instructor S Student Evaluation Scores

In addition to these improvements, a majority of the semi-flipped classroom students agreed (9.2/10 on Strongly Disagree to Strongly Agree Likert-scale) with the statement, *I like the* *"flipped classroom" teaching model. i.e. Watching the lecture videos outside of class and spending more class time doing team exercises*. One student commented "I really found the video/activities helpful" (see Appendix A, Table I for complete comment summaries). But, not all students prefer the format (see Appendix A, Table II). One student stated "I really did not care for the videos at all. They really were just difficult to focus on and weren't very helpful." A vast majority of the 33 students strongly agreed (9.5/10 on Strongly Disagree to Strongly Agree Likert-scale) with the statement: *The in-class programming exercises improved my comprehension of the course material.* Not surprisingly, students tended to disagree (5.3/10 on Strongly Disagree to Strongly Agree Likert-scale) with the statement, *Prefer normal course over flipped*.

Fully-Flipped Classroom (F)

The fully-flipped section had 16 students, only 4 of which were CS majors. The instructor had created 54 lecture videos, each of which is approximately 10 minutes in length. The lecture videos were created using Camtasia. The videos included lecture narration of PowerPoint slides, coding demonstrations in BlueJ, and interactive quizzes. The quizzes did not count towards the students' grades; their purpose was to help the students assess their understanding of the concepts as they progressed through the videos. The flipped section's lecture was scheduled on Mondays and Wednesdays for 85-minutes each session. Before each class, students were required to watch ~2 lecture videos and read the related textbook sections. Finally, in addition to their textbook, students were required to bring a laptop to class. Similar to the semi-flipped section, students were asked prior to the first class if they had a laptop available to bring to class. All students reported that they did but again the department had "loaner" laptops available.

Instructor F used two techniques to strongly encourage students to watch lecture videos. First, students were required to enter their name and email address in order to watch the lecture videos. The instructor was then emailed a detailed report as to how much of the content each student watched. Students can play the video in the background and turn the sound off so certainly this is imperfect. Secondly, the videos contained secret phrases, which had to be submitted via a Blackboard quiz. The phrase could be given verbally or written on the slides and could change several times throughout the video; the intent being to check if students were actually paying attention to the video. Of course, students could share the phrase with each other so this is not foolproof either.

Each class session would begin by instructor F soliciting the students for questions regarding the lecture video and readings, which typically took only a few minutes. Once all questions were answered, the students took a short online quiz (via Blackboard LMS) to assess their knowledge based on the lecture videos and readings. Quizzes were automatically graded and solutions were given by the LMS. Students were encouraged to ask questions regarding the quiz. Quizzes were fairly low-stakes, similar to Bormann (2014). The quizzes only counted for 10% of their final grade and the 3 lowest quizzes were dropped. However, students were unaware that quizzes would be dropped until the last 2 weeks of the semester. The intention was to motivate them to watch the lecture videos and learn from the mistakes they made in the quizzes.

After completing the quiz and getting answers to their questions, the students would begin the in-class activity. The class activities were directly related to the concepts covered on that day's lecture videos and readings. Students were encouraged to work with their fellow classmates as they worked through the in-class activities. In addition to the instructor, a sophomore TA, that had taken the course last year, was available to help answer questions during class. The TA would complete the in-class activities a day or two before each class to prepare. Both the instructor and the TA would move about the room answering questions and checking on shyer/quieter students.

The class activities were developed to take the average student the entire class time to complete. Some students did require more time so all students were given until the beginning of the next class to submit the solution. All assignments were submitted via the LMS and graded within a few days so the students had quick and frequent feedback. Stronger students that could complete the activities quickly were allowed to leave if they completed all their work and submitted it for grading. In some cases, more challenging extra credit activities were offered as an option if they wanted to stay or they were free to use the time to work on homework. Some would even stay to help their fellow classmates work through the activity. This was encouraged but closely monitored to make sure the help was in the form of guidance not simply providing the solution. The only restriction was that students were not allowed to help each other with homework; all homework questions needed to be asked of the instructor or an official departmental tutor.

Instructor F had taught this course the previous year in a fully-flipped format with zero in-class lecturing, other than answering questions. Instructor F was concerned that students may not be as connected with the instructor as they would be in a more traditional setting. Therefore, the first three weeks of the Spring 2015 course, Instructor F began each class with ~15 minutes of lecturing to supplement the lecture videos. During the fourth week, students were anxious to jump right into the class activities so the instructor decided to go back to the fully-flipped format, only answering questions at the beginning of class and discontinuing traditional in-class lecturing. The only exceptions were two exam reviews. Students were given a practice exam to complete several questions and then the instructor led a short review/discussion of the solutions.

A majority of the fully flipped classroom students agreed (8.93/10 on Strongly Disagree to Strongly Agree Likert-scale) with the statement, *I like the "flipped classroom" teaching model. i.e. Watching the lecture videos outside of class and spending more class time doing team exercises*. One student reported the following: "I absolutely LOVE the flipped classroom. The lectures outside of class were short, sweet, and to the point and I actually enjoyed going to class. The class activities were extremely helpful in understanding the material" (also see Appendix A, Table I). Nonetheless, not all students prefer this method of learning (see Appendix A, Table II). As one student stated on the evaluation, "I didn't like the reversed classroom experience. I much rather prefer to listen to a lecture in class and then do the homework on my own time." A majority of students strongly agreed (9.71/10 on Strongly Disagree to Strongly Agree Likert-scale) with the statement: *The in-class programming exercises improved my comprehension of the course material.*

The frequent quizzes were helpful in assessing if students are watching the lecture videos and completing the readings. This was also a good way to give immediate feedback to students regarding their progress. However, the volume of quizzes can be overwhelming to the students. As one student commented, "I did not like that we had so many quizzes. We had them in the required lecture videos, after the lecture videos, and in class almost every day. At times it seemed like too much." Nonetheless the same student acknowledged the benefit by adding, "However, I do think they were helpful (just tedious)". Table 3 shows student evaluation data for the two semesters where instructor F taught the course in a flipped format.

Measure	S 2015	S 2014
Classes helpful	9.4	8.86
Classes interesting	9.27	8.95
Course rating	9.47	9.29

Table 3. Instructor F Student Evaluation Scores

The Spring, 2014 section had zero in-class lecturing, while the Spring 2015 had 8% of its in-class time devoted to more traditional lecturing. This included the 15 minute sessions at the beginning of class during the first 3 weeks of the course and the exam review sessions.

Another item to note is that the Spring 2015 section had a high number of IS students versus CS students. Many of these students in particular expressed concern about the difficulty of the course material during the first week, but after a few weeks of in-class coding practice they seemed to gain confidence and ultimately did quite well. Four of the five A's on the final exam were achieved by non-CS majors.

There may be some value in doing some traditional lecturing during the first few weeks of a flipped classroom environment to connect with the students. While more empirical data is needed to confirm this, instructor F did feel more connected with the students in the Spring 2015 course. Having a short lecture at the beginning of each class during the first 3 weeks allowed the instructor to learn all the students' names and get a better sense of their abilities. This also likely allowed the students to get to know their instructor better and perhaps made them more comfortable asking questions.

General Observations & Challenges

The three instructors agree that the flipped model slows down the pace of the course. In practice, the flipped classroom students spend more time on in-class activities than listening to lecture making it difficult to cover as much content as a traditional course. While in principle one can add more video lectures to cover more content, this is difficult to achieve for many reasons outlined below. In practice, our observation was that students in the traditional lecture get exposed to more content and the flipped students get less exposure but get more practice programming.

Videos are extremely time-consuming to make. It was not uncommon for it to take 5-6 hours to create a 10-minute video. Videos have no value unless students watch them. While students can also miss lecture, absences can be precisely measured whereas measuring whether or not students watch the videos is challenging. Instructors F and S used various techniques described in the previous section to mitigate this issue but no technique was perfect.

Videos can be more engaging than reading the textbook and embedded quizzes can help verify that a video is being watched, but ultimately there is no way to verify that students are thoroughly watching the videos. Similar to text book reading, the benefits depend on the individual learner.

If the fully- and semi-flipped students watch all the videos and typically use the entire class period or more to complete the in-class activities, they could have more "seat time" than students in a traditional setting. In other words, the explicit time spent on lecture-related activities could be higher. These students definitely get more programming practice, in the form of in-class activities, than traditional students typically get but it is not necessarily the case that they are spending more time on the course. For instance, some students were able to complete in-class activities quickly and either leave class early or move on to other tasks (e.g. homework or watching lecture videos). It is also certainly the case that some students simply did not watch the lecture videos, reducing their "seat time". Finally, unlike traditional classrooms, students were given time in-class to work on homework assignments. Since they are able to get help from their instructor during this time, this could have significantly reduced the amount of out of class time students spent on these assignments versus traditional classroom students.

Whether or not fully- and semi-flipped students are actually getting more seat time than traditional students depends on a variety of factors such as lecture video length, student willingness to watch the videos, and student ability. Nonetheless, fully- and semi-flipped students may perceive that they have more seat time than a traditional class and may not like it. But, the student evaluation data and comments did not reveal frustration with increased time commitment. However, some students did report not liking the lecture videos (see Appendix A, Table II). Some of this could be related to the length of the videos. Instructor F's videos were typically ~10 minutes each, while instructor S's videos were 40 minutes on average.

Perhaps most importantly, the lecture format did not have an impact on students' performance on the final exam, which was identical for all sections. Table 4 shows how close the mean final exam scores were for students in the three different lecture sections.

Instructor F	Instructor S	Instructor T
84.3 (n=16)	84.5 (n = 33)	83.4 (n=13)

Table 4. Mean Final Exam Scores

4. CONCLUSIONS & FUTURE WORK

There can be an incredible start-up cost to creating flipped classroom materials. Creating the lecture videos is very time consuming but they can certainly be reused. Bergmann (2012) recommends short interactive videos that are 1½ minutes in length per grade level -- under 20 minutes for an introductory college level course. While some instructors may use generic lecture videos to flip their classroom, Sams and Bergmann (2012) recommend that instructors create their own. One possible danger is tying lecture videos too closely to a specific textbook (e.g. using textbook slides). This could make it difficult to switch textbooks or even go to a newer edition.

Our comparative evaluation of flipped versus traditional was consistent with previous work. We observed that flipped learning provides students opportunities to learn in a more differentiated manner than traditional linear and passive forms, which is consistent with Willey and Gardner (2013). We observed and informally measured that the vast majority of students completed the required prerequisite tasks on a fairly regular basis while only a very small portion did not, which is consistent with Davies et al. (2013), Gaughan (2014), and Murphree (2014).

The semi-flipped model is an effective compromise for an instructor who cannot devote the time needed to replace all lectures with videos. Based on our observations, the semi-flipped model produces similar positive impressions from students without the need to replace all traditional lecturing. However, instructor S notes that early in the semester students were confused about the format and it took nearly 3 weeks before the vast majority of students were coming to the activity session prepared, whereas, instructor F reports a quicker turnaround time with respect to student comfort and preparation.

We observed that flipped learning empowered students through more active learning, which is consistent with (Lage et al., 2000). Specifically, students were held more accountable for studying material prior to coming to regular lectures. However, while we can conjecture that students in the semi- and fully-flipped sections may have more programming practice, and should therefore perform better when assessed, the average final exam scores shown in Table 4 indicate no significant difference in student performance based on the lecture format. More generally, whether or not semi-flipped yields a better understanding of the course material is yet to be determined.

It is also unknown what balance of flipped versus traditional works best and how to structure that balance. Instructor S had one day a week dedicated to lecture and one for in-class activities whereas Instructor F only did some short lecturing at the beginning of the semester and for exam review. While instructor T initially planned to stick with traditional lecturing in the classroom, the schedule forced some adjustment including incorporating programming activities during lecture time. Ultimately, this addition was well received by both the students and instructor.

The flipped model, whether full or semi, creates certain advantages that are difficult to achieve in a traditional model. For example, in a lecture only model, it is often not revealed that a student is lost until the first homework assignment or even the first exam. Doing poorly on these higher stakes items can have devastating effects on some students, crushing their confidence and willingness to keep trying. In a flipped or semi-flipped classroom, students can experience weekly or even daily feedback via low-stake activities and quizzes. This enables both students and instructors to get early and frequent feedback. Students who stumble along the way on low-stakes deliverables still have time to recover before higher stake exams. And, instructors can detect problems early and intervene quickly to get students back on track. In addition, as students work through the in-class exercises, the instructor can also detect students that need extra attention and identify general pain points for all students, which can then be addressed quickly via a short ad-hoc lecture or a supplemental lecture video.

While the quizzes and questions that encourage video watching are low-stakes with respect to grading, there are still consequences that encourage students to take them seriously. Many students naturally want to make a good impression with the instructor and are embarrassed if it is revealed that they are not prepared. When a student does poorly on a major test, it can be unclear to both the instructor and student if the poor performance is the result of a lack of understanding or a lack of preparation. Whereas, when a student fails to answer a trivial question that was directly answered in a video, the source of the student's problem is clearer to both student and instructor.

Another major advantage of the flipped classroom is that strong students are not held back during class activities. They are free to work as quickly as possible and, at least in the case of these course offerings, were allowed to leave if they completed all their work.

Based on our observations and survey results, the in-class activities associated with the flipped model had the most profound impact in improving students' impressions of the course, whereas the videos themselves were not viewed as positively. Thus, improvements can be achieved without flipping an entire course as instructor T and S learned by adding in-class activities without completely replacing traditional lectures with video. But, using lecture time for activities certainly decreased the amount of material that was covered in the course. This was true even when videos were used extensively. While video lectures can substitute for traditional lectures, in practice, this is challenging to achieve because some students may not learn the material as effectively from video lectures.

Since some students will prefer traditional lecturing, it might make sense to not only offer the different sections but also advertise them as such. This would allow students to self-select the teaching style that best fits their needs. Future research might include assessing student satisfaction after self-selection.

5. REFERENCES

Bergmann, J. & Sams, A. (2012). Flip Your Classroom: Reach Every Student in Every Class Every Day.International Society for Technology in Education.

Bishop, J. L., & Verleger, M. A. (2013, June 23-26). *The Flipped Classroom: A Survey of the Research.* Paper presented at the 120th ASEE Annual Conference & Exposition, Atlanta, GA.

Bormann, J. 2014. Affordances of Flipped Learning and its Effects on Student Engagement and Achievement. Doctoral Dissertation, University of Northern Iowa. Chicago.

Clark, K. R. (2015). The Effects of the Flipped Model of Instruction on Student Engagement and Performance in the Secondary Mathematics Classroom *Journal of Education, 12*(1), 91-115.

Danker, B. (2015). Using Flipped Classroom Approach to Explore Deep Learning in Large Classrooms. *IAFOR Journal of Education, 3*(1), 171-186.

Davies, R. S., Dean, D. L., & Ball, N. 2013. Flipping the classroom and instructional technology integration in a college-level information systems spreadsheet course. Educational Technology Research and Development, 61(4), 563-580.

Findlay-Thompson, S., & Mombourquette, P. (2014). Evaluation of a flipped classroom in an undergraduate business course. *Business Education and Accreditation, 6*(1), 63-71.

Frydenberg, M. (2013). Flipping Excel. *Information Systems Education Journal*, 11(1) pp 63-73.

Gaughan, J. E. (2014). The Flipped Classroom in World History. *The History Teacher, 47*(2), 221-244.

Gunyou, J. (2015). I Flipped My Classroom: One Teacher's Quest to Remain Relevant. *Journal of Public Affairs Education, 21*(1), 13-24.

Herreid, C. F., & Schiller, N. A. (2013). Case Studies and the Flipped Classroom. *Journal of College Science Teaching, 42*(5), 62-66.

Lage, M. J., Platt, G. J., & Treglia, M. (2000). Inverting the classroom: A gateway to creating an inclusive learning environment. *The Journal of Economic Education, 31*(1), 30-43.

Michail N. Giannakos, John Krogstie, and Nikos Chrisochoides. 2014. Reviewing the flipped classroom research: reflections for computer science education. In Proceedings of the Computer Science Education Research Conference (CSERC '14), Erik Barendsen and Valentina Dagiené (Eds.). ACM, New York, NY, USA, 23-29.

Murphree, D. S. 2014. "Writing wasn't really stressed, accurate historical analysis was stressed": Student Perceptions of In-Class Writing in the Inverted, General Education, University

History Survey Course. History Teacher, 47(2), 209-219.

Papadopoulos, C., & Roman, A. S. 2010. Implementing an inverted classroom model in engineering statics: Initial results. In proceedings of the American Society for Engineering Education (ASEE).

Saulnier, B. (2015). The Flipped Classroom in Systems Analysis & Design: Leveraging Technology to Increase Student Engagement. *Information Systems Education Journal*, 13(4) pp 33-40.

Strayer, J. F. (2012). How Learning in an Inverted Classroom Influences Cooperation, Innovation, and Task Orientation. *Learning Environment Research, 15*(2), 171-193.

Vaughan, M. (2014). Flipping the Learning: An Investigation into the use of the Flipped Classroom Model in an Introductory Teaching Course. *Education Research and Perspectives, An International Journal, 41*, 25-41.

Willey, K., & Gardner, A. 2013. Flipping your classroom without flipping out. Paper presented at 41 SEFI Conference, Retrieved:https://www.academia.edu/5153284/Flipping_your_classroom_without_flipping_out

Zhang, D., Zhou, L., Briggs, R. O., & Nunamaker, J. F. (2006). Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Information & Management, 43*(1), 15-27.

Editor's Note:

This paper was selected for inclusion in the journal as a EDSIGCon 2015 Meritorious Paper. The acceptance rate is typically 15% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2015.

Appendix A – Sample Quizzes and Activities

Sample Quiz Questions from Semi-Flipped Class (week 2)

Question	Consider these four lines of code: a. `JPanel panel = new JPanel();` b. `ImageIcon me = new ImageIcon("eric.jpg");` c. `JLabel image = new JLabel(me);` d. `panel.add(image);` Which line of code (a, b, c or d) uses a default constructor to create a new object with default values?

Question	Consider these four lines of code: a. `JPanel panel = new JPanel();` b. `ImageIcon me = new ImageIcon("eric.jpg");` c. `JLabel image = new JLabel(me);` d. `panel.add(image);` Which line of code (a, b, c or d) uses a default constructor to create a new object with default values?

Sample Activity Segment from Semi-Flipped Class (week 2)

1. Download BasicPrograms.zip and save to an appropriate folder on your computer.
2. Unzip/extract the *BasicPrograms* folder. On a PC, right-click the zip file and select `Extract All`. On a MAC, double-click the zip file to extract the folder.
3. In BlueJ, open the *BasicPrograms* project. Select `Project --> Open Project` and then find the *BasicPrograms* project folder.
4. Modify the `AdLib` program so that it creates a humorous "ad lib" story. AdLib example
5. You should prompt the user for at least 7 words (nouns, verbs, adjectives, etc.) and output at least two sentences. Be creative and have fun, but do not spend more than 25 minutes on this part.
6. You should prompt the user for a particular word using `JOptionPane.showInputDialog` and store the word using a String variable.
7. You should output each sentence with a separate `System.out.println` statement.

Sample Quiz Questions from Semi-Flipped Class (week 11)

Question	If you did not know the exact size of an array called nums, how could you access the last value?
Answer	nums[nums.length]
	nums[nums.size]
	✔ nums[nums.length-1]
	nums[nums.size-1]

Question	Which code shows how to declare an array called nums for storing 10 integer values?
Answer	int nums = new array[10];
	✔ int[] nums = new int[10];
	int[10] nums = new int[];
	int array[10] nums;

Sample Activity Segment from Semi-Flipped Class (week 11)

1. In the array project, create a new class called `IfStatement`
2. Replace all the code inside the class with one `main` method that does the following:

    - Prompt the user to type the temperature and their mood
    - Store the temperature as a double and the mood as a String
    - Print "play" if the user's mood equals "happy" and the temperatures is between 50 and 90 (inclusive)
    - Print "play" if the user's mood is not equal to "happy" and the temperatures is between 75 and 85 (not-inclusive)
    - Print "stay inside" if the user's mood equal "sad" or "average" and the temperatures is below 75.
    - Print "unsure" if all of the above conditions are false.

Appendix B – Student Evaluation Open-Ended Questions

Table I: Liked about course/instructor

Traditional	• He was enthusiastic about the class and took the time to explain how to make our code better. He would help us with any problems on our activities as well and explain the solutions thoroughly. • The in class activities were key. That's where I learned most of the material. • I enjoyed the activities that the professor gave at the end of class. I found that they really helped me learn the material. • I think this way of running the class was effective.
Semi-Flipped	• Interacting and working with classmates at least once a week was helpful and productive. • I really found the video/activities helpful and his attitude toward wanting each individual student to succeed. • I liked the activity sessions of the course, and I thought that they were very helpful. • I think that the course was fine. I liked the lecture style of the instructor and it helped me a lot. • I learned a lot despite the incredibly fast pace. If I didn't understand something, the assignments were designed to let me figure it out on my own. • the homework. I like to do the homework
Fully-Flipped	• exercises in class helped alot • I really enjoyed the flipped classroom setting. The videos gave me the right amount of information and then being able to go right into class and do the exercises helped to make the concepts clearer. Also hiving the time do homework in class was very helpful, the homework was challenging, and having the ability to ask questions made it less frustrating • I liked that the class is completely hands on, with this type of course it is extremely helpful to be constantly practicing the material because being lectured on it won't make sense unless you are actually being challenged to figure out what the material means. • I liked that it was a lot of hands-on activities rather than lectures, I think it suits well with the course material. • I enjoyed the … and the flipped classroom • The homework assignments were the best part of the class because it really tested you on the material. • I really liked the format. Watching the videos before class and then getting straight to work. In practice it was kind of like having two lab classes. • I absolutely LOVE the flipped classroom. The lectures outside of class were short, sweet, and to the point and I actually enjoyed going to class. The class activities were extremely helpful in understanding the material.

Table II: Disliked about course/instructor

Traditional	• I did not like the quizzes we had in the beginning, those were tough. Also, I wish the activities counted for more than just participation points. • Progressed through concepts slowly. • Sometimes the lectures seemed unorganized but I think that is because we didn't go over every slide and instead skipped around. All the important material was still taught so it was not much of a problem.
Semi-Flipped	• Online videos were an interesting idea but it was difficult to understand the concepts at times and the quizzes were sometimes difficult. • I really did not care for the videos at all. They really were just difficult to focus on and weren't very helpful. • I also feel like there was too much material in the traditional lectures. I think that watching someone code for an hour and a half doesn't really help me at all. • I think that it would have been more helpful if the material had been presented and then we worked with partners or in small groups to practice the code. • There definitely needs to be traditional lectures to introduce things like arrays, Java in general, etc, but I really think that watching someone else doing a program really isn't practicing anything.
Fully-Flipped	• videos outside of class • I did not like that we had so many quizzes. We had them in the required lecture videos, after the lecture videos, and in class almost every day. At times it seemed like too much. However, I do think they were helpful (just tedious). • I didn't like the reversed classroom experience. I much rather prefer to listen to a lecture in class and then do the homework on my own time.

A Tale of Two Curricula: The Case for Pre-requisites in the IS Model Curriculum

John H. Reynolds
john.reynolds@gvsu.edu

Roger C. Ferguson
roger.ferguson@gvsu.edu

Paul M. Leidig
paul.leidig@gvsu.edu

School of CIS
Grand Valley State University
Allendale, Michigan 49401, USA

Abstract

The most recent Information Systems (IS) Model Curriculum recommendations is IS2010. While the goal of this revision was to update the curriculum from IS2002, the end result was a change in curriculum design philosophy whereby a pre-requisite structure that fostered increasing depth of knowledge was flattened to make the curriculum easier to traverse for the student. At the same time, the number of core courses was reduced from ten to seven by either combining subject matter or eliminating content. This paper examines the usefulness of having perquisites to increase the student's "depth of knowledge" and explores how to analyze the need for those pre-requisites. The data show that five years after the release of IS2010, ABET accredited IS Programs in business schools seem to be embracing the underlying philosophy of IS2010. On the other hand, ABET accredited IS Programs outside business schools continue to embrace the curriculum design philosophy of IS2002. The IS community is now at a critical juncture due to these two differing curriculum design philosophies, both in terms of curriculum content and assessment methods.

Keywords: Model Curriculum, Pre-requisites, IS2010, IS2002

1. INTRODUCTION

Many papers have been written on the subject of curriculum design in Information Systems, including the use of course pre-requisites. While these papers have discussed and suggested using course pre-requisites, none have really addressed the underlying rationale for having pre-requisites.

> One of the most prevalent problems in course and curriculum design is the tendency of faculty to make false assumptions about the knowledge and skills that students bring to their courses. These incorrect assumptions lead to failure for the students who are ill prepared, boredom for their classmates who are often more than adequately prepared, and frustration for the faculty (Diamond, 2008).

In the Information Systems (IS) discipline, the need for this type of discussion is clearly seen in a side-by-side comparison of the course architecture in the current IS2010 Model Curriculum shown in Figure 1, which illustrates a flattened curriculum structure (Topi, Valacich, Kaiser, Nunamaker, Jr., Sipior, de Vreede, G. &

Wright, R., 2010), herein after referred to as IS2010.This is compared to its predecessor IS2002 shown in Figure 2 (Gorgone, Davis, Valacich, Topi, Feinstein, & Longenecker, Jr., 2002), herein after referred to as IS2002, which illustrates a more hierarchical curriculum structure.

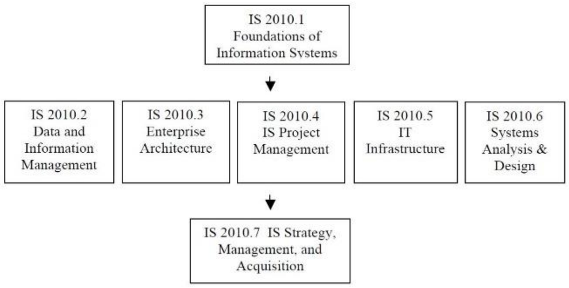


Figure 1 – IS2010 Structure

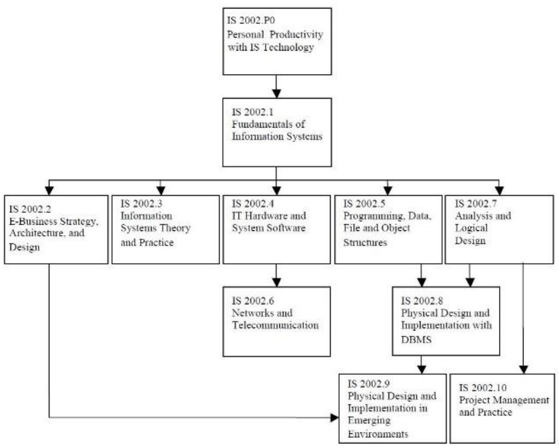


Figure 2 – IS2002 Structure

Bloom's Taxonomy and it's adaption in the IS Model Curriculum shows how pre-requisites play an important role in defining "depth of knowledge" throughout the curriculum models. This paper then compares the required courses of ABET accredited IS Programs in business schools with those ABET accredited IS Programs outside business schools for both their pre-requisite structure and overall content.

2. PRE-REQUISITES AND DEPTH OF KNOWLEDGE

One paper defined a pre-requisite as "the skills and information necessary to succeed in a given instructional unit within a curriculum." (Young, 2011). Regarding programming courses, Walker (2010) notes that "upper-level courses commonly expect students to have mastered the basics of programming at the beginning level." A typical pre-requisite, referred to by some as a direct pre-requisite, takes the form of requiring a class (or set of classes) prior to taking a course. Direct pre-requisites typically target specific skills that are needed (or believed to be needed) for the advancement into the next course. For example, the pre-requisite for a Database II course would be a Database I course.

Inherent Challenges

This section examines the problems that pre-requisites cause institutions, curriculum developers, and students. From a student's perspective, any pre-requisite could cause a delay in graduation and in some cases, a significant delay, depending on the availability of the particular course. If a course is offered only in one term and its pre-requisite is only offered in the preceding term and is full, a student's graduation could be delayed an entire year.

For curriculum developers, pre-requisites that are inserted into a model curriculum could cause institutions not to adopt the model curriculum. In other words, the ability, in terms of cost and resources for an institution to adopt a model curriculum that has a significant pre-requisite structure would be higher than to adopt a model curriculum with a flattened (minimal) pre-requisite structure. Higher costs are incurred when scheduling for faculty and rooms is more difficult since courses must be offered to allow students to complete the sequences in a timely manner. Students, too, are burdened with more complicated schedules and potentially longer times to graduate due to full/conflicting class schedules, thereby potentially resulting in lower student enrollment. Simply put, having a significant pre-requisite structure in your model curriculum causes roadblocks for students, institutions and curriculum developers, but what is the cost to the student's education and career success?

Bloom's Taxonomy in the Model Curricula

A pre-requisite is determined to be useful for a course if that pre-requisite gives the course the ability to have the student reach a higher level of ability on Bloom's taxonomy as modified and articulated in the Appendix 3 of IS2010 (see Table 1) and referred to as a Depth of Knowledge Metric (DKM). For example, a pre-requisite of a Database I course for a Database II course would be useful if and only if Database I is required to allow a student in Database II to reach a higher level in the DKM. Otherwise, the course (i.e., Database I) does not meet the definition of useful and may not be a wise use of a pre-requisite. The following section shows how different pre-requisites structures, i.e. having or not having a significant pre-requisite structure, affect a student's ability to reach higher levels knowledge.

3. STRUCTURE OF IS2002 VS IS2010

Referring to Figures 1 and 2 above, the IS2002 Model Curriculum has three significant two-course pre-requisite sequences that do not appear in IS 2010: Hardware/Software (HW/SW) to Networking, Programming to Database, and Systems Analysis and Design (SAD) to Project Management. In the case of the first sequence, not only isn't the pre-requisite required for Networking, IS2010 combines HW/SW and Networking into one course.

To validate the sequences described above in IS2002 and following the overall methodology suggested by Vuong, Nixon, and Towle (2011), a further analysis of the Programming to Database sequence shows that the course labeled IS 2002.8 – "Physical Design and Implementation with DBMS" has a pre-requisite of IS 2002.5 – "Programming, Data, File and Object Structures". The description of IS2002.8 states

> Students will demonstrate their mastery of the design process *acquired in earlier courses* by designing and constructing a physical system using database software to implement the logical design.

Based on this description there is a clear requirement of comprehension (level 2 in the DKM) because the phrase "acquired in earlier courses." In this case the course IS2002.5 (Programming, Data, File and Object Structure) is where the students would have acquired that knowledge. The description of IS2002.5 states

> Students will gain in-depth understanding of defining and measuring events that produce data, both simple and complex, and principles, concepts, and practices of successful software development.

The description of this course uses terms that are taken directly from the DKM indicating a Level 1 depth of knowledge. Given the definition above, the pre-requisite of IS2002.5 is useful for IS2002.8 because the student moves to a higher level in the DKM, and without this pre-requisite the student would not reach that higher level.

While these three two-course sequences are the primary focus of this study, a further analysis of the two model curricula in their entirety also supports the proposition that each follows a different curriculum design philosophy, particularly as it relates to pre-requisites in the core.

To reach IS2002.9 a student will have to take the following sequence first: IS2002.0, IS2002.1, IS2002.5 and IS2002.8. The benefit of this long sequence of classes is that students, after completing IS2002.9, reach Level 3 (Application) in the DKM. This high level could not have been reached without pre-requisites and hence the pre-requisites are useful.

In contrast, analysis of the IS2010 model curriculum shows a more flattened structure. Even though senior standing might be assumed, to reach IS2010.7 (the capstone course), according to the curriculum model, a student only has to take in sequence IS2010.1 and IS2010.4. Based on the same analysis of their use of terms from the DKM in their course descriptions, both of these courses, IS2010.1 and IS2010.4, give students knowledge (Level 1) and hence satisfy the definition of usefulness above. However, by that same standard students only reach level 2 (comprehension) in the DKM.

This minimal set of pre-requisites allows students some flexibility in scheduling of courses and allows students an easier time to fit in all of their classes before graduation. In addition, this minimal set may even help cut cost for the institution. However, students will reach a higher level in the DKM when useful pre-requisites are utilized. Other institutions have come to this same realization with both research and teaching experience indicating that a lack of ongoing integration between courses creates a learning barrier. If a subsequent course's concepts do not begin where preceding ones end, "students lose sight of the overall goal of the curriculum" (McGann, Frost, Matta, & Huang, 2007).

4. COMPARING ACCREDITED IS PROGRAMs

Since the publication of IS 2010, several journal articles have collected data from IS programs to evaluate those programs against the IS 2010 Model Curriculum. One article noted the need for updating programs to the current Model Curriculum (Apigian and Gambill, 2010), another demonstrated a very detailed research design to classify and evaluate programs (Mills, Velasquez, & Fadel, 2012), and another sought to analyze the adoption rate of the new model curriculum (Bell, Mills, & Fadel, 2013). Unfortunately, none of these sought information from any IS programs that reside in academic units outside business schools.

The purpose of this study is to evaluate IS programs by matching their curriculum structure to the curriculum design philosophy of either IS2002 or IS2010. The authors chose ABET accredited IS programs with the understanding those programs ostensibly share a common baseline, while at the same time including programs outside of business schools with those in business schools, since ABET accreditation is complimentary to AACSB accreditation (Hilton and Lo, 2007).

The programs were then divided into two groups: those programs that were housed in a business school and those programs that were outside a business school. There are 15 institutions in the first group and 24 institutions in the later (see Appendix A). One institution had a combined undergrad/grad program and was excluded from the sample.

For each program, the core curriculum requirements of the IS program were evaluated to see if any of the three two-course pre-requisite sequences noted above (see Section 3) existed; hence, the following hypotheses were generated.

1) Does the first Networking course have a pre-requisite Hardware/Software (HW/SW) course?

H_1: Information Systems Programs outside business school will be different in the percentage that have a Hardware/Software (HW/SW) course as a pre-requisite to a first Networking course.

2) Does the first Database course have a pre-requisite Programming course?

H_2: Information Systems Programs outside of a business school will be different in the percentage that have a Programming course as a pre-requisite to a first Database course.

3) Does the first Project Management course have a pre-requisite Systems Analysis and Design course?

H_3: Information Systems Programs outside of a business school will be different in the percentage that have a Systems Analysis and Design course as a pre-requisite to the first Project Management course?

One might argue that it is difficult to have a course as a pre-requisite if the course is not part of the required courses in the IS program, which naturally leads to a further set of hypotheses regarding only the courses that are not in both model curricula. For the sake of brevity, they are combined into one question and subsequent separate hypotheses:

4) Are any of the individual courses in the previous hypotheses (HW/SW, Networking, and Programming) part of the required courses?

H_4: Information Systems Programs outside business school will be different in the percentage that require a HW/SW course.

H_5: Information Systems Programs outside business school will be different in the percentage that require a Networking course.

H_6: Information Systems Programs outside business school will be different in the percentage that require a Programming course.

Statistical Methodology

For each of the three hypotheses, the null hypothesis will be accepted or rejected using the significance level of .05. To compare two independent groups based on binary variables, most statistics guidelines suggest using the chi-square test of independence as long as the sample sizes are large enough. Sauro and Lewis (2008) contend, however, that the "latest research suggests that a slight adjustment to the standard chi-square test, and equivalently to the two-proportion test, generates the best results for almost all sample sizes" (p. 75).

To determine whether a sample size is adequate for the chi-square test, calculate the expected cell counts in the 2x2 table to determine if they are greater than 5. When the values in this study met this test, the chi-square test results were used. When the values of one or the other of the subgroups did not meet this test, the N-1 chi-square test was used. The formula for the N-1 chi-square test (Sauro and Lewis, 2008) is shown in the next equation using the standard terminology from the 2x2 table:

$$\chi^2 = \frac{(ad - bc)^2 (N-1)}{mnrs}$$

When the values for both groups in the study failed to meet the threshold, the more conservative Fisher Exact Test was used. The formula for this test is also given by Sauro and Lewis:

$$\rho = \frac{m!\,n!\,r!\,s!}{a!\,b!\,c!\,d!\,N!}$$

Test Results
Hypotheses are supported when the null hypothesis is rejected. In this study, the null hypothesis is rejected when there is a statistically significant difference between the proportions represented by p<.05. Accordingly, the first hypothesis (H_1) is supported since there is a significant difference between the 7% of Business School IS Programs and the 42% of IS Programs outside a business school that require a Hardware/Software course as a pre-requisite to a first Networking course. The second hypothesis (H_2) is also supported since there is a significant difference between the 53% of Business School IS Programs and the 88% of IS Programs outside a business school that require a programming course as a pre-requisite to a first Database course. Although the programs outside of business schools had a higher percentage requiring a pre-requisite of an analysis and design course, the third hypothesis (H_3) is rejected since there is no significant difference. Chart 1.0 shows the comparison of the proportions for these three course pre-requisite sequences.

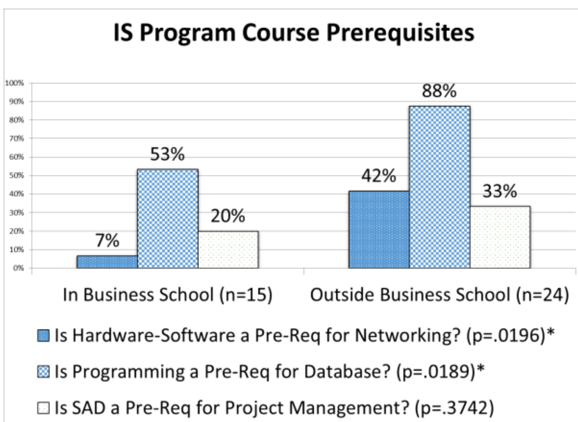


Chart 1.0 – Course Pre-requisites

The fourth hypothesis (H_4) is accepted as there is a significant difference between the 7% of IS Programs in business schools and the 58% of IS Programs outside business schools that require a HW/SW course. H_5 and H_6 are rejected. It is worth noting that significant statistical differences remain the same between the two groups for H_1, and H_2, even when the data from the individual course results are factored in as dependent variables.

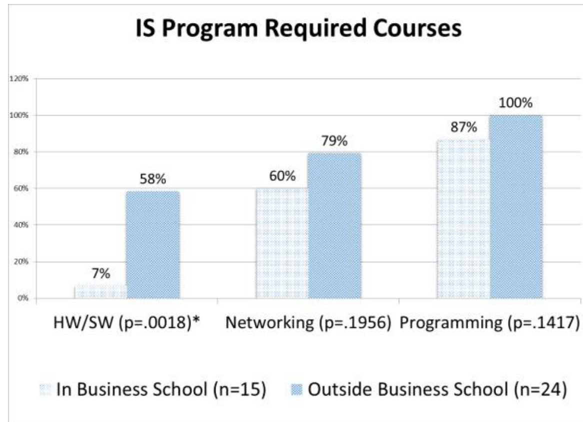


Chart 2.0 Required Courses

Of those IS programs that are in a business school and did not offer a HW/SW class, 6 have created the IT Infrastructure course proposed in IS2010 that essentially combines what IS 2002 called HW/SW and Networking into one class. Two of the IS programs outside of a business school have also created an IT Infrastructure course.

Database is listed in both model curricula as a required core course, and only one school out of all the schools in the study does not require a Database course (one school combines Database and Networking). In contrast, Project Management is listed as a core course in both model curricula, but only 53% of all the schools in the study require this class in the core. Even among those programs, there is little agreement on what should be the pre-requisite – there are as many that require database as those who require SAD and several programs require both.

Lastly, it is worth noting over 25% of the business schools in this study are AACSB accredited – one does not mention any special accreditation and three are accredited by the ACBSP.

5. CONCLUSIONS

The curriculum design philosophy of the pre-requisite structure is significantly different between the IS2002 and IS2010 model curricula. A student graduating from an institution that models their program after IS2002 will, by design, have a greater depth of knowledge in specified knowledge areas where there is a prescribed pre-requisite structure compared to a student graduating from an institution that model their program after IS2010.

The rejection of H_1 and H_2 show a propensity of business school IS programs toward adopting a flatter pre-requisite structure and the rejection of H_4 shows a trend toward fewer technical core courses, thereby implying agreement with the IS2010 philosophy. On the other hand, IS programs outside business schools have shown a desire to continue with the IS2002 philosophy, both in the pre-requisite structure and inclusion of technical core courses. While many of the business schools would argue that they do make these courses available as electives, this study was focused only on the required courses of every student that graduates from a program.

The data show that the IS community is now at a critical juncture. Previous efforts at a unification of the differences in the IS community have apparently failed, as this was the stated goal of the IS2010 authors, yet the data suggest that this unity is a myth. This paper shows the validity of the efforts of some in the IS community to develop a sister model curriculum (Waguespack, ISECON 2014), acknowledging the fact that there are two different IS program philosophies and goals.

In the end, neither philosophy is better or worse than the other, but these differences will eventually affect program identities and assessment. It would be disingenuous for one program to be classified and/or evaluated by the standards of the other, therefore these differences must eventually be acknowledged in future accreditation and assessment standards.

6. REFERENCES

Apigian, Charles H. and Gambill, Stanley E. (2010). Are we Teaching the IS 2009* Model Curriculum? *Journal of Information Systems Educators*; 21(4), 411-420.

Bell, C., Mills R., & Fadel, K. (2013). An Analysis of Undergraduate Information Systems Curricula: Adoption of the IS 2010 Curriculum Guidelines. *Communications of the Association for Information Systems,* 32(2), 73-94.

Carlsson, S., Hedman, J., & Steen, O. Integrated (2010). Integrated Curriculum for a Bachelor of Science in Business Information Systems (BISD 2010). *Communications of the Association for Information Systems,* 26(24), 525-546.

Diamond, Robert M. (2008). Designing and Assessing Courses and Curricula: A Practical Guide, 3rd ed. Jossey-Bass, San Francisco.

Gorgone, J., Davis, G., Valacich, J., Topi, H., Feinstein, D. & Longenecker, Jr., H. (2002). IS 2002: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. ACM, AIS, and AITP, New York, NY and Park Ridge, IL.

Hilton, T. & Lo, B. (2007). IS Accreditation in AACSB Colleges via ABET. *Journal of the Association for Information Systems*, 8(1), 1-15.

McGann, S., Frost, R., Matta, V. & Huang, W. (2007). Meeting the Challenge of IS Curriculum Modernization: A Guide to Overhaul, Integration, and Continuous Improvement. *Journal of Information Systems Educators*, 18(1), 49-62.

Mills, R., Velasquez, N. & Fadel, K. (2012). Examining IS Curriculum Profiles and the IS 2010 Model Curriculum Guidelines in AACSB-Accredited Schools. *Journal of Information Systems Educators,* 23(4), 417-428.

Topi, H., Valacich, J., Kaiser, K., Nunamaker, J., Jr. Sipior, J., de Vreede, G. and Wright, R. (2010). IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. ACM and AIS, New York, NY and Park Ridge, IL.

Vuong, A., Nixon, T., & Towle, B. (2011). A Method for Finding Pre-requisites in a Curriculum, *EDM 2011*. Retrieved July 6, 2015 from http://educationaldatamining.org/EDM2011/wp-content/uploads/proc/edm2011_paper8_short_Vuong.pdf.

Walker, Henry M. (2010). Pre-requisites: Shaping the Computing Curriculum. *ACM Inroads*, 1(4).

Knowledge Levels, Templates for Objective Writing, and Meaning of the Depth Levels with Associated Learning Activities

IS'90,'94,'95, 2002, 2010 Depth of Knowledge	**Bloom Levels of Knowledge**	**Template for Writing Behavioral Objectives Students completing ... will be able to**	**Meaning of Depth of Knowledge Level and Activities Associated with Attaining that Level**
0 No Knowledge			
1 Awareness	1 Knowledge Recognition	Define ... List characteristics of ... Name components of ... Diagram ... List advantages/disadvantages of ...	Introductory Recall and Recognition Class presentations, discussion groups, reading, watching videos, structured laboratories. Involves only recognition, but with little ability to differentiate. Does not involve use.
2 Literacy Strong Knowledge	1 Differentiation in context	Compare and contrast ... Explain ... Write/execute simple ... Define functional capabilities that are ... Describe interrelations of ... to related objects	Knowledge of Framework and Contents, Differential Knowledge Continued lecture and participative discussion, reading, team work and projects, structured labs. Requires recognition knowledge as a pre-requisite. Requires practice. Does not involve use.
3 Concept/Use Skill	2 Comprehension Translation/ Extrapolation Use of Knowledge	Use ... Communicate the idea of ... Form and relate the abstraction of ... as ... Given a set of ..., interpolate/extrapolate to ... List concepts/major steps in ...	Comprehension and Ability to Use Knowledge *when Asked/Prompted* Requires continued lab and project participation, presentation involving giving explanations and demonstrations, accepting criticism; may require developing skills in directed labs.
4 Detailed Understanding, Application Ability	3 Application Knowledge	Search for correct solution to ... and apply it to ... Design and implement a ... for ... Write syntactically correct ... and/or debug ... Apply the principles of ... to ... Implement a ... and maintain it	Selection of the Right Thing and Using It *without Hints* Semi-structured team-oriented labs where students generate their own solutions, make their own decisions, commit to and complete assignments, and present and explain solutions.
5 Advanced	4 Analysis 5 Synthesis 6 Evaluation	Develop/originate/institute ... Construct/adapt ... Generate novel solutions to ... Come up with new knowledge regarding ... Evaluate/judge the relative value of ... with respect to ...	Identification, Use and Evaluation of New Knowledge An advanced level of knowledge for those very capable of applying existing knowledge in which *denovo* solutions are found and used in solving and evaluating the proposed new knowledge.

Table 1 – IS 2010 Depth of Knowledge Metric (DKM)

Appendix A - List of Universities with ABET Accredited IS Programs Grouped by Academic Unit Location (as of May 2015)

Business Schools

1. East Tennessee State University (AACSB)
2. Gannon University (AACSB)
3. Kennesaw State University (AACSB)
4. Lock Haven University of Pennsylvania (ACBSP)
5. Metropolitan State University of Denver
6. Quinnipiac University (AACSB)
7. Rowan University (AACSB)
8. Slippery Rock University (ACBSP)
9. The University of Tampa (AACSB)
10. University of Central Missouri (AACSB)
11. University of Houston - Clear Lake (AACSB)
12. University of North Alabama (ACBSP)
13. Virginia Commonwealth University (AACSB)
14. West Texas A&M University (AACSB)
15. Wright State University (AACSB)

Non-Business Schools

1. Arkansas Tech University
2. California State University, Chico
3. California University of Pennsylvania
4. City University of Seattle
5. Drexel University
6. Fitchburg State University
7. Florida Memorial University
8. Grand Valley State University
9. Illinois State University
10. Jacksonville State University
11. New Jersey Institute of Technology
12. Pace University
13. Radford University
14. Regis University
15. Robert Morris University
16. Southern Utah University
17. State University of New York at Brockport
18. University of Houston
19. University of Nebraska at Omaha
20. University of North Florida
21. University of Scranton
22. University of South Alabama
23. University of South Carolina
24. Utah Valley University

The Case for Inclusion of Competitive Teams in Security Education

Anthony Serapiglia
Anthony.Serapiglia@stvincent.edu
CIS Department
St. Vincent College
Latrobe, PA 15650

Abstract

Through industry news as well as contemporary reporting, the topic of computer security has become omnipresent in our daily lives. Whether the news is about corporate data breaches, international cyber espionage, or personal data compromises and identity theft – EVERYONE has had to deal with digital security in some way. Because of this, one of the fastest growing areas of need in the CIS discipline and workforce is for skilled and knowledgeable security workers, and a gap has formed that has left hundreds of thousands of jobs unfilled possibly compromising overall security even more. While nothing trumps actual experience, it is possible to instill some experience into undergraduate students through simulations and especially cyber-security competitions. Inclusion of a competitive team in a second level security course is shown to increase student satisfaction with the course material, instructor effectiveness, and student perception of preparedness in the field.

Keywords: Security, Hacking, CyberSecurity, Competition, Curriculum, Pedagogy

1. INTRODUCTION

"Never trust a dynamiter who has all his fingers…." Old railroad saying…

In the early 2000's, the path to a security job in computing/information assurance/networking was through experience. It was common to see job postings that required 10 years of experience or more for anything that related to security. The positions of Chief Security Officer (CSO) and Chief Information Security Officer (CISO) simply did not exist. In 2015, by modest estimates, more than 209,000 cybersecurity jobs in the U.S. are unfilled, and postings are up 74 percent over the past five years (Carapezza, 2015; Resa, 2014).

According to the United States Department of Labor, the job outlook for Information Security Analysts predicts a growth rate of close to 40% through the year 2022. (BLS, 2014) In a changing atmosphere of perception and understanding of how pervasive security must be within organizations, what used to be categorized as an entry level Systems Administrator position is now often categorized as a security job. The growth in need for workers who are ready to fill these positions has far outstripped the pool of those with ten years of experience. This gap has presented to Computer and Information Science (CIS) programs a common problem facing many disciplines, how best to take the "ten years of experience" and instill that knowledge in a usable form into our students. The purpose of this paper is to highlight how the inclusion of a competitive team participating in an interscholastic cyber-security competition into an advanced security course affected student course evaluations in six categories: "Course work contributes to objectives", "Feel challenged and motivated", "Stimulates interest", "Makes me feel involved", "Effective Instructor", and "Successful course" while also providing practical experience in cybersecurity and information systems management.

2. ENVIRONMENT REVIEW

The year of 2014 saw an unprecedented wave of computer/information/network security events in the headlines and international news. Security breaches at retailers continued to a point where generic headlines became 'fill in the blank' templates of store name and how many personal records were compromised. After the Target point of sale (POS) breach during the holiday shopping season of late 2013 (Vijayan, 2014), so many other retailers were compromised; from Home Depot (Krebs, 2014) and Neiman Marcus (Katz, 2014) to UPS (Hardekopf, 2014) and the State of New York (Virtanen, 2014); that Forbes magazine began a running web listing of the top 20 data beaches of 2014 (Forbes, 2014).

Internationally, the Attorney General of the United States took the unprecedented step of charging five Chinese military hackers for "Cyber Espionage against U.S. corporations and a labor organization for commercial advantage (DOJ, 2014)." In November of 2014, Sony Pictures Entertainment became the victim of a hacking campaign that saw reportedly 100 terabytes of data stolen and public release of many sensitive and embarrassing documents and e-mails in retaliation for "The Interview" a movie lampooning an assignation attempt of Kim Jong-un the leader of North Korea. While some debate still remains on North Korea's direct involvement (Zeter, 2014; Kopan, 2014), Director of Homeland Security Jeh Johnson, FBI Director James Comey, and U.S. Secretary of State John Kerry all have made statements condemning North Korea for the "provocative and unprecedented attack (Kerry, 2014)" with eventual economic sanctions levied by the United States against North Korea because of the attack (Lederman, 2015). From an individual standpoint, 2014 also saw "The Fappening", a breach of Apple's cloud services that led to over 500 private pictures and videos of celebrities made public through a targeted attack of their phone backups held in cloud storage (Alexander, 2014; Hamil 2015).

With so many incidents affecting so many different areas, from government to corporate to individuals, 2014 seemingly became the year in which the "black hats" had a decisive edge against the "white hats".

3. ORGANIZATIONAL EFFORTS

Security is amazingly amorphous in how hard it is to define directly and with precision. While its presence today is ubiquitous in our collective consciousness and daily routine, it is also ever sifting in the form that threats take and the severity to which they expose us. Once mocked and criticized, Donald Rumsfeld's statement on collecting data in support of existence of state sponsored terrorism in Iraq has become much used in risk assessment (Girard, 2014; Neve, 2014):

> "Reports that say that something hasn't happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns -- the ones we don't know we don't know. And if one looks throughout the history of our country and other free countries, it is the latter category that tend to be the difficult ones." (Rumsfeld, 2002)

While many may have thought this quote to be simply political doublespeak, many who deal in computer and information security simply nodded their heads in understanding. It is always easy to break down any computing issue into a binary form: one or zero, on or off, true or false, protected or unprotected, known or unknown… However, 'in the field' experiences have shown that there is often a shade of grey between the two ends. Often there is an unknown unknown lurking that cannot be prepared for directly. Intuition, experience, hunches can all come into play in preparation and response when that third option inserts itself into the binary world. Unfortunately for those either looking to break into the world of security, or for those trying to fill security positions, those skills are not easily come by quickly or in a classroom.

David Sanger, New York Times Chief Washington Correspondent is an expert on cybersecurity. "The hardest thing about teaching anything about cybersecurity is the same thing that's the hard part about writing and reporting about cybersecurity, which is, it's moving so fast," Sanger explains (Carapezza, 2015). A Harvard graduate, he is a senior fellow and adjunct lecturer now teaching a course on cybersecurity, national security, strategy, and the press that draws from today's headlines in his lectures and case studies (Harvard, 2015). This approach is not uncommon. One of the issues that has always added complexity to Computing, Information Science, and Information Systems curriculum development has been that these programs exist in multiple different schools which by nature focus on different outcomes whether the program is housed in a Business School, Communications School, Library Science School, Engineering School, or even a Mathematics School.

The Association for Computing Machinery (ACM) has provided model curriculum guidelines since the 1960s. The 2013 model curriculum is the latest update. In it Information Assurance and Security is broken out into it's own Knowledge Area (KA) for the first time. In defining the KA, industry standards of CIA (Confidentiality, Integrity, and Availability) are used in conjunction with providing for authentication and non-repudiation. Broadening the scope, CS2013 acknowledges that both assurance and security concepts are needed to ensure a complete perspective, "Information assurance and security education, then, includes all efforts to prepare a workforce with the needed knowledge, skills, and abilities to protect our information systems and attest to the assurance of the past and current state of processes and data (ACM, 2013)."

The model curriculum guidelines for Information Systems version 2010 lists security and risk management as one of a group of five high level IS capabilities. Under the heading of Understanding, Managing and Controlling IT Risks, this is more clearly defined as, "IS graduates should have strong capabilities in understanding, managing, and controlling organizational risks that are associated with the use of IT-based solutions (e.g., security, disaster recovery, obsolescence, etc.). At the undergraduate level, the emphasis should be on in-depth understanding of a variety of risks. Because IT solutions are so closely integrated with all aspects of a modern organization, it has become essential to manage the risks related to their use in a highly systematic and comprehensive way (ACM, 2010).

Other organizations have become leaders in defining what professional certifications should encompass. The International Information Systems Security Certification Consortium, $(ISC)^2$, was formed in 1989 as a group to determine a Common Body of Knowledge (CBK) that has become the basis for what has been the leading security certification for years, the Certified Information Systems Security Professional (CISSP) certification. As of 2015 the CBK includes eight domains of focus: Security and Risk Management, Asset Security, Security Engineering, Communications and Network Security, Identity and Access Management, Security Assessment and Testing, Security Operations, and Software Development Security (ISC2, 2015) One of the hallmarks that sets the CISSP certification apart from others has been the added requirement that not only do candidates have to pass an exam related to the CBK, but they must also show that they possess a minimum of five years of direct full-time security work experience in two or more of the security domains. However, one of the critiques of the CISSP certification as the industry has matured, is that the CBK is "an inch deep and a mile wide" with this phrase even becoming the title of a popular website devoted to helping candidates prepare for the test (https://inchdeepmilewide.wordpress.com).

The alternative to the CISSP certification is offered by the EC-Council (The International Council of Electronic Commerce Consultants) with their flagship certification being the Certified Ethical Hacker (CEH). The CEH certificate has been offered since 2003 (Goldman, 2012) and is heavily centered on practical skills education and specifically penetration testing techniques. The name itself has been controversial, becoming both an asset and a possible hindrance to the organization and certificate holders (D'Ottavi, 2003; Olson, 2012). The word "hacker" has carried multiple meanings through the years and has not always been looked favorably and more conservative executives are wary of the negative association.

The other leading organization in developing curriculum and certification programs is the SANS Institute (the name is derived from SysAdmin, Audit, Networking, and Security). Founded in 1989, the organization created their Global Information Assurance Certification (GIAC) in 1999. GIAC tests and validates the ability of practitioners in information security, forensics, and software security. SANS as an organization has grown to provide training seminars on ground and online, with the SANS Technology Institute was granted regional accreditation by the Middle States Commission on Higher Education and now offers two Masters of Science degree programs (SANS, 2014). The SANS Reading Room - a research archive of information security policy and research documents delivers over one million downloads per year to professionals globally.

A common thread amongst these organizations in their curriculum models and certification paths, is that although both the CISSP and CEH require proof of field experience, these organizations have had a focus on providing support materials for the classroom and promoting standards of what should be included and expected of the students/certificate candidates.

4. COMPETITIONS

Security is at its very heart a competition. Any activity that pits one entity against another can be considered competition. Competitions take many forms and occur at many levels of intensity and consequence. Some are recreational, contested for fun. Others are blood sport, grave consequences of life and death at stake. So it is with security. Actions that were once considered fantasy only possible in movies such as War Games (1983), aggressive digital attacks have occurred causing physical damages with lasting global ramifications. The success of Stuxnet (Gross, 2011; Langer, 2011; Zetter, 2014) proved that state sponsored cyber-attacks could be as successful in the 21st century as the clandestine saboteurs of the World Wars in the 20th century was in disabling factories and other machines of war. On one hand, one country can say an act of war occurred. On another, some could say the slowing or stopping the production of a weapon of mass destruction saved countless lives.

As a training ground for education, competitions provide a fertile field for enabling students to gain experience quickly in a focused and controlled environment. This concept is not new, flight simulators have been used to train pilots for decades, and with technological innovations the use of simulators have grown to include sport and race car drivers to law enforcement and combat training. In the cybersecurity world, "capture the flag" (CTF) competitions are the simulated crucible in which the curriculum lessons are tested and validated by the students. Instead of a playing field with physical flags to capture, Red teams and Blue teams defend and attack computer networks and the flags are data and services that are either preserved or disabled.

University of California Santa Barbara (UCSB) has grown a series of live exercises in their Computer Science department into the iCTF – International Capture the Flag competition which claims to be the largest and longest running having started in 2001. Starting as a local onsite competition, it has grown to international scope and has developed an open source framework for hosting virtual networks to facilitate the hosting of other competitions (UCSB, 2015).

Perhaps the best known of the CTF competitions is the one held in conjunction with the annual DEFCON event in Las Vegas. Started in the fourth year of the conference, 1996, it has been a prominent feature of each edition since. The CTF "game" at DEFCON has evolved and grown through the last 20 years and has become a model from which many others are patterned. "At its core CTF is meant to test computer and network security. To some, that seems to be a fairly narrow focus area, but most Defcon attendees realize that "cyber security" is actually a very large and diverse field. Services range from poorly implemented or configured crypto, SQL-injection, cross-site-scripting, buffer overflows, timing attacks, heap exploits, malformed network constructs, custom interpreters, the list is truly endless. (DEFCON, 2015)."

5. INTEGRATING COMPETITIONS

'College A' is a small Catholic Liberal Arts college in the Mid-Atlantic region. Overall enrollment at the college is approximately 1,200 with a range of 60 to 70 Computing Information Science (CIS) majors. In the past four years, enrollment in the CIS program has grown nearly 30%. With limited resources, several courses are offered on every other year cycle. The Security track is one of these. During the 2012/2013 academic year a redesign of the first and second level security courses was undertaken. The first level course was organized to be in line with the (ISC)^2 Common Body of Knowledge (CBK) associated with the CISSP certification. As an introductory course, the "inch deep, mile wide" coverage of the security world is utilized to introduce students to width and breadth of the entirety of the security world. The second level course was redesigned to be more in line with the outcomes defined in the Certified Ethical Hacker (CEH) certification path, with more in depth coverage of networking, cryptography, penetration testing, and forensics.

The first level course received a good reception from the students. The general response was an eye opening experience to the wider CBK and the eight different domains. While positively received, there was a definite gap between book material and skills material that was recognized by the students. This was expressed in at least two of the comments in the course end student evaluations: "The idea of computer security is such a broad spectrum of information. I feel that I have learned a lot, but still don't know if I could handle such tasks in the work environment." And "I definitely learned a lot more about security than I thought I would. Intellectual fulfillment for me is top notch for this course, as I feel my eyes have been opened significantly. I wish there was a more interactive way to teach this, but I do also understand that to get through 10 domains, PowerPoints and lecturing are necessary. I particularly enjoyed the hands-on labs, so if more of those can be integrated without losing time to cover all of the material, then I would highly recommend that."

One of the challenges in developing the second level course was the choice of text. With a broad range of topics, the problem was encountered that no one text book was as in depth in any one of the topics as multiple texts devoted to the specific topics would be. Multiple texts would be burdensome to the students and the prevailing culture of the school would not allow a course with no formally stated required text. A decision was made to go with one text that had good coverage of two of the topics, with supplemental materials from the SANS Institute and other Internet sources on the other topics. Opinions varied on the overall success of the course. While most responses of the student surveys were positive in their comments, the survey numbers for several student evaluation categories showed a decrease between the two courses.

	1st Level F2012 (n=16)	2nd Level S2013 (n=8)
Course work contributes to objectives	5.3	5.0
Feel challenged and motivated	5.12	5.0
Successful course	5.5	5.1
Stimulates interest	5.44	5.1
Makes me feel involved	5.06	4.9
Effective Instructor	5.31	4.9

Fig 1. Student Course Evaluation Scores comparing 1st semester intro class to second semester 'advanced' class

In addressing the drop in numbers between the two courses, it was determined that the material simply needed to be brought together in a more cohesive manner that showed the interconnected nature of the disparate topics. With limited time and resources, as well as cost to student concerns, it was decided that the best course of action was to incorporate an interscholastic cyber-security competition into the course.

MACCDC

Since the mid 2000's, several groups have come together to help fill the gap between the curriculum and course material side of security education and the practical skills and experience side. In combination, the National Cyberwatch Center, The Collegiate Cyber Defense Competition (Mid-Atlantic division, MACCDC), and the National Cyber League have formed a robust environment that provides support materials, training 'gymnasiums' with practice and exploration opportunities, and a hosted high level competitive CTF style event. Participation in the NCL cost $20 per student, and registration for a team in the MACCDC competition cost $250.

During the second half of the fall semester in the first level course, students were made aware that there would be a competition team as part of the second level course in the spring. As preparation for the competition, exercises could be worked through on a volunteer basis through the NCL website.

During the spring semester, the focus of the course materials in the second level class was much the same as it had been in the previous delivery of the course two years prior. The general topics of focus were the same, with much based on the CEH path of certification. Topics related to networking, Systems Administration, and penetration testing were focused upon early as the preliminary stages for the MACCDC competition were schedule halfway through the semester.

An early concern was how to select students for the team, or whether to field two teams for the competition. For better or worse, this decision was moot, as the scheduling of the preliminary rounds fell during the time of spring break. The rules of the MACCDC call for a team no greater than eight, and eight was the number of students that would still be available to be on campus to participate.

The preliminary round was held virtually with each team/school logging into a hosted virtual environment. For the qualifying session, each team was given access to four virtual servers, two Windows based instances and two LINUX instances. Each had several services running including a SQL server, Active Directory, a web server, and a software PBX instance. From the start of the competition clock, the student team was given 15 minutes to familiarize themselves with the environment and to take any preliminary hardening actions they could. After 15 minutes a "Red Team" of aggressors comprised of event coordinators and administrators began to try and disrupt the services and functions of the servers. Scoring was comprised of three different areas: service uptime, 'flags' found (discovery questions answered about the system), and 'injects' or work orders that were given at intervals during the three-hour time period.

During the 2015 MACCDC, 30 schools participated. These schools comprised a wide variety of shape and size from large state universities, to small private liberal arts colleges, to several regional community colleges. As a first time participant, the team fielded by this school had a goal of discovery as much as competition. While not scoring in the top ten, the team did not finish last.

Observing the team during the competition showed how this crucible brought out the best, and unfortunately the worst, in the students. Three groups formed in focusing on separate tasks. One overall 'captain' was able to organize efforts, but eventually struggled to keep track of everything. Four students who had previous Sys Admin experience through internships and side projects became leaders and took charge of the individual groups. Only two sophomores threw up their arms in overwhelmed defeat.

Comments from the students highlighted their experience directly after; "I think I learned more in that three hours than I have in my previous three years…" and "Thank you for putting in the time and getting everything together and giving us the opportunity to compete. I enjoyed doing this and feel I learned some pretty cool things. At the very least it gave us a guide of what to learn and put time into."

In class after break, the team participants gave a presentation to the full class detailing the experience and highlighting some of the essentials of what they learned. Amongst the list were command and control issues such as ensuring you have multiple avenues of access into a machine and the use of the whiteboards to make sure everyone was on the same page with status reports. Other notes included the importance of your "go to" list of reference sites for basic shell/terminal commands, the importance of prioritizing your discovery to essential services, and above all else – first step, change the admin passwords!

Results

Even though only half of the students in the course were able to participate in the actual competition, all of the students benefitted by the focus of how the course materials fit the idea of the competition. The materials did not change significantly in their content or presentation from the previous delivery of the course. These positive effects were shown in the student course evaluations as compared to the same course two years earlier.

	2nd Level S2013 (N=8)	2nd Level S2015 (N=16)
Course work contributes to objectives	5.0	5.67
Feel challenged and motivated	5.0	5.60
Successful course	5.1	5.38
Stimulates interest	5.1	5.38
Makes me feel involved	4.9	5.44
Effective Instructor	4.9	5.56

Fig 2. Student Course Evaluation Scores comparing 'advanced' class scores from 2013 without competition, and 2015 with competition

Even though the number of students in the course doubled, metrics associated with student satisfaction levels, students' perceptions of the success of the course, and the instructor all increased.

6. CONCLUSIONS

Through industry news as well as contemporary reporting, the topic of computer security has become omnipresent in our daily lives. Whether the news is about corporate data breaches, international cyber espionage, or personal data compromises and identity theft – EVERYONE has had to deal with digital security in some way. Because of this, one of the fastest growing areas of need in the CIS discipline is for skilled and knowledgeable security workers. A gap has formed that has left hundreds of thousands of jobs unfilled, possibly compromising overall security even more. While nothing trumps actual experience, it is possible to instill some experience into undergraduate students through simulations and especially competition.

Several organizations, including ACM, ISC2, the EC-Council, SANS Institute, etc. have outlined model curriculums with input from leading educational and industry experts. These organizations have developed materials and provided testing infrastructure to certify students in the discipline. However, another area is needed to validate skill sets and working knowledge of the material, and that is being filled by the organizations that are developing and hosting competitions. Just as Shotokan Karate is broken into three parts – kihon (basic instruction), kata (practice of patterns of moves), and kumite (sparring) – the discipline of cyber security should encompass three parts of equal measure – basic instruction, practice, and finally a testing bed of competition.

Incorporating the third area of competition into an existing second level security course produced significant increases in student satisfaction with the course and the instructor. As one student summed it up, “This class made me feel confident that I know more about security than most people my age. I feel I have learned more in his classes than anywhere else, because you get tossed into a situation and you need to figure it out. Problem Solving skills are some of the most important skills someone in our field can have, and this course helps us develop those skills.”

9. REFERENCES

Alexander, E. (2014, September 1). Jessica Brown Findlay: Downton Abbey star is linked to list of celebrities targeted by hackers. Retrieved June 2, 2015, from http://www.independent.co.uk/news/people/jessica-brown-findlay-downton-abbey-actress-is-linked-to-list-of-celebrities-targeted-by-hackers-9704785.html

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, 2014-15 Edition, Information Security Analysts, on the Internet at http://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm (visited May 22, 2015).

Carapzza, Kirk. "With More than 200,000 Unfilled Jobs, Colleges Push Cybersecurity." PBS. January 22, 2015. Accessed April 27, 2015.

Central Challenges of American National Security, Strategy, and the Press. (2015). Retrieved June 4, 2015, from http://www.hks.harvard.edu/degrees/teaching-courses/course-listing/iga-211

Computer Science Curricula 2013: Curriculum - ACM. (2013, December 20). Retrieved February 1, 2015, from http://ACM.ORG

"Condemning Cyber-Attack by North Korea". United States Department of State. December 19, 2014. Retrieved December 24, 2014.

DEF CON Hacking Conference: CTF History. (n.d.). Retrieved June 3, 2015, from https://www.defcon.org/html/links/dc-ctf-history.html

D’Otavi, A. (2003, February 3). Interview: Marcus J. Ranum, the "father" of the firewall. Retrieved June 4, 2015, from http://www.infoservi.it/interview-marcus-j-ranum-the-father-of-the-firewall/1057

Girard, John; Girard, JoAnn (2009-06-01). A Leader's Guide to Knowledge Management: Drawing on the Past to Enhance Future Performance. Business Expert Press. pp. 55–. ISBN 9781606490198. Retrieved 10 February 2014.

Goldman, J. (2012, May 2). How to Become a Certified Ethical Hacker. Retrieved April 5, 2015, from http://www.esecurityplanet.com/hackers/how-to-become-a-certified-ethical-hacker.html

Gross, M. (2011, April 1). A Declaration of Cyber-War. Retrieved June 1, 2015, from http://www.vanityfair.com/news/2011/04/stuxnet-201104

Hamil, J. (2015, June 12). How did hackers get their paws on up to 600 celebs' sexy selfies? Retrieved June 1, 2015, from http://www.mirror.co.uk/news/technology-science/technology/fappening-sexy-celeb-selfie-hacker-5871352

Hardekopf, Bill. "UPS Reveals Data Breach in 51 Stores." LowCards. August 21, 2014. Accessed June 7, 2015. http://www.lowcards.com/ups-stores-reveals-data-breach-51-stores-26855.

Information Systems Curricula 2010: Curriculum - ACM. (2010, May 20). Retrieved February 1, 2015, from http://ACM.ORG

International Information Systems Security Certification Consortium. (n.d.). Retrieved June 14, 2015, from https://www.isc2.org

Katz, Karen. Security Info at Neiman Marcus. June 15, 2014. Accessed March 16, 2015. http://www.neimanmarcus.com/NM/Security-Info/cat49570732/c.cat?icid=topPromo_hmpg_ticker_SecurityInfo_0114.

Kopan, T. (2014, December 29). U.S.: No alternate leads in Sony hack. Retrieved April 16, 2015, from http://www.politico.com/story/2014/12/fbi-briefed-on-alternate-sony-hack-theory-113866.html

Krebs, Brian. "Krebs on Security." Krebs on Security RSS. September 14, 2014. Accessed June 3, 2015. http://krebsonsecurity.com/2014/09/home-depot-hit-by-same-malware-as-target/

Langer, R. (2011, March 1). Cracking Stuxnet, a 21st-century cyber weapon. Retrieved June 2, 2015, from http://www.ted.com/talks/ralph_langner_cracking_stuxnet_a_21st_century_cyberweapon?language=en

Lederman, Josh (January 2, 2015). "US slaps sanctions on North Korea after Sony hack". Associated Press. San Francisco Chronicle. Retrieved January 5, 2015.

Neve, Geert de; Luetchford, Peter (2008). Hidden Hands in the Market: Ethnographies of Fair Trade, Ethical Consumption, and Corporate Social Responsibility. Emerald Group Publishing. pp. 252–. ISBN 9781848550582. Retrieved 10 February 2014.

Olson, P. (2012, July 31). Exploding The Myth Of The 'Ethical Hacker' Retrieved June 3, 2015, from http://www.forbes.com/sites/parmyolson/2012/07/31/exploding-the-myth-of-the-ethical-hacker/

Resa, Dan. "The Growth of Cybersecurity Jobs." Growth of Cybersecurity Jobs. March 1, 2014. Accessed May 16, 2015. http://www.burning-glass.com/research/cybersecurity/.

Rumsfeld, D. (2002, February 12). United States Department of Defense. Retrieved February 5, 2015, from http://www.defense.gov/transcripts/transcript.aspx?transcriptid=2636

SANS: Accreditation & Authorization. (n.d.). Retrieved June 8, 2015, from https://www.sans.edu/about/authorization

The Big Data Breaches of 2014. (2014). Retrieved June 3, 2015, from http://www.forbes.com/sites/moneybuilder/2015/01/13/the-big-data-breaches-of-2014/

The UCSB iCTF Competition. (n.d.). Retrieved June 16, 2015, from http://ictf.cs.ucsb.edu/

U.S. Charges Five Chinese Military Hackers for Cyber Espionage Against U.S. Corporations and a Labor Organization for Commercial Advantage. (2014, May 19). Retrieved June 1, 2015, from http://www.justice.gov/opa/pr/us-charges-five-chinese-military-hackers-cyber-espionage-against-us-corporations-and-labor

Vijaya, Jaikumar. "Target Breach Happened Because of a Basic Network Segmentation Error." Computerworld. February 6, 2014. Accessed June 7, 2015. http://www.computerworld.com/article/2487425/cybercrime-hacking/target-breach-happened-because-of-a-basic-network-segmentation-error.html.

Virtanen, Michael. "22.8 Million Personal Records of New Yorkers Exposed." Rochester Democrat and Chronicle. July 16, 2014. Accessed June 3, 2015. http://www.democratandchronicle.com/story/news/2014/07/15/data-security-breaches-new-york/12706213/.

Zetter, K. (2014, November 3). An Unprecedented Look at Stuxnet, the World's First Digital Weapon. Retrieved June 1, 2015, from http://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/

The Relative Efficacy of Video and Text Tutorials in Online Computing Education

Guido Lang
guido.lang@quinnipiac.edu
Quinnipiac University
Hamden, CT 06518

Abstract

This study tests the effects of tutorial format (i.e. video vs. text) on student attitudes and performance in online computing education. A one-factor within-subjects experiment was conducted in an undergraduate Computer Information Systems course. Subjects were randomly assigned to complete two Excel exercises online: one with a video tutorial and one with a text tutorial. The instructions in the video tutorial and the text tutorial for the same exercise were identical – differing only in their presentation format. Following each tutorial, subjects completed a short test and a survey. Results suggest that tutorial format does not cause significant differences in student performance, time spent on tutorial, time spent on test, perceived time spent, perceived difficulty, perceived helpfulness, enjoyment, motivation, or likelihood to recommend the tutorial. Given this finding, educators and instructional designers are best advised to minimize the effort and cost involved in creating and implementing tutorials.

Keywords: tutorial format, video tutorial, text tutorial, student attitudes, student performance

1. INTRODUCTION

The widespread adoption of video tutorials and video-based learning in online learning systems and massive open online courses (MOOCs), such as Khan Academy, edX, and Coursera, has led to a plethora of new developments in the field (Giannakos, 2013). As a result, many different forms of video are used in online learning today: traditional lecture-style videos, slides with voice-over, tablet captures, screencasts, interviews, etc. Among the many praised benefits of videos for online learning is the ability to build rapport and motivate learners (Hansch et al., 2015). At the same time, the creation and implementation of videos requires significant effort and cost.

However, the efficacy of video tutorials has not been clearly established in the literature. In fact, a recent review of video in online learning concluded: "Yet, considering that video is the main method of content delivery in MOOCs, it is disconcerting how little research has been done to actually measure its pedagogical effectiveness" (Hansch et al., 2015, p. 13). The present work aims to address this shortcoming and provide insights into the relative efficacy of video tutorials. In particular, it aims to address the following research question: What are the effects of tutorial format (i.e. video or text) on student attitudes and performance?

This paper proceeds as follows. The next section provides an overview of previous work investigating the efficacy of video tutorials in the context of online computing education. Afterwards the methodology is introduced. This is followed by the results, a discussion, and finally a conclusion.

2. LITERATURE REVIEW

The topic of video tutorials and video-based learning is gaining significant attention in the research community, as indicated by two recent meta reviews (Yousef, Chatti, & Schroeder, 2014; Giannakos, 2013). However, only a handful of studies have conducted experimental investigations into the efficacy of video tutorials in the context of online computing education. In contrast to other academic subjects, computing education tends to combine conceptual understanding with technical skills. Thus, tutorials are of particular importance in computing education. Among the few efficacy studies in this field are Lee, Pradhan, and Dalgarno (2008), Breimer, Cotler, and Yoder (2012), Lloyd and Robertson (2012), and Tekinarslan (2013).

Lee, Pradhan, and Dalgarno (2008) evaluated the impact of video tutorials in an introductory programming course. Video tutorials were used as part of a scaffolding exercise to introduce students to object-oriented programming using BlueJ, a Java development environment. Subjects were randomly assigned to have access to video tutorials about BlueJ or not to have access to any tutorials. The video tutorials showed how BlueJ was used but did not have any narration. Subsequently, subjects completed a paper-based test, requiring them to write Java code to perform a number of tasks. Findings suggest that video tutorials did not have an impact on student performance. However, because the tutorials were not directly related to the test material and did not include any explanations, it is possible that the negative result was due to limitations in the design of the study and the tutorials.

Breimer, Cotler, and Yoder (2012) examined differences between video and text tutorials with respect to concept learning, task completion time, retention, and student impression as part of a database exercise using Microsoft Access. Subjects were randomly assigned to either receive video or text tutorials and subsequently completed a test and survey. The video tutorials lasted about 35 minutes and the corresponding text tutorial consisted of 1600 words and 20 screenshots. Interestingly, the authors did not find any statistically significant differences between the two groups with regards to the dependent variables. However, the authors found that differences in tutorial format caused differences in student learning behavior. Specifically, the average completion time of subjects in the video condition was nearly twice the duration of subjects in the text condition. Thus, it is possible that the negative findings were due to information or usage differences between video and text tutorials.

Lloyd and Robertson (2012) assessed the effect of video tutorials vs. text tutorials on learning outcomes in the context of teaching statistics using SPSS. Subjects were randomly assigned to receive a video tutorial (lasting about 12 minutes) or a text tutorial demonstrating how to conduct an independent samples *t*-test analysis in SPSS. Subsequently subjects were tasked to solve a statistics problem by applying the knowledge gained in the tutorial. Findings show that subjects in the video tutorial condition performed significantly better than subjects in the text tutorial condition.

Tekinarslan (2013) conducted an experiment investigating the effect of video tutorials on student learning when teaching Microsoft Excel. Subjects were taught Excel during computer lab sessions and were randomly assigned to either have access to screen recordings of the lab sessions or not to have access to the recordings. At the end of the semester, subjects' knowledge of Excel was assessed using a test. Results suggest that subjects with access to the video tutorials performed significantly better than subjects without access to tutorials. Thus, the author concludes that providing video tutorials is better for student performance than providing no tutorials at all.

In summary, previous experimental investigations into the efficacy of video tutorials in the context of online computing education found mixed results. However, previous work focused solely on student performance – without considering student attitudes. This leaves entirely open the question if and to what extent students' attitudes are differently affected by video and text tutorials. It is possible that while having no effect on student performance, video tutorials positively or negatively affect student attitudes.

3. METHODOLOGY

The purpose of this study is to test if and how tutorial format (i.e. video vs. text) affects student attitudes and performance. An experiment was conducted as part of an undergraduate introductory Computer Information Systems course at mid-sized, private university in the northeastern United States. Students participated in the experiment in exchange for extra credit (worth approximately 5% of the final grade).

The experiment, which took place entirely online, asked subjects to complete two Excel exercises: one exercise on using LEFT and SEARCH functions to extract and copy text strings into separate cells (exercise 1) and one exercise on using INDEX and MATCH functions to lookup values in a table (exercise 2). Each exercise consisted of a short tutorial – in video or text format – followed by a test and a survey. The video and text tutorials for each exercise were carefully designed to be equivalent in their information content. In other words, the instructions contained in the video tutorial for an exercise were identical to the instructions contained in the text tutorial for the same exercise – differing only in how they were presented (i.e. video or text). The video tutorial for exercise 1 was 2:50 minutes long. The corresponding text tutorial consisted of approximately 350 words and 10 screenshots (taken from the video). Likewise, the video tutorial for exercise 2 was 3:53 minutes long and the corresponding text tutorial consisted of approximately 400 words and 10 screenshots (taken from the video). Following the tutorial, subjects completed a short test, which required the application of knowledge from the tutorial to solve a problem using Excel. The test consisted of one open-ended question and three multiple-choice questions. To facilitate subjects' use of Excel during the test, the experimental website included a browser-based version of Excel. After the test, subjects completed a short survey measuring their attitudes towards the preceding tutorial. Afterwards subjects completed the second exercise – following the same process of tutorial, test, and survey. Further details about the experimental setup, including the tutorials, test questions, and survey items can be found in the Appendix.

The experiment was a one-factor (tutorial format: video vs. text) within-subjects design. Subjects were randomly assigned to receive one exercise with a video tutorial and one exercise with a text tutorial. Thus, each subject was exposed to both levels of the independent variable (i.e. tutorial format). To counter potential carryover and learning effects, the order of the exercises and the order of the tutorial format (i.e. video or text) were randomized for all subjects.

Using a combination of server log data and subjects' responses, the following dependent variables were measured:

- *Student performance*: Number of correct answers on the test (out of 4 total)
- *Time spent on tutorial*: Amount of time spent on the tutorial (based on server log data)
- *Time spent on test*: Amount of time spent on the test (based on server log data)
- *Perceived time spent on exercise*: Subjective amount of time spent on the exercise (survey item)
- *Perceived difficulty*: Subjective level of perceived difficulty of the exercise (survey item)
- *Perceived helpfulness*: Subjective level of perceived helpfulness of the tutorial (survey item)
- *Enjoyment*: Subjective level of enjoyment in completing the exercise (survey item)
- *Motivation*: Subjective level of motivation to complete the exercise (survey item)
- *Likelihood to recommend tutorial*: Subjective level of likelihood to recommend the tutorial to a friend (survey item)

Moreover, gender, undergraduate major, undergraduate level, general prior knowledge (of Excel), and specific prior knowledge (of the content covered in each exercise) were measured as control factors in this study.

4. RESULTS

A total of N = 75 subjects completed the study. Detailed demographics of the sample are presented in Table 1.

Table 1. Sample Demographics

Gender	
Male	50 (67%)
Female	25 (33%)
Undergraduate Level	
Freshman	9 (12%)
Sophomore	24 (32%)
Junior	28 (37%)
Senior	14 (19%)
Undergraduate Major	
Non-CIS	38 (51%)
CIS	37 (49%)

To test if the randomization of subjects to both order of exercises and order of the tutorial format was unbiased, the control factors were entered in a one-way between-subjects multiple analysis of covariance (MANCOVA). No statistically significant effects were found (all *Wilks' Lambda* > .96, $F(2,69) < 1.29$, $p > .28$), suggesting that the randomization was indeed successful.

To identify potential differences between the two exercises, a one-way within-subjects MANOVA testing the effects of exercise (i.e. exercise 1 vs. exercise 2) on the dependent variables was conducted. A statistically significant effect was found (*Wilks' Lambda* = .60, $F(9,64) = 4.69$, $p < .001$). Multiple paired samples t-tests were used to make post-hoc comparisons. Two statistically significant differences emerged: While subjects spent about one minute less on the test in exercise 1 than on the test in exercise 2 ($M_{\text{Ex1}} = 209.72$, $SD_{\text{Ex1}} = 166.57$, $M_{\text{Ex2}} = 273.53$, $SD_{\text{Ex2}} = 218.11$, $t = -2.30$, $p = .02$, all measures in seconds), they also performed significantly better on the test in exercise 1 than on the test in exercise 2 ($M_{\text{Ex1}} = 2.68$, $SD_{\text{Ex1}} = 1.16$, $M_{\text{Ex2}} = 2.11$, $SD_{\text{Ex2}} = 1.13$, $t = 3.96$, $p < .001$). No other differences reached statistical significance (all *t*s < 1.22, *p*s > .23). This suggests that the test accompanying exercise 1 might have been easier than the test accompanying exercise 2. Apart from this difference, the two exercises had otherwise statistically indistinguishable outcomes with regards to the dependent variables.

Finally, a one-way within-subjects MANOVA was conducted to test the effects of tutorial format (i.e. video vs. text) on the dependent variables. No statistically significant effect was found (*Wilks' Lambda* = .85, $F(9,64) = 1.29$, $p = .26$).

5. DISCUSSION

The results suggest that tutorial format does not cause any differences in the dependent variables. In other words, whether subjects received a video tutorial or a text tutorial did not affect their performance, time spent on tutorial, time spent on test, perceived time spent on exercise, perceived difficulty, perceived helpfulness, enjoyment, motivation, or likelihood to recommend the tutorial.

These results contradict previous research, which found that video tutorials lead to better student performance than text tutorials (Lloyd and Robertson, 2012). Comparing this work to the previous research, it appears that potential differences in the length of the video tutorials may influence the effect on student performance. Whereas the average video length in this study was 3:16 minutes, the average video length in the previous research was 12 minutes (Lloyd and Robertson, 2012). However, this study also lends support to previous work, which found that video tutorials and text tutorials do not differ with regards to their effect on student performance (Breimer, Cotler, and Yoder, 2012). The average video duration in the study by Breimer and colleagues was 35 minutes, which is significantly longer than the average video duration in the present study.

This research is among the first to show that student attitudes appear to be insensitive to differences in tutorial format. However, it is important to point out that subjects were not asked directly which tutorial format they prefer – as this might be influenced by previous experiences or general preferences. Instead, the present work compared specific student attitudes after each exercise was completed. Clearly, future research is needed to further investigate if and to what extent this finding can be replicated in other contexts.

Future research is also needed to address some of the limitations of this study. In particular, subjects participated in the experiment for extra credit. As such, the experiment was not part of the normal classroom routine. Consequently future work should integrate the experimental setup into the regular classroom. Moreover, the number, duration, and subject matter of the tutorials may influence the relative effectiveness of video or text tutorials. Future work may wish to expand and test for differences across a wider range and number of tutorials. Lastly, it is possible that the sample used in this study is not representative of the larger body of undergraduate students in the United States. Future research should capture additional psychological measures, such as learning style, to describe and control potential sample-specific differences.

These limitations notwithstanding, the present research suggests that educators should carefully consider the additional effort and cost involved in creating and implementing video tutorials. The current trend among online learning systems and MOOCs to use videos extensively may be driven by considerations unrelated to student attitudes or performance. As such, educators and instructional designers should not base their decision to create or implement video tutorials on the false belief that video tutorials are inherently better than text tutorials.

6. CONCLUSION

Does tutorial format (i.e. video or text) affect student attitudes and performance in online computing education? The present study answers this question using a single-factor within-subjects experiment varying tutorial format while keeping the information across tutorial formats constant. As part of an undergraduate Computer Information Systems course subjects were randomly assigned to complete two Excel exercises: one with a video tutorial and one with a text tutorial. The instructions contained in the video tutorial for an exercise were identical to the instructions contained in the text tutorial for the same exercise – differing only in how they were presented. Following each tutorial, subjects completed a short test and a survey. Findings suggest that tutorial format does not cause significant differences in student performance, time spent on tutorial, time spent on test, perceived time spent, perceived difficulty, perceived helpfulness, enjoyment, motivation, or likelihood to recommend the tutorial. In short, tutorial format appears to be unrelated to student attitudes and performance in online computing education. Consequently, educators and instructional designers are well advised to look beyond the current trend to use video tutorials and instead make choices that minimize the effort and cost involved in creating and implementing tutorials.

7. REFERENCES

Breimer, E., Cotler, J., & Yoder, R. (2012). Video vs. Text for Lab Instruction and Concept Learning. *Journal of Computing Sciences in Colleges*, 27(6), 42-48.

.

Giannakos, M. N. (2014). Exploring the Video-Based Learning Research: A Review of the Literature. *British Journal of Educational Technology*, 44(6), E191-E195.

Hansch, A., Hillers, L., McConachie, K., Newman, C., Schildhauer, T., & Schmidt, P. (2015). Video and Online Learning: Critical Reflections and Findings from the Field. HIIG Discussion Paper Series No. 2015-02.

Lee, M., Pradhan, S., & Dalgarno, B. (2008). The Effectiveness of Screencasts and Cognitive Tools as Scaffolding for Novice Object-Oriented Programmers. *Journal of Information Technology Education*, 7, 61-80.

Lloyd, S., & Robertson, C. (2012). Screencast Tutorials Enhance Student Learning of Statistics. *Teaching of Psychology*, 39(1), 67-71.

Tekinarslan, E. (2013). Effects of Screencasting on the Turkish Undergraduate Students' Achievement and Knowledge Acquisitions in Spreadsheet Applications. *Journal of Information Technology Education: Research*, 12, 271-282.

Yousef, A., Chatti, M., & Schroeder, U. (2014). Video-Based Learning: A Critical Analysis of The Research Published in 2003-2013 and Future Visions. Proceedings of the Sixth International Conference on Mobile, Hybrid, and On-line Learning (eLmL 2014), 112-119

APPENDIX: MATERIALS

EXERCISE 1

VIDEO TUTORIAL: https://www.youtube.com/watch?v=eV5P8rfXZck (2:50 min)

TEXT TUTORIAL

In this tutorial you'll learn how to use the LEFT and SEARCH functions to extract and copy text strings into separate cells. Please pay close attention to the following instructions.

Introduction

B2 | fx |

	A	B	C
1	***E-Mail***	***Username***	
2	jdod@domain.com		
3	bsmith@domain.com		
4	rmkentley@domain.com		
5	pconnor@domain.com		
6			

This example demonstrates how to extract the username from an e-mail address. An e-mail address consists of two parts: the username (e.g. jdod) and the domain (e.g. domain.com). The two parts are separated by an @ sign. We're going to use two functions to extract the username: The LEFT function and the SEARCH function.

Step 1

LEFT | fx | =LEFT(A2,4)

	A	B	C
1	***E-Mail***	***Username***	
2	jdod@domain.com	=LEFT(A2,4)	
3	bsmith@domain.com		
4	rmkentley@domain.com		
5	pconnor@domain.com		
6			

The LEFT function returns the first characters in a text string. So, for example, LEFT(A2, 4) returns the first 4 characters of the text string in cell A2, which is "jdod".

B2 | fx | =LEFT(A2,4)

	A	B	C
1	***E-Mail***	***Username***	
2	jdod@domain.com	jdod	
3	bsmith@domain.com	bsmi	
4	rmkentley@domain.com	rmke	
5	pconnor@domain.com	pcon	
6			
7			

When we copy that formula in the remaining cells, we always get the first 4 characters of the cells in column A. However, the problem is that the usernames are not always 4 characters long.

Step 2

SEARCH | fx | =SEARCH("@",A2)

	A	B	C	D
1	***E-Mail***	***Username***		
2	jdod@domain.com	=SEARCH("@",A2)		
3	bsmith@domain.com			
4	rmkentley@domain.com			
5	pconnor@domain.com			
6				

We have to change the formula to automatically account for the number of characters there are before the @ sign. To do that, we use the SEARCH function. The SEARCH function returns the position of a specific character in a text string. For example, SEARCH("@", A2) returns the position of the @ sign in cell A2.

B2 | fx | =SEARCH("@",A2)

	A	B	C	D
1	***E-Mail***	***Username***		
2	jdod@domain.com	5		
3	bsmith@domain.com			
4	rmkentley@domain.com			
5	pconnor@domain.com			
6				

Notice the formula returns the number 5, as the @ sign is at the fifth position in cell A2. Now we can use the SEARCH function within the LEFT function to determine how many characters to return.

Step 3

LEFT | fx | =LEFT(A2,SEARCH("@",A2))

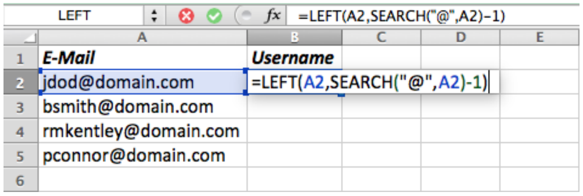
	A	B	C	D
1	***E-Mail***	***Username***		
2	jdod@domain.com	=LEFT(A2,SEARCH("@",A2))		
3	bsmith@domain.com			
4	rmkentley@domain.com			
5	pconnor@domain.com			
6				

So then LEFT(A2, SEARCH("@", A2)) returns the first number of characters in cell A2, based on the position of the @ sign in cell A2.

B2 | fx | =LEFT(A2,SEARCH("@",A2))

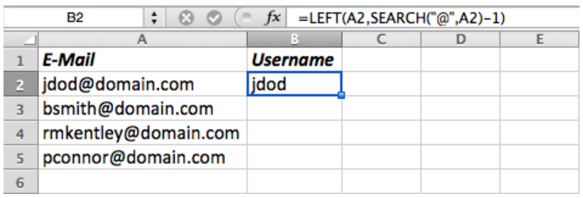
	A	B	C	D
1	***E-Mail***	***Username***		
2	jdod@domain.com	jdod@		
3	bsmith@domain.com			
4	rmkentley@domain.com			
5	pconnor@domain.com			
6				

Step 4


LEFT =LEFT(A2,SEARCH("@",A2)-1)

	A	B	C	D	E
1	*E-Mail*	*Username*			
2	jdod@domain.com	=LEFT(A2,SEARCH("@",A2)-1)			
3	bsmith@domain.com				
4	rmkentley@domain.com				
5	pconnor@domain.com				
6					

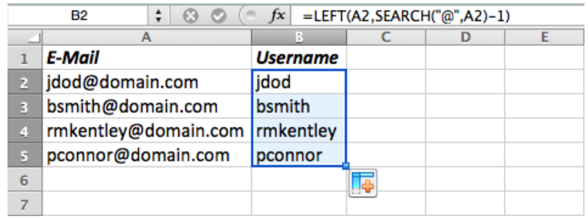
We must reduce the result of the SEARCH function by 1 character. To do that, we edit the function and add "-1" directly after the SEARCH function, but still within the LEFT function.


B2 =LEFT(A2,SEARCH("@",A2)-1)

	A	B	C	D	E
1	*E-Mail*	*Username*			
2	jdod@domain.com	jdod			
3	bsmith@domain.com				
4	rmkentley@domain.com				
5	pconnor@domain.com				
6					

Notice the formula now returns "jdod," which is exactly what we want.

Finish


B2 =LEFT(A2,SEARCH("@",A2)-1)

	A	B	C	D	E
1	*E-Mail*	*Username*			
2	jdod@domain.com	jdod			
3	bsmith@domain.com	bsmith			
4	rmkentley@domain.com	rmkentley			
5	pconnor@domain.com	pconnor			
6					
7					

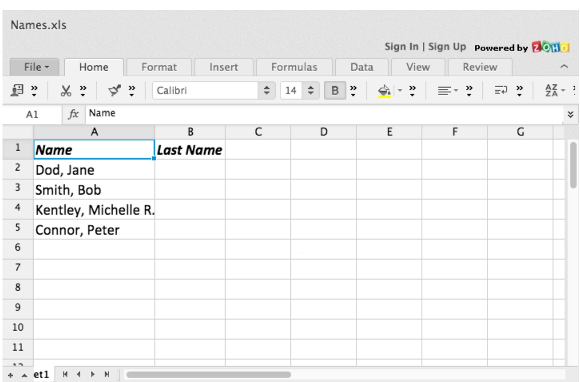
We copy the formula to the remaining cells in column B. Notice the final result is a clean list of usernames.

<u>TEST (EXERCISE 1)</u>

This exercise tests your understanding of the LEFT and SEARCH functions covered in this tutorial.

Question 1

Below you are given a list of names. Your task is to write a formula in cell B2 that automatically extracts the last name from the name in cell A2. Use the interactive spreadsheet below to practice. (The interactive spreadsheet shown below is embedded via an iFrame linking to an online Excel file via http://sheet.zoho.com/view.do?url=...)

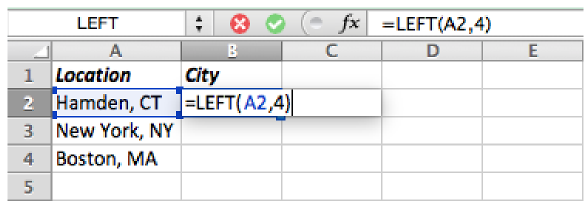


	A	B	C	D	E	F	G
1	*Name*	*Last Name*					
2	Dod, Jane						
3	Smith, Bob						
4	Kentley, Michelle R.						
5	Connor, Peter						
6							
7							
8							
9							
10							
11							

What formula did you type into cell B2 above? (Open-ended question; correct answer is "=LEFT(A2;SEARCH(",";A2)-1)", without quotations)

Question 2

What will be the result of the formula in cell B2 below?

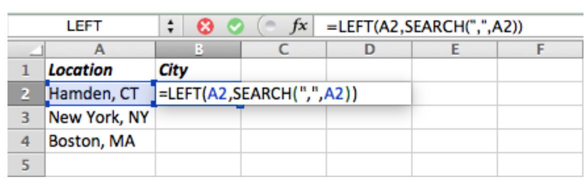

LEFT =LEFT(A2,4)

	A	B	C	D	E
1	*Location*	*City*			
2	Hamden, CT	=LEFT(A2,4)			
3	New York, NY				
4	Boston, MA				
5					

a) Hamd (correct answer)
b) Hamden
c) Hamden,
d) Hamden, CT
e) None of the above

Question 3

What will be the result of the formula in cell B2 below?

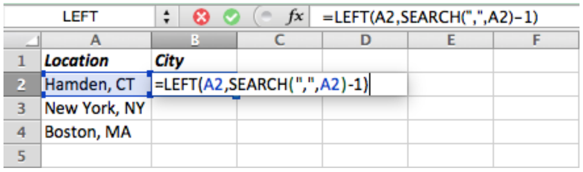

LEFT =LEFT(A2,SEARCH(",",A2))

	A	B	C	D	E	F
1	*Location*	*City*				
2	Hamden, CT	=LEFT(A2,SEARCH(",",A2))				
3	New York, NY					
4	Boston, MA					
5						

a) Hamd
b) Hamden
c) Hamden, (correct answer)
d) Hamden, CT
e) None of the above

Question 4

What will be the result of the formula in cell B2 below?



LEFT =LEFT(A2,SEARCH(",",A2)-1)

	A	B	C	D	E	F
1	*Location*	*City*				
2	Hamden, CT	=LEFT(A2,SEARCH(",",A2)-1)				
3	New York, NY					
4	Boston, MA					
5						

a) Hamd
b) Hamden (correct answer)
c) Hamden,
d) Hamden, CT
e) None of the above

SURVEY (EXERCISE 1)

What was your previous knowledge of the content covered in Exercise 1?
(None – Poor – Fair – Good – Excellent)

How difficult was the content covered in Exercise 1 for you?
(Very difficult – Difficult – Neutral – Easy – Very easy)

How motivated were you to complete Exercise 1?
(Not at all motivated – Slightly motivated – Somewhat motivated – Moderately motivated – Very motivated)

How enjoyable was Exercise 1?
(Not at all enjoyable – Slightly enjoyable – Somewhat enjoyable – Moderately enjoyable – Very enjoyable)

How helpful were the instructions in Exercise 1?
(Not at all helpful – Slightly helpful – Somewhat helpful – Moderately helpful – Very helpful)

How likely is it that you would recommend a tutorial like the one in Exercise 1 to a friend?
(Not at all likely – Slightly likely – Somewhat likely – Moderately likely – Very likely)

How long did it take you to complete Exercise 1? (Provide an estimate in minutes.)
(Open-ended question)

EXERCISE 2

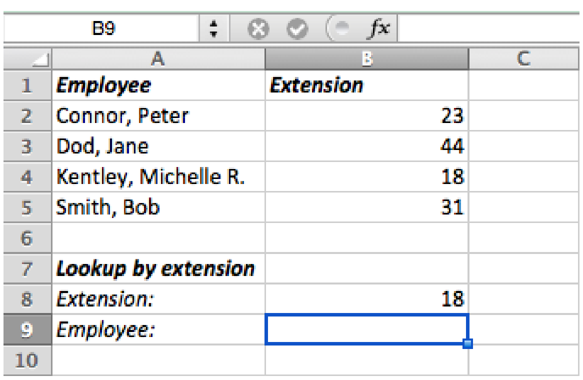
VIDEO TUTORIAL: https://www.youtube.com/watch?v=NpQTu30BwT0 (3:52 min)

TEXT TUTORIAL

Instructions

In this tutorial you'll learn how to use the INDEX and MATCH functions to lookup values in a table. Please pay close attention to the following instructions.

Introduction

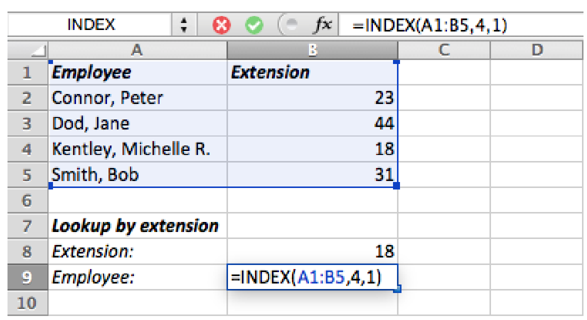


B9

	A	B	C
1	***Employee***	***Extension***	
2	Connor, Peter	23	
3	Dod, Jane	44	
4	Kentley, Michelle R.	18	
5	Smith, Bob	31	
6			
7	***Lookup by extension***		
8	*Extension:*	18	
9	*Employee:*		
10			

This example demonstrates how to lookup an employee by their extension number. We're going to use two functions to do that: The INDEX function and the MATCH function.

Step 1



INDEX =INDEX(A1:B5,4,1)

	A	B	C	D
1	***Employee***	***Extension***		
2	Connor, Peter	23		
3	Dod, Jane	44		
4	Kentley, Michelle R.	18		
5	Smith, Bob	31		
6				
7	***Lookup by extension***			
8	*Extension:*	18		
9	*Employee:*	=INDEX(A1:B5,4,1)		
10				

The INDEX function returns a value from a table, given a row and column number. So, for example, INDEX(A1:B5, 4, 1) returns the value from the table A1:B5 where the row number is 4 and the column number is 1.

B9 | fx =INDEX(A1:B5,4,1)

	A	B	C	D
1	***Employee***	***Extension***		
2	Connor, Peter	23		
3	Dod, Jane	44		
4	Kentley, Michelle R.	18		
5	Smith, Bob	31		
6				
7	***Lookup by extension***			
8	*Extension:*	18		
9	*Employee:*	Kentley, Michelle R.		
10				

Notice the formula returns "Kentley, Michelle R.," which is located in row 4, column 1, in table A1:B5. However, for our example, we would like to automate the formula, so that we don't have to specify which row number we are looking for.

Step 2

MATCH | fx =MATCH(18,B1:B5,0)

	A	B	C	D
1	***Employee***	***Extension***		
2	Connor, Peter	23		
3	Dod, Jane	44		
4	Kentley, Michelle R.	18		
5	Smith, Bob	31		
6				
7	***Lookup by extension***			
8	*Extension:*	18		
9	*Employee:*	=MATCH(18,B1:B5,0)		
10				

To do that, we use the MATCH function. The MATCH function returns the row number of a search item in a column. So, for example, MATCH(18, B1:B5, 0) returns the row number that contains the value 18 in column B1:B5. The last argument in the formula, 0, just tells the function to find the first row that is exactly equal to the search item.

B9 | fx =MATCH(18,B1:B5,0)

	A	B	C	D
1	***Employee***	***Extension***		
2	Connor, Peter	23		
3	Dod, Jane	44		
4	Kentley, Michelle R.	18		
5	Smith, Bob	31		
6				
7	***Lookup by extension***			
8	*Extension:*	18		
9	*Employee:*	4		
10				

Notice the formula returns 4, because the value 18 is located in the 4th row of column B1:B5. Now we can use the MATCH function within the INDEX function to automatically find the extension number we are looking for.

Step 3

INDEX | fx =INDEX(A1:B5,MATCH(18,B1:B5,0),1)

	A	B	C	D	E
1	***Employee***	***Extension***			
2	Connor, Peter	23			
3	Dod, Jane	44			
4	Kentley, Michelle R.	18			
5	Smith, Bob	31			
6					
7	***Lookup by extension***				
8	*Extension:*	18			
9	*Employee:*	=INDEX(A1:B5,MATCH(18,B1:B5,0),1)			
10					

So then INDEX(A1:B5, MATCH(18, B1:B5, 0), 1) returns the value from table A1:B5, where the row contains the value 18, and the column number is 1.

B9 | fx =INDEX(A1:B5,MATCH(18,B1:B5,0),1)

	A	B	C	D	E
1	***Employee***	***Extension***			
2	Connor, Peter	23			
3	Dod, Jane	44			
4	Kentley, Michelle R.	18			
5	Smith, Bob	31			
6					
7	***Lookup by extension***				
8	*Extension:*	18			
9	*Employee:*	Kentley, Michelle R.			
10					

Notice the formula returns again "Kentley, Michelle R.," which is the name of the employee whose extension is 18. However, we don't want to hard code the extension 18 into our formula, because we might want to look up other extensions in the future.

Step 4

INDEX | fx =INDEX(A1:B5,MATCH(B8,B1:B5,0),1)

	A	B	C	D	E
1	***Employee***	***Extension***			
2	Connor, Peter	23			
3	Dod, Jane	44			
4	Kentley, Michelle R.	18			
5	Smith, Bob	31			
6					
7	***Lookup by extension***				
8	*Extension:*	18			
9	*Employee:*	=INDEX(A1:B5,MATCH(B8,B1:B5,0),1)			
10					

To do that, we replace the value 18 in the formula with a reference to cell B8. Given that cell B8 currently contains the value 18, the result should stay the same.

B9 | fx =INDEX(A1:B5,MATCH(B8,B1:B5,0),1)

	A	B	C	D	E
1	***Employee***	***Extension***			
2	Connor, Peter	23			
3	Dod, Jane	44			
4	Kentley, Michelle R.	18			
5	Smith, Bob	31			
6					
7	***Lookup by extension***				
8	*Extension:*	18			
9	*Employee:*	Kentley, Michelle R.			
10					

Notice the formula returns "Kentley, Michelle R.," which is exactly what we want.

Finish

B9 | =INDEX(A1:B5,MATCH(B8,B1:B5,0),1)

	A	B	C	D	E
1	***Employee***	***Extension***			
2	Connor, Peter	23			
3	Dod, Jane	44			
4	Kentley, Michelle R.	18			
5	Smith, Bob	31			
6					
7	***Lookup by extension***				
8	*Extension:*	44			
9	*Employee:*	Dod, Jane			
10					

To test if our formula behaves as expected, we change the extension number in cell B8 to e.g. 44. Notice the formula returns "Dod, Jane," which is the name of the employee whose extension is 44.

TEST (EXERCISE 2)

This exercise tests your understanding of the INDEX and MATCH functions covered in this tutorial.

Question 1

Below you are given a table of product names and product IDs. Your task is to write a formula in cell B9 that automatically returns the product name for any product ID that is specified in cell B8. Use the interactive spreadsheet shown below to practice. (The interactive spreadsheet shown below is embedded via an iFrame linking to an online Excel file via http://sheet.zoho.com/view.do?url=...)

Products.xls

A1 | Product Name

	A	B	C	D	E	F	G	H
1	***Product Name***	***Product ID***						
2	Sandwich	35						
3	Muffin	83						
4	Cookie	49						
5	Coffee	65						
6								
7	***Lookup by product ID***							
8	*Product ID:*	83						
9	*Product Name:*							
10								
11								

What formula did you type into cell B9 above? (Open-ended question; correct answer is "=INDEX(A1:B5;MATCH(B8;B1:B5;0);1)", without quotations)

Question 2

What is the result of the formula in cell B9 below?

INDEX | =INDEX(A1:B5,3,1)

	A	B	C	D	E
1	***Capital***	***State***			
2	Montgomery	Alabama			
3	Juneau	Alaska			
4	Phoenix	Arizona			
5	Little Rock	Arkansas			
6					
7	***Lookup by state***				
8	*State:*	Arkansas			
9	*Capital:*	=INDEX(A1:B5,3,1)			
10					

a) Alaska
b) Juneau (correct answer)
c) Arizona
d) Little Rock
e) None of the above

Question 3

What is the result of the formula in cell B9 below?

MATCH | =MATCH("Alaska",B1:B5,0)

	A	B	C	D	E	F
1	***Capital***	***State***				
2	Montgomery	Alabama				
3	Juneau	Alaska				
4	Phoenix	Arizona				
5	Little Rock	Arkansas				
6						
7	***Lookup by state***					
8	*State:*	Arkansas				
9	*Capital:*	=MATCH("Alaska",B1:B5,0)				
10						

a) 2
b) 3 (correct answer)
c) Juneau
d) Alaska
e) None of the above

Question 4

What is the result of the formula in cell B9 below?

INDEX | =INDEX(A1:B5,MATCH(B8,B1:B5,0),1)

	A	B	C	D	E	F	G
1	***Capital***	***State***					
2	Montgomery	Alabama					
3	Juneau	Alaska					
4	Phoenix	Arizona					
5	Little Rock	Arkansas					
6							
7	***Lookup by state***						
8	*State:*	Arkansas					
9	*Capital:*	=INDEX(A1:B5,MATCH(B8,B1:B5,0),1)					
10							

a) Arkansas
b) Little Rock (correct answer)
c) Phoenix
d) Alabama
e) None of the above

SURVEY QUESTIONS (EXERCISE 2)

Analog to survey questions for exercise 1 (see above).

DEMOGRAPHIC QUESTIONS

What is your general knowledge of spreadsheet applications (such as Excel)?
(None – Poor – Fair – Good – Excellent)

What is your gender?
(Male – Female)

What is your major?
(Open-ended question)

What is your undergraduate level?
(Freshman – Sophomore – Junior – Senior)

Use of Failure in IS Development Statistics: Lessons for IS Curriculum Design

Herbert H. Longenecker, Jr.
longeneckerb@gmail.com
University of South Alabama
Mobile, AL 36688

Jeffry Babb
jbabb@wtamu.edu
West Texas A&M University
Canyon, TX 79116

Leslie Waguespack
lwaguespack@bentley.edu
Bentley University
Waltham, MA 02452

William Tastle
tastle@ithaca.edu
Ithaca College
Ithaca, NY 14850

Jeff Landry
University of South Alabama
jlandry@usouthal.edu
University of South Alabama
Mobile, AL 36688

Abstract

The evolution of computing education reflects the history of the professional practice of computing. Keeping computing education current has been a major challenge due to the explosive advances in technologies. Academic programs in Information Systems, a long-standing computing discipline, develop and refine the theory and practice of computing appropriate to professional practice. As the computing professions advance, so too does our conceptualization and design of curricula in information systems. Subsequently, our organizing bodies (i.e. DPMA, AITP, AIS, ACM, etc.) coordinate and cooperate in the development of curricular guidelines and models. These models serve to establish shared core and methodology among practitioners and educators. This paper presents the case that there are lessons in the history of the computing fields – particularly among the system development failures – that can inform the design of curricula aimed at preparing computing students for professional practice. Given the repetitive nature of many of these failures, we posit that failures can shed light into the "dark places," and, with care, illuminate the essential nature of the information systems discipline and the body of knowledge and skill sets essential to our educational task.

Keywords: Computer Information Systems (CIS), Building CIS Programs, CIS Curricula and Specialties, Body of Knowledge

1. INTRODUCTION

In the 1960's, as the computing sciences emerged, it became obvious that educated professionals were needed. This led to the development of university programs of study with a naturel division into the (1) development of both the hardware and software of the machines and (2) business and organizational applications. Over time, Academic curricula developed for each area. The hardware characteristic at the time including small memories, limited peripheral storage, and severely restricted input and output was a necessary curricular consideration of all academic programs. From these earliest beginnings, a consistent sentiment is well expressed by Gordon Davis**,** "Information systems is about the development and deployment of systems." (Davis, 1994) While the words development and deployment have evolved, the concept remains similar. As educators in the field of Information Systems, we are reminded that "computer" is not a dirty word to be avoided; it is the wellspring that binds the computing disciplines and allows for Information Systems to ground itself. To wit, some educators and programs have taken to claiming this heritage by prefixing "computer" to the program. We investigate the origins of this sentiment and explore whether Gordon Davis' sentiment remains relevant.

This paper outlines an argument that as computing education has evolved, propelled by computing's successes and its transformative power, there have been numerous and significant failures. While the litany of these failures is entrenched in our canon, we should recognize these failures – of specification, design, development, and implementation – as evidence of blind spots: core deficiencies in the professional practice of computing that require more curricular attention – particularly in the academic discipline known generally as "information systems."

Among the marvels of human history is how computing power has grown considerably and consistently since the inception of the various computing disciplines. In the nascency of computing disciplines, computers were programmed using primarily the FORTRAN and COBOL languages. As tools like these were widely used, curricula specifications in computing, such as ACM '68 and ACM '71, accounted for curricular formulations that reflected the needs of industry. At the time, in the business domain, the computing function was mainly accounting and reporting functions (to the CFO and upper level management). As academic programs emerged to support the computing needs of government and businesses, a graduate's professional track was typically to first be a programmer, and then a progression to analysts and beyond (eventually these functions required management). This progression was consistent such that there was little diversity of expectations; most everyone went through these stages, even up to the mid 1990's.

The Rise of Computing in Business

From its beginnings in the 1960s, computing increasingly factored into daily life, and subsequently into culture. For instance, the January 1983 issue of TIME magazine had the IBM PC as the "man of the year." Given the impact of shrinking computing architectures such that "everyday" people could compute in their own homes, public awareness of computing, and the impacts it was having on daily life, had grown tremendously. Subsequently, enrollments in university programs in computing-related disciplines became very large. With wider exposure, the public quickly realized that the PC was far from trivial to learn and use; to the contrary, many realized that attempting to get these machines to do anything often resulted in adversity and outright failure.

It seems that computing finds new traction in the public imagination every 5 years or so (the PC, Software, The Internet, The World Wide Web, Web 2.0, Mobile, etc.). In these cases, a gold rush mentality develops, and enrollments in computing programs spike, and then the realities of actually working with these new technologies set in, and, enrollments return to their previous levels. While computing, and the academic programs that have arisen to develop professionals in computing, have matured and diversified, the core knowledge and fundamental skill sets remain largely unchanged. During the earliest expansions and contractions of student interest in academic programs leading to a career in computing, computing education retained its essential focus: the application of problem solving and logic through the medium of the computer to produce useful business tools, computing artifacts. The surge of student interest mirrored the rapid realization in most organizations of the power and potential of computing to exploit information for business success.

In the 1970s and 80s, the computer center director and his million-dollar machine owned the landscape as long as the payroll was produced on time. It was rarely possible to win in an outright slugfest against this power. Upon the entry and ascendancy of the PC, a counter to the power of central computing was the fact that the many varieties of the PC were both affordable and powerful. What ushered in has become known as the end-user computing era – functional units had computing power right in their areas, and would call upon computing professionals to develop applications and systems around these machines. As they became inter-networked, this reallocation of power became even more pervasive. Even in cases where the director of computing controlled institutional purchasing policy, the lure of opportunity and proliferation of the PC was hard to restrain. The imagination and creativity of many functional managers were fired by the PC's potential: hundreds if not thousands of new types of applications could be developed. The earliest movements to decentralize computing brought on new demand and interest; computing grew more influential in the business culture and the collective consciousness.

Computers Get Smaller, Cheaper and Omnipresent

By the 1980s, large firms were no longer the only organizations depending on computing. Both the cost and complexity placed computing within the reach of medium and many smaller organizations. This trend continues today. Moreover, the shift in the nature of business and organizations reshaped through computing empowers many small businesses to flourish. Consider the result of a 2015 SBE Council Report:

"In 2011, according to US Census Bureau data, there were 5.68 million employer firms in the United States. Firms with fewer than 500 workers accounted for 99.7% of those businesses, and businesses with less than 20 workers made up 89.8 %. In the number of non-employer firms in 2012 – there were 22.7 million in 2012 –the share of US businesses with less than 500 workers increased to 99.9%, and the firms with less than 20 workers increased to 98% (SBE Council, 2015). Clearly, there was considerable growth in the base and with it, small to medium sized companies brought computer technology into their operations, utilizing them in similar ways for similar purposes.

Hence, hindsight reveals a number of very significant factors that developed during the mid-1980s and 1990s that produced the basis for a remarkable growth-pattern of business computing during the latter 1990's and beyond. Among these factors were: growth of the electronics industry in building new "chips", cheaper disk storage, and un-paralleled communications capability. What also emerged were patterns of expansion and contraction as the tools of computing, and creative (and profitable) applications thereof, advanced. However, what seems to be the case in most of the boom/bubbles created around computing is that the essence of the artifact had not changed fundamentally, whereas uses did as computing expanded, improved, and was ever more available.

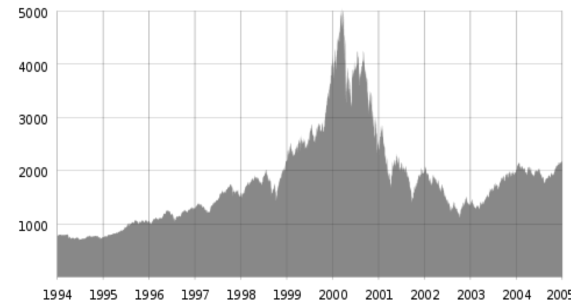


Figure 1 - The NASDAQ Composite index spiked in the late 90s and then fell sharply as a result of the dot-com bubble. https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=dot-com%20bubble

Widespread Proliferation

Computing witnessed another bubble around 2000, not unlike the case with the PC: the dot-com bubble. While the bubble itself did not take place until just after 2000 (WiseGEEK, 2015) several technological advances led the business community to invest very large sums of money in new internet dependent start-ups. More than 60% of these organizations failed. What both the PC and Dotcom bubbles reveal is how little is known about what computing is and is not. Fad-like and fleeting, extant ideas were repackaged in new buzzwords and theories where, in fact, the essentials of computing were not intrinsically different. Certainly, advances in computing hardware are there, as is the proliferation of content via the Internet and its TCP/IP architecture, but these are evolutionarily advanced upon us which, as they proliferated, of which industrious and entrepreneurial individuals took advantage. There can be little doubt that the significant effort released by the CERN group (Berners-Lee, 1991) as reflected by Lee's discovery of the WWW. In general, there is lack of public awareness and understanding of computing which has also given rise to other peripheral phenomenon like information security and privacy, specialization in network and systems administration, and what is commonly referred to as "IT." However, at its core, the actions and goals driving end-user computing as well as their development, and the "dot-com" era, are still rooted in the activities described by Gordon Davis – "…the development and deployment of systems." With this widespread proliferation of massively inter-connected computing, society grew so enamored with what computers "can do" they began to ignore the fact that people need to know "how computers do it;" this leads to the illusion that knowing "how to use" and "how to develop a computer system" are somehow isomorphic.

Rise and Fall

The context of the computing disciplines, as they exist today, is best understood if the conditions surrounding the dot-com bubble are understood. Rather than focus on economic theory surrounding the propensity for "bubbles" to form in concert with the ebb and flow of market forces, it is best to discuss the evolution of computing such that the conditions that fostered the dot-com bubble are understood. To understand them is to realize that the fundamentals of computing continue to move at an evolutionary pace, interspersed with remarkable advances in the power and availability of computing.

The brief (and interpretive) recounting of computing history provided thus far reveals patterns in the computing disciplines that reflect the nature of computing as mechanized information processing – a nature that does not change with the advance of the technologies that mechanize it. We do not seek to identify these fundamentals in an exhaustive and conclusive manner; we only seek to illustrate that our own history can partially refocus who and what we are, as educators and practitioners of the computing discipline generally known as information systems. Our position taken here is that to discount or ignore those fundamentals is folly that cannot be overcome by technological advances.

2. FACTORS CONTRIBUTING TO THE ONSET OF THE "DOT-COM" ERA

Millions of Potential Computers

The 1983 TIME article regarding PC, if nothing else, put people and organizations on notice there would be a lot of computing capacity for all who were desirous of it. As the hardware advanced, multi-user operating systems also advanced supporting remote computer access and otherwise inter-connecting computers. The infrastructure enabling the inter-computer/inter-user data connections were mechanisms first used in the 1960s and, via evolutions in computing, are still the foundation of what is in use today. However, there is little doubt (historically) that networking is the technology that really transformed an otherwise home computing environment into a connected purposeful structure (Berners-Lee, 1991). Networking extended systems and their information to wider contexts and more people. Thus, the impact of computing has been profoundly felt amongst people and that awareness of the "what" was possible with computing also increased.

3. FACTORS WHICH MAY HAVE ACCELERATED THE COLLAPSE OF THE DOT-COM BUSINESS

The WWW is both a plus and minus to business in the 1995-1998 era. On one hand, since the hypertext protocol would run on the average office PC, any PC with the right hardware (A network interface card to use the Ethernet standard) and software (a web browser), could connect to any other host (PC, server, or otherwise) which was discoverable using a combination of TCP/IP and DNS. Since a "server" could be located anywhere, then the PC could serve as an input device for a program running on the server in a multi-user environment. A catalog order-entry inventory system would be a possible system to utilize such an architecture — Amazon comes to mind as a company that capitalized early on the mechanics that eventually promulgated most of the successful "dot-com" ventures. Ideas of this magnitude are very attractive. However, it is important to revisit the following proposition: as computing grows and continues to change the human (and natural) environment, many humans remain unaware or ignorant of the fundamental nature of what computing is (and is not). As is the case with many human endeavors, while computing is behind a plethora of successes, computing also has much to teach about failure. It is not that the hardware is principally flawed (although possible, these errors are usually ironed out), it is rather the "soft systems" that are often causal in computing failure. To wit, an entire sub-discipline of computing is alive and well, and will likely remain so into the distant future, because of human failure to understand computing: IT support. Adages like PEBCAK (Problem Exists Between the Chair and Keyboard) are humorous responses to real problems – a fundamental failure to become familiar with what is computing.

As we ruminate on the dot-com phenomenon in order to understand the progression of computing, it seems that business owners and investors had not read the Standish Group's

CHAOS Report (1995, 2001) which cited significant failure statistics for any IT/Software project. Indeed, they predicted the failure statistics for new software to be at about 80% - which means 4 out of 5 software projects (of which nearly all significant "IT Revolutions" have centered on) will fail. The translation of these results to a business was that business planning and requirements were poorly conducted if at all. During the late 1990's it was also uncertain if requirements could be translated into good code. Unfortunately, investors lost track of the facts and in the dot-com area spent all of their cash on the anticipated success of these ideas. As all of investors' funds were consumed, the "dot.bust" era resulted in the loss of an enormous of cash. Why? Again, we proffer a simple proposition: the average individual is largely ignorant as to what computing is (and is not), and what it takes to generate a good solution. Moreover, the Standish Group's CHAOS report presents the case that this ignorance is often willful and deliberate. While most would agree that the age of enlightenment, science, and significant progress brought about by technologies has negated a tendency to invest in magic, and yet evidence continues to hold that computing is largely treated as though it were magic.

4. LESSONS FROM FAILURE: CURRICULA NECESSARY FOR DEVELOPMENT

The motivation for this section is based on a Failure Analysis of Information Systems of the Standish Group (1995, 2001). While we are sympathetic to software development firms as well as the academics who trained the workers who wrote the defective software, we must highlight these failures as a reminder of our raison d'être. As an illustration, we reflect on two such information systems failures:

a. The Affordable Care Act website launch of 2013
b. Denver Airport Baggage handling system failures of 1995

Both of these systems fit the general observation of Standish Group failures:

a. Totally inadequate business planning and requirements development
b. Poor project management
c. Enormous cost overruns
d. No concept of task
e. Many intended specs abandoned
f. Embarrassed leaders who tended to lie about results and cover them up

In the case of the Affordable Care Act website launch, it turns out that 55 contractors were involved in developing and deploying this failed system launch for a considerable cost of $400 million. It is somewhat exacerbating to learn that the company that fixed it was a small agile team in which the cost of repair was only $4 million. Given the controversy surrounding the legislation itself, it would seem that the failures of the ACA website would have been avoided, but many of the root causes lie within the ACA project itself and have become well-known through studies and reports on software systems development over the years. Again, one must question the levels of ignorance that persist in projects that seek to harness the power of computing.

The exposition of aspects of the history of computing, and the degree to which we can relate this history to the computing disciplines, has been presented as a means of understanding the task before those who consider curriculum design for computing disciplines such as information systems. Among the issues faced is occurrence of the boom-bust cycle as the evolutionary advances of computing build to the point that revolutionary uses are manifested in a rapid manner in markets. Thus, the demand for professionals proficient in computing rapidly expand during these times. Throughout the short history of the computing professions, both market and management failures have not lead to consistency in how computing is situated within society. What professional imperatives exist, if any, when we continue to grapple with the failure statistics presented by the Standish Group. When these matters are considered from an educator's perspective, one must look to what can and cannot be done in academia. We develop curricula to serve as a guide that gives cohesion, focus, and definition. What responsibility do we have in academia to share in the blame for these failures? Are we lacking in project management? Surely, and perhaps certainly, in the last 20 years it is likely that these failed projects have involved well-qualified, certified, and perhaps even experienced project managers. Even if our computing curricula, particularly in the case of information systems, were to be turned over to being largely about IT project management, we contend again, propositionally, that public willful ignorance of computing would doom these individuals. Verily, through the experiences of some of the authors of this paper, many undertaking training in the various forms of IT and software project management regimens and paradigms do not really understand what is computing. For some programs, appending "computer" prior to information systems is not a glib marketing strategy; it is a reminder of the primacy of understanding computing. As an extension, we reflect on Gordon Davis' words - Information systems is about the development and deployment of systems – we realize that these "systems" rely on a non-trivial understanding of what is computing and what is required to make artifacts which reliably serve this purpose. Software engineering has arisen to take this mantle, and challenges information systems educators to decide who they are, what they represent, and how this shall be delivered as a curriculum to produce graduates who are useful in the computing discipline.

Orienting Curriculum Design

Persistent failures in information systems implementation can also be seen as an opportunity. A possible "prescription of a cure" to inoculate computing professionals from the "folly" of technology solutions is to design our curriculum such that computing professionals understand the limits of the mechanization and the critical essence of understanding "information and organizational processes" in computer system asset value. Put metaphorically, "building safe and functional furniture rests more in the skill and knowledge of the woodworker's tools then it does in the intended purpose for the table or chair!" Our task in the development of information systems curricula is to provide this assurance. Moreover, we should ensure that society is concerned for and supportive of an appropriate computing curricula. Our curricula should underscore how and why professional competency is relevant.

We suggest that striving for excellence in information systems must involve great education for students and professionals. We further suggest generation of great systems must revolve on teams of IS professionals who are skilled in the applications as stated herein.

A. For great systems:

1) A set of clear exit objectives toward which curriculum productions are aimed. These objectives, or outcomes, will describe the behavior of IS professionals in developing excellent information systems;
2) A body of knowledge representing all the necessary ingredients of curricular skill specifications. These skills must describe behaviors which when combined appropriately and executed will reliably produce behaviors associated with the exit objectives;
3) A clear pathway which may be constructed connecting the body of knowledge and derived productions to the exit objectives wherein sub-exit-knowledge products which provide successively, a linked network mapping the body of knowledge to the exit objectives.

B. For systems built without excellence—potential failure:

Lack of excellence in IS development—i.e. failure—is:

1) Expensive to organizations as well as individual investors;
2) Represents an unwillingness to search development methodologies for steps that might lead to success, including project management.

Figure 2 moved to the appendix (Curriculum Models: A) is a general model, B) is the model of IS'97 model, and C) is the Model of a to-be-proposed 2016 CIS model curriculum.)

We are impressed with the vastness of the Standish group 44,000+ samples which represent a significant degree of failure. Likewise, their focus on what makes a successful system gives great credibility to their recommendations. Appendix 1 is a set of abstractions from our analysis of the Standish Groups effort. We argue that these statements represent a positive set of elements that might well represent the most significant requirements for a body of knowledge (Figure 2C): If you followed these constructs we are determined you would have the capacity of generating a great system.

Appendix 2 is the detailed body of knowledge (see also Figure 2C) recently determined by survey of a group of professionals (Longenecker, 2015). Within this document is Appendix 3, which is in an abstraction of Appendix 2. For a challenge, the fidelity of the mapping of appendix 3 to Appendix 1 was studied. For each element of Appendix 3 we then asked if the individual element provide support for one or more elements of Appendix 1. The procedure was repeated for the abstracted body of knowledge, Appendix 3. For both appendix 2 and 3 we found a very good match of new proposed body of knowledge (Appendix 2 and 3) with the Standish group recommendations (Appendix 1).

Appendix 4A contains the body of knowledge for IS'97. The elements of Appendix 4A map reasonably well to the exit skills shown in Appendix 4B determined by the IS'97 task force.

Information Systems curriculum development is supported by previous model curricula. Earlier models contain work that has been expanded over the years as the field has grown. The early documents initiated and paved the way for current work. These documents include:

- ACM 71 – Teichroew, 1971; Nunamaker et al 1982
- DPMA 80
- DPMA 86
- AITP 90 – Longenecker et al, 1991
- AITP 95 – Couger et al, 1995; Gorgone et al, 1994; Longenecker et al, 1995
- AITP 97 –
- AITP/ACM/AIS 2002 – Gorgone, et al; (Landry et al, 2000)
- AIS/ACM 2010 – Topi, et al, 2009, 2010

Documents most pertinent to students educated during the dot-com era were those of IS'95 and IS'97. The Body of Knowledge for these documents were identical and are reprinted as Appendix 4. These elements although they were widely reviewed and accepted, do not map very well to appendix 1. Therefore, and unfortunately, the exit objectives of Appendix 4B do not map well to the currently identified elements of Appendix 1. The meaning of this incomplete mapping is that information systems curriculum constructed (Figure 2B) based on these objectives may well be deficient.

5. CONCLUSIONS

We have presented a case that, by understanding some key transitions in the history of computing, coupled with the great promises and failures of computing technologies over the years, we may better know why the information systems discipline may exist and what purpose it may serve. Many could bear testimony that information systems, as an academic discipline, has benefitted from the many "rising tides" across the last 60-odd years of computing as an academic pursuit. However, we have also, directly or indirectly, been party to the failures of computing. This is an odd sentiment to express as computing, in and of itself, is mathematically sound such that "computing" isn't failing but rather, it is our utilization of computing to deliver information systems that are useful, reliable, and robust. The Standish Reports have been examined to illustrate the point that, just as the fundamentals of computing are consistent as they evolve (thus far), so too are the root causes of failure. When an industry produced "failed" or "distressed" projects 4 out of 5 times, there certainly is enough blame to go around. In an attempt to connect the various high-water marks in our history, we propose that the point is past where collective willful ignorance can stand much longer. In the evolutionary process of defining and refining a model curriculum for the discipline, it would seem that a centerpiece of renewed efforts to define and refine should take the Standish Group (and that of others) findings and utilize these persistent problems as a base. While the marketplace will always provide those who can connect the promises of computing technologies to consumers, we must not wait for rising tides to define a forced discipline upon us. Perhaps starting with persistent failure is a good way to define our purpose and worth.

In conclusion, the fidelity of the mappings between appendences 2 and 3 with appendix 1 indicate if we build a curriculum based on our body of knowledge, we may be able to avoid the issues that plagued traditional software developers. We feel that this method of triangulation will produce the most useful results.

6. REFERENCES

ACM Curriculum Committee on Computer Science 1968. Curriculum 68: Recommendations for the Undergraduate Program in Computer Science. Communications of the ACM, 11:3, March 1968, pp. 151-197.

Berners-Lee, T (1991). "The birth of the world wide web," retrieved at http://timeline.web.cern.ch/timelines/The-birth-of-the-World-Wide-Web

Couger, J. D., Davis, G.B., Feinstein, D.L., Gorgone, J.T. and Longenecker, H.E. (1997). IS'.97: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, Data Base, Vol. 26 No. 1, pp. I-94.

Couger, J. D., Davis, G. B., Dologite, D. G., Feinstein, D. L., Gorgone, J. T., Jenkins, M., Kasper, G. M. Little, J. C., Longenecker, H. E. Jr., and Valachic, J. S. (1995). IS'95: Guideline for Undergraduate IS Curriculum, MIS Quarterly (19:3), 1995, pp. 341-360.

Davis, G., J. T. Gorgone, J. D. Couger, D. L. Feinstein, and H. E. Longenecker. (1997). IS'97: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. ACM SIGMIS Database, 28(1).

Davis, G.B., Couger, J. D., Feinstein, D.L., Gorgone, J.T. and Longenecker, H.E. "IS '97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems," ACM, New York, NY and AITP (formerly DPMA), Park Ridge, IL, 1997.

Dot-com Bubble, David Schepp (2001). 'Warren Buffet: I told you so', BBC News 3-13-2001, retrieved at http://news.bbc.co.uk/2/hi/business/1217716.stm

DPMA 1981. DPMA Model Curriculum, 1981. Park Ridge, Illinois: Data Processing Management Association.

DPMA 1986. DPMA Model Curriculum, 1986. Park Ridge, Illinois: Data Processing Management Association, 1986.

Gorgone, J.T., Davis, G.B. Valacich, J., Topi, H., Feinstein, D.L. and Longenecker. H.E. (2003). IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Data Base 34(1).

Gorgone, John T., J. Daniel Couger, Gordon B. Davis, David L. Feinstein, George Kasper, and Herbert E. Longenecker 1994. "Information Systems '95," DataBase, Volume 25, Number 4, November 1994, pp. 5-8.

Kautz, Karlheinz, Sabine Madsen, and Jacob Nørbjerg. "Persistent problems and practices in information systems development." Information Systems Journal 17.3 (2007): 217-239.

Landry, J. P., Longenecker, H.E., Haigood, B. and Feinstein, D.L.. 2000. "Comparing Entry-Level Skill Depths Across Information Systems Job Types: Perceptions of IS Faculty," Proceedings of Sixth Americas IT 2008. IEEE/ACM Joint Task Force on Computing Curricula.

Information Technology 2008, Curriculum Guidelines for Undergraduate Degree Programs in Information Technology, ACM and IEEE-Computer Society, November 2008. Retrieved at http://www.acm.org/education/education/curricula-recommendations

Longenecker, H.E., Feinstein, D.L., Couger, J.D., Davis, G.B. and Gorgone, J.T. (1995). "Information Systems '95: A Summary of the Collaborative IS Curriculum Specification of the Joint DPMA, ACM, AIS Task Force," Journal of Information Systems Education, Volume 6, Number 4, pp. 174-187.

Longenecker, H. E., Jr., D. L. Feinstein, J. D. Couger, G. B. Davis, and J. T. Gorgone (1995). Information Systems '95: A Summary of the Collaborative IS Curriculum Specification of the Joint DPMA, ACM, AIS Task Force. Journal of Information Systems Education, Volume 6, Number 4, pp. 174-187.

Longenecker, H.E., Feinstein, D.L. and Babb, J.S. (2013). Is there a need for a CIS Model Curriculum?, Proceedings of ISECON, San Antonia 2013.

Nunamaker, J.F., Couger, J.D. and Davis, G.B. (1982). "Information Systems Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs," Communications of the ACM, Volume 25, Number 11, November 1982, pp. 781-805.

Segal, B (1995). "A Short History of Internet Protocols at CERN," CERN IT-PDP-TE, retrieved at http://ben.home.cern.ch/ben/TCPHIST.html

SBE Council, 2015. Small Business Facts and Data, retrieved at http://www.sbecouncil.org/about-us/facts-and-data/

Teichroew, D (1971). "Education related for the use of computers in organizations", CACM 14,9 (September 1971).

The Standish Group 1995. retrieved at http://www.projectsmart.co.uk/docs/chaos-report.pdf

The Standish Group Report 2001. "Extreme Chaos" retrieved at http://www.cin.ufpe.br/~gmp/docs/papers/extreme_chaos2001.pdf

Topi, H., Valacich, J., Wright, R.T., Kaiser, K.M., Nunamaker, J.F., Sipior, J.C., and Vreede, G.J. (2010). IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems, Association for Computing Machinery (ACM), Association for Information Systems (AIS)", retrieved July 14, 2012: http://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf

WiseGEEK (2015)."What was the dot-com bubble?," retrieved at http://www.wisegeek.org/what-was-the-dot-com-bubble.htm

Figure 2

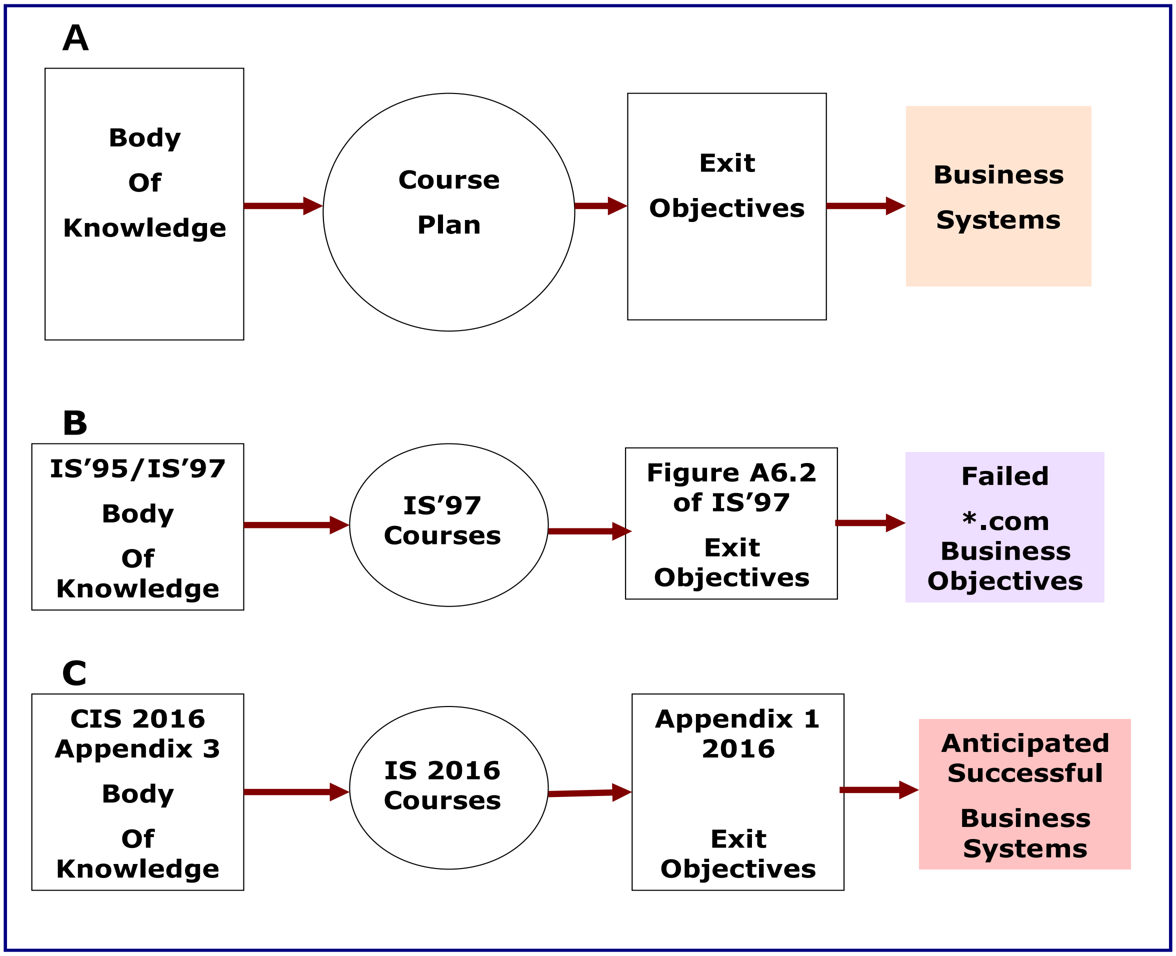


Figure 2. Curriculum Models: A) is a general model, B) is the model of IS'97 model, and C) is the Model of a to-be-proposed 2016 CIS model curriculum.

Appendix 1 – Successful Exit Objectives Statements Abstracted from The Standish Group (2001, 1995).

EO-1 Accurate business plan developed by end users, management and development team
- Identify stakeholders; ensure executive support
- Identify and qualify business knowledgeable project manager to deliver a competitive business plan
- Establish user—developer—management interactions to ensure involvement, and clear business objectives
- Choose a development methodology (e.g. Agile, Lean UX…)

EO-2 Exceptional requirements analysis
- Must use a User-Centered Focus
- Must express IT alignment with a high degree of maturity
- Identify Deployment System Requirements
- Must be tied to a verification and validation mechanism
- Must involve excellent team, personal and interpersonal skills

EO-3 Translation of Requirements info viable software
- Should consider using Agile approach
- Must use well established software engineering and programming practices, including reuse
- Must have exceptional database modeling and implementation skill
- Must apply quality principles

EO-4 Deploy Software Product
- Install system on IT host
- Test Software
- Test System and certify

EO-5 Project Management based on established formal written methodology
- Initiate project thoroughly understood by the project manager
- Establish project communication
- Must set important milestones and check points
- Perform project risk management
- Assure project/product security
- Utilize reusability
- Develop WBS tied to system development life-cycle
- Establish configuration management
- Execute project subject to quadruple constraint; minimize scope
- Use project control tools PERT/Gantt; requirement tools; collaborative tools

Appendix 2 – Knowledge Areas Sorted by Depth of Knowledge CIS Students Must Master in an Undergraduate Curriculum for Depth 2.5 or Greater

KA	Knowledge Areas Sorted by Expert Expected Depth of knowledge	Depth 0-5	Emphasis theory -50 practice +50	Expert Confidence
A22	Database	4.3	28	97%
B03	Data Retrieval and / Manipulation with Database Languages	4.0	26	93%
A23	Analysis and Specification / of System Requirements	4.0	19	90%
A19	Analysis of Business / Requirements	3.9	20	94%
A21	Information Systems Design /	3.9	21	93%
C01	Programming / Fundamentals	3.8	32	94%
A07	Web Systems and / Technologies	3.8	32	92%
A15	Approaches to Systems / Development	3.7	24	96%
G08	Project Plan, Scope, and / Initiation	3.7	21	90%
D14	Systems Analysis & / Design	3.6	23	92%
B05	Data and Information / Modeling at Conceptual and logical Levels	3.5	17	90%
C09	Software / Requirements	3.5	18	93%
A24	Team and Interpersonal / Skills	3.5	20	94%
G07	Leading Project / Teams	3.5	21	91%
B01	Database Systems and / Distributed Databases	3.4	18	86%
C15	Software / Design	3.4	20	90%
A06	Information Technology / Fundamentals	3.4	23	91%
D09	Systems Development / Concepts and Methodologies	3.3	15	91%
C05	Human Computer / Interaction	3.3	17	92%
C16	Software Development / Fundamentals	3.3	25	92%
B12	Data Integrity and / Quality	3.2	19	91%
A20	Information and Business / Analysis	3.2	15	90%
D11	Systems Implementation and / Testing Strategies	3.2	16	89%
C06	Module Design and / Construction	3.2	18	91%
C19	Software / Testing	3.2	21	91%
G10	Project Execution & / Control	3.2	20	87%
A01	Impact of Information / Systems on Organizational Structure and / Processes	3.1	15	91%
D06	System Deployment and / Implementation	3.1	17	89%
B07	Physical Database / Implementation / Data Definition Language	3.0	16	92%
A13	Business Intelligence and / Decision Support	3.0	14	90%
M03	Basic Scripting/ / Programming	3.0	25	92%
B04	Teams and Interpersonal / Skills	2.9	15	89%
M01	Basic Data / Analysis	2.9	21	86%
G12	Project / Quality	2.9	8	82%
C13	Security and Privacy, / Vulnerabilities, Risks, Mitigation	2.9	10	78%
B08	Stored Procedure / Implementation	2.8	20	85%
B10	Data and Database / Administration	2.8	18	88%
A03	Identification of / Opportunities for IT enabled Organizational / Change	2.8	10	92%
A16	Different Approaches to / Implementing Information Systems	2.8	15	91%
C02	Programming / Languages	2.8	14	92%
C17	Software / Construction	2.8	30	89%
G06	IS Project Strategy and / Management	2.8	8	85%
G03	Establishing Project / Communication	2.8	13	88%
G09	Work Break-down / Structure	2.8	18	90%
G13	Project / Closure	2.8	13	84%
E04	Networks and / Communications	2.8	15	81%
H02	Probability and / Statistics--Basic probability theory, random variables and / probability distributions, estimation theo...	2.8	8	82%

B11-	Data Management and / Transaction Processing	2.7	22	90%
A17-	Business Process Design and / Management	2.7	15	89%
A26-	Computer / Networks	2.7	21	89%
H01-	Math and Statistics for / IT	2.7	18	87%
D15-	User / Experience	2.6	16	89%
G11-	Project / Standards	2.6	11	83%
A02-	Individual and / Organizational Knowledge Work Capabilities	2.6	14	89%
C04-	Integrative Programming and / Technologies	2.6	18	78%
B13-	Security attacks and / mitigations	2.6	15	84%
B06-	Scripting	2.5	25	91%
A09-	Enterprise / Architecture	2.5	4	79%
D07-	System Verification and / Validation	2.5	12	86%
A25-	Configuration and Change / Management	2.5	12	85%
C03-	Programming / Environments	2.5	16	87%
C07-	Software Engineering / Process	2.5	15	89%
C18-	Software / Quality	2.5	11	88%
C20-	Software / Maintenance	2.5	17	89%
D05-	System Integration and / Architecture	2.5	13	83%
F02-	Information Assurance and / Security	2.4	3	78%

Knowledge Areas on Required Depth for an Undergraduate CIS Major (Longenecker H, et al, 2015.) The above is a partial list of knowledge areas 2.4 and above. These areas will involve the most effort through the cognitive load they demand. It is noteworthy that these knowledge areas map very closely to the demands of the Standish Group specification (see Appendix 1).

Appendix 3 – CIS Body of Knowledge (9-3-2015)

BK-1 Database
- Database Components (entities, attributes, relationships, drawing, scripting)
- Database Structuring (Create, Modeling, Quality, integrity, data types, data, and indexes)
- Database Access (DDL, DML, Transaction Processing, Stored Procedures; blocking injection attacks)
- Database Services (ETL, Report Services, BI, DSS, Backup, Replication, Security Management, Administration)

BK-2 Information System Development
- IS Development: Planning; intellectual purpose; Feasibility; privacy; security; alignment security
- IS Development: Make or Buy
- IS Problem Definition, Requirements Elicitation; BPR Analysis
- IS Organization Development with New IS (IT enabling, improved IT alignment, lower resistance, raise involvement)
- IS Design Maturity (levels within apprenticeship, design-leadership)
- IS System Verification/Validation Planning
- IS Development Test Plan
- IS Verification with Customer
- Assertion of Quality Policy
- IS Test and Validate (Module, Application, System)
- IS Final Evaluation, Deployment and Operation
- IS Team and Interpersonal Skills (Leadership, Empowerment, Change, Meetings, Teams, Innovative learning)
- IS Life Cycle Tools (Methodologies, Support Systems, Bloom and Learning for Clients and Students)

BK-3 Information Systems Design
- IS Design Architecture, Frameworks, Creativity, Reflection, video, voice
- IS Application Design (Requirements, Modules, Verification)
- IS Design Paradigms (cash management, new accounts, new addresses, new organization interaction, international actions, interfaces management, security procedure, Sarbanes Oxley, HCI management, HH device utilization)
- IS BPR, Data Transformations, Reporting, and BI
- IS Design Standards, Privacy and Security, Policies, Regulation and Compliance
- IS Design Quality (Verification and validation, qualitative and quantitative-assessments)
- IS Systems Testing and Implementation
- IS Configuration and Change Management

BK-4 Software Planning, Programming, Testing
- Programming Logic and Design (computers, programming, programs, control structures, sequence, selection, loop, arrays, records, modules, parameters, OO, events, files and DB)
- Programming Implementation (Languages, Environments, Compilers, Local, Web Environment; Code-a-little--test-a-lot; scripting)
- Languages (C++, C#, VB.net, Java Script, HTML, ASP)
- OO Programming (OO Structures, concepts, implementation with an IDE, testing
- Software Engineering (Requirements, Simple Algorithms and Data Structures, Modules, Box Structured Design, Programming, Quality)
- Software Implementation (Requirements, Design, Modular Top Down Implementation, Testing, Validation, packaging, installation, operation)
- Software Management (Development, Maintenance, documentation, standards, performance)

BK-5 IS Project Management
- Project Initiation (Strategy, Stakeholder analysis, Plan, Scope)
- Project Communication (Classification, Frequency, Responsibilities, Monitoring)
- Project Staff (Function, Responsibilities, Qualification, Reporting)
- Risk Management
- Work Break Down (Structure, Schedule--upcoming and completed events)
- Teaming (ensure team training, ensure leadership development, managing disputes)

Project Execution and Control (quadruple constraint, controlling activities, negotiating changes; ensure standards & quality; tools: Gantt, PERT)
Project Closedown (Acceptance Reviews, Final Reporting)

BK-6 Technology, IT Management and Security
IS Professionalism (systems thinking, organizational behaviors, legal issues, ethical issues, social issues, concepts of performance, practicing success habits, life-long learning)
Using IT governance (ITIL, regulatory standards, compliance)
IA Fundamentals (Vulnerabilities, Risks, Mitigation, threats, attacks, incident management, Security Policy Principles and Design)
Computer Architecture and Organization
Parallel and Distributed Computing
Devices (cable, fiber, modem, router, switch packet shaper, protocols, servers, sniffers)
Networks and Communications, security issues
Operating Systems Concepts, security issues
Storage management systems
System Operation, Administration and Maintenance
Virtualization and zero client

Appendix 4A – Body of Information Systems Knowledge for IS'95 and IS'97

1.0 Information Technology
- 1.1 Computer Architectures
- 1.2 Algorithms and Data Structures
- 1.3 Programming Languages
- 1.4 Operating Systems
- 1.5 Telecommunications
- 1.6 Database
- 1.7 Artificial Intelligence

2.0 Organizational and Management Concepts
- 2.1 General Organization Theory
- 2.2 Information Systems Management
- 2.3 Decision Theory
- 2.4 Organizational Behavior
- 2.7 Managing the Process of Change
- 2.8 Legal and Ethical Aspects of IS
- 2.9 Professionalism
- 2.10 Interpersonal Skills

3.0 Theory and Development of Systems
- 3.1 Systems and Information Concepts
- 3.2 Approaches to Systems Development
- 3.3 Systems Development Concepts and Methodologies
- 3.4 Systems Development Tools and Techniques
- 3.5 Application Planning
- 3.6 Risk Management
- 3.7 Project Management
- 3.8 Information and Business Analysis
- 3.9 Information Systems Design
- 3.10 Systems Implementation and Testing Strategies
- 3.11 Systems Operation and Maintenance
- 3.12 Systems Development for Specific Types of Information Systems

IS Body of Knowledge for IS'95 and IS'97. This body of knowledge was widely reviewed both by academic and industry professionals and became the basis for IS'95 and IS'97. This body of knowledge is at best partially supportive of Appendix 1. At best it represents a very incomplete match to the thought process that emerged and was specified by the Standish Group (1991, 1995).

Appendix 4B – Exit Objectives (from IS'97, Figure A6.2)

D. Systems Development

D.1 Software Development

Is Life Cycle: Developing With Packages
Implementing And Event Driven Applications
Is Application Development/Code Generate
Is Database And Is Implementation
Is Database Application Implementation
Is Database Application Structuring
Is Development Testing
Is Applications, Production Systems

D.2 Database

Database Terminology And Concepts
Implementing A Simple Database Design
Is Database Applications Development
Is Data Modeling
Is Database Conceptual/Logical Models

D.3 Sys Analysis/Design

Info Analysis: Individual Vs Group
Info Analysis: Finding Is/It Requirements
Is Development Standards
Is Development Risks/Feasibility
Is Conversion Planning
It Systems Specification
Problem Solving, With Packages
Is Analysis And Design Tasks
Is Continuous Improvement And Is
Is Design And Implementation
Is Rapid Prototyping
Is Requirements And Specifications
Systems And Quality Metrics/Assessment
Is Development And Conversion
Is Functional Specifications
Is Requirements And Database

D.4 Teams/Interpersonal

Interpersonal, Synergistic Solutions
Interpersonal, Consensus Development
Interpersonal, Group Dynamics
Interpersonal, Agreements/Commitment
Personal, Time/Relationship Management
Personal, Presentation
Interpersonal, Empathetic Listening
Interpersonal, Goal/Mission Alignment
Personal, Proactivity/Principled Action

D.5 Project Management

Is Development Project Close Down
Is Development And Project Management
Quality And Performance Management
Is Development Project Planning
Is Development Project Management
Is Development Project Management
Is Development Project Management Tools
Is Life Cycles And Projects

E. Is Deployment And Management

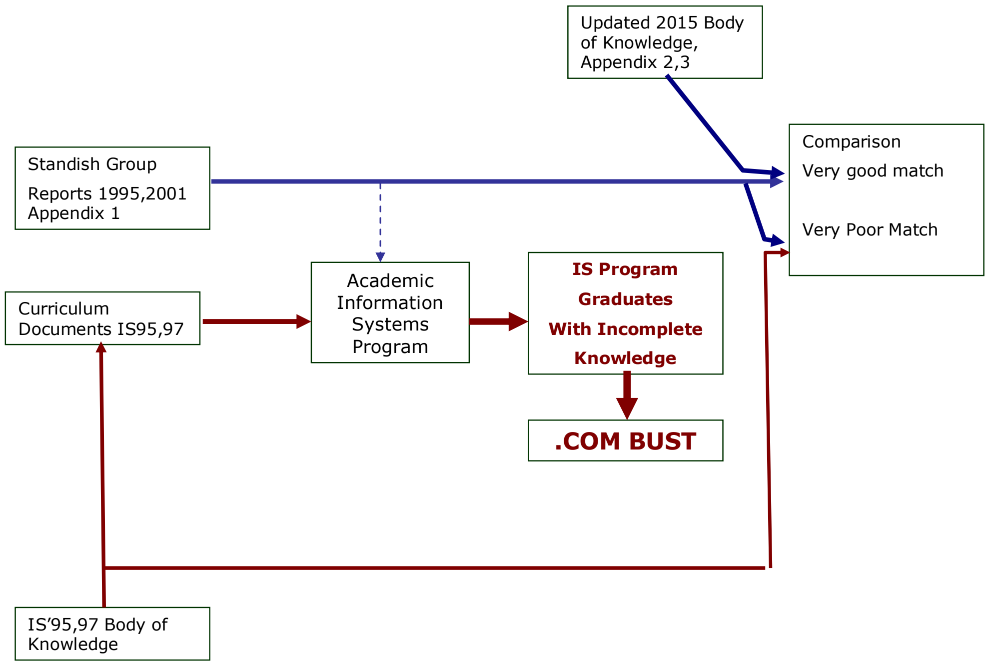
E.1 Support Services
Personal, Life-Long Learning

E.2 Systems Integration
Telecom, Installation, Implementation
Telecom, Lan, Install, Configure
Os, Install Multi-Media
Os, Interoperability And Sys Integration
Os, Install Multi-User System
Is Commercial Implementations

E.3 Management Of Is The Function
Is Careers
Personal, Performance Evaluation
Is Implementation, Outsourcing
Is Policies And Standards
Is Management And Dept Organization
Is Responsibility To Sell Designs To Mgt
Personal, Leadership And Is

E.4 Information Resource Management
Is Implementation And Outsourcing
Information Use Strategies

Appendix 5 – 2015 Updated BK Matches Standish Group Reports, yet a Poor Match is Obtained between IS95, IS97 BK.



Determine Curriculum Sufficiency by Mapping Body of Knowledge to Project Failure Statistics recorded and analyzed by the Standish Group Recommendations (1995, 2001). The .COM Bust occurred because of the incompetency of professionals. Ultimate success required a project manager skilled in business; the technical requirements of software design, programming, and database were necessary, but not alone sufficient!

Introducing IT Strategy in an Introductory Course

David M. Woods
woodsdm2@miamioh.edu
Computer & Information Technology Department
Miami University Regionals
Hamilton, OH 45011, USA

Abstract

Professionals working in technology fields face continuing challenges to be involved in the decision making process about how technology is used by organizations rather than just implementing these decisions. Developing skills for thinking and acting strategically are key skills for our students. This has been recognized by the addition of an IS strategy course in the latest revision of the recommended curriculum, but programs have been slow to add this course. This paper investigates introducing learning activities related to IS/IT strategy in an introductory IT course. Including strategy activities throughout the curriculum could provide an alternative or complement to a dedicated strategy course.

Keywords: IT Strategy, Active Learning, Experiential Learning, IS Curriculum, Pedagogy

1. INTRODUCTION

Strategy is a key concept in the IT and IS curricula. The IS 2010 curriculum (IS 2010) recognizes this explicitly with the IS 2010.7 IS Strategy, Management, and Acquisition course that is recommended for all IS Majors and Minors. Strategy is a broad topic and the IS2010.7 course "explores the issues and approaches in managing the information system function in organizations and how the IS function integrate / supports / enables various types of organization capabilities." Another way to consider IT/IS strategy is the IT/IS activities that help a larger organization achieve its goals.

Elements of strategy can also be found in several of the pervasive themes from the IT 2008 curriculum, including "user centeredness and advocacy," "professionalism (life-long learning, professional development, ethics, responsibility)," and "interpersonal skills" (Information Technology 2008).

Despite this, several recent reviews of IS programs have found a limited presence of strategy courses in these programs. Additionally, a review of the published literature found few examples discussing how strategy is being taught in IS and IT programs. An alternative or complement to a dedicated IS/IT strategy course could be to include strategy throughout the curriculum.

While the IS 2010 curriculum (IS 2010) recommends that the IS 2010.7 course be a capstone course that is "either the last or one of the last courses that students take," it could be useful to introduce these topics earlier in the course of study in preparation for a capstone activity.

This paper will discuss how two strategy activities were included in an introductory IT class in an effort to help students develop a broader view of the IT and IS fields.

2. STRATEGY IN THE CURRICULUM

As discussed in detail in an earlier work (Woods & Howard, 2015), since the adoption of the IS 2010 curriculum, several studies have found limited progress in the addition of the IS2010.7 course to programs offered in the United States.
One early review (Apigan & Gambill, 2010), looking at courses from the preliminary IS 2009 curriculum, found that only 35.4% of the programs reviewed had an IS Strategy, Management, and Acquisition course.

Later reviews of AACSB and ABET accredited programs found similar results with only 29% of 127 AACSB programs reviewed (Bell, Mills, & Fadel) and only 27% of 37 ABET (Feinstein, Longenecker, & Shrestra, 2014) programs having the IS 2010.7 course.

A more recent study (Hwang, Ma, & Wang, 2015) reviewed 2,229 courses in 394 undergraduate IS programs in the United States. This study found that the IS 2010.7 course was only present in 61 (15.5%) of the programs reviewed.

Another study (Mills, Velasquez, & Fadel, 2012) offers a more in-depth analysis of the IS 2010 curriculum, including some discussion about factors affecting adoption of the curriculum.

Some factors that may affect efforts to add a strategy course may include the measured pace of curriculum updates and pressure to reduce the time students need to complete a program resulting in an effort to reduce the number of required courses. An alternative could be to look at including IS/IT strategy topics throughout the curriculum.

3. THE NEED FOR IT STRATEGY

As programs work to update their curriculum and possibly add the IT 2010.7 IS Strategy, Management, and Acquisition course, there is a continuing need to ensure that IS and IT graduates have developed strategy skills. For this discussion, strategy is broadly defined as efforts to develop a high level plan for achieving goals in an uncertain environment. For IS/IT discussions of strategy must be considered in the context of a business or organizations overall strategy.

While the IS 2010 model curriculum was the first to add a required course with a focus on strategy, the topic has appeared in previous model curriculum. A review of IS Curricula (Longenecker, Feinstein, & Clark, 2013) documented the depth of knowledge expected for the skills included in curricula since 1973. Table 1 in Appendix 1 shows skills related to strategy, when they first appeared in a model IS curriculum and the initial and current depth of knowledge expected for the skill. For the three skills shown, all have been in the model curriculum for over 30 years and the expected depth of knowledge has increased with time.

Stories about the failure of large IT projects are regularly in the press. In-depth analysis of these often identify a disconnect between organizational and IT understanding of the goals of the project as factors contributing to project failure. Understanding how IT efforts support the goals of an organization is a key part of IT strategy, and features in the topics and learning outcomes proposed for the IT 2010.7 course.

The "Beyond IT Failure" blog (http://www.zdnet.com/blog/projectfailures/) regularly features IT project failures with in-depth discussion of factors, including strategy failures, contributing to the project failure. For example, failure of a $30 million ERP (Enterprise Resource Planning) implementation (Krigsman, 2010) offers material for a class discussion or case study.

Discussions of "rogue IT" are another category of IT strategy failures (Krigsman, 2013). The term "rogue IT" refers to staff in an organization developing and/or implementing technology solutions without involving the formal IT staff of an organizations. Rogue IT occurs for many reasons, but often reflects an IT strategy that is not aligned with the larger organization's strategy.

Other evidence for disconnects between IT and organizational strategy can be seen in surveys of IT and organizational leaders. A McKinsey study (Khan & Sikes, 2014) found that "IT has become less effective at enabling business goals." Similarly, CIO Magazine's 2015 State of the CIO survey (CIO Magazine Staff, 2015) finds that 54% of line of business executives "view the IT group as an obstacle to their mission" and that only 43% of business leaders view the IT group as either a business leader or business partner.

In addition to benefitting students in their future professional career, developing strategy skills could also benefit students in other ways. Students can also apply skills related to strategy to career planning, lifelong learning, and professional development since all of these involve planning for achieving a goal in an uncertain environment.

With a clear need for IS/IT students to study strategy, but slow adoption of the IS 2010.7 course from the IS 2010 curriculum, what other approaches might be useful? One idea would be to add IS/IT strategy learning activities throughout the curriculum and build to the desired depth of understanding over the student's career. The remainder of this paper discusses a recent effort to add strategy activities to an introductory IT course.

4. AN OVERVIEW OF THE COURSE

The Computer and Information Technology Department at Miami University offers several degree options. At the bachelor's level, students can earn a degree in Information Technology or a focused degree in Health Information Technology. Several associate degrees are also offered.

Among the core courses for all of the degrees offered are two Introduction to IT courses. Both are three credit courses covering fundamental IT topics. The course discussed in this paper is the second of the two courses. It covers a range of topics including computer architecture, data representation, operating systems, a survey of programing languages, and tools used by IT professionals. It also addresses problem solving in an IT context, including algorithms, analysis, development, and testing.

When considering how to include IT strategy activities in the course, several potential approaches were considered, but in the end it was decided to look at how IT strategy is part of IT problem solving and also to extend the discussion of tools used by IT professionals to include tools with IT strategy applications.

Another consideration was whether to introduce IT strategy as a separate module or to address it throughout the semester. In the end, one IT strategy learning activity was designed to run throughout the semester while the other was designed as a small, standalone group project.

The particular course session where the activities discussed in this paper were implemented was a fully face-to-face class format where the class met for an hour and twenty minutes two times a week. For this session, the initial course enrollment was 20 students, with 17 completing the class.

5. IT STRATEGY CLASSROOM ACTIVITIES

One IT strategy activity was introduced on the first day of class as a daily discussion activity. The goals of the activity were to get students to engage with looking at technology developments in the world around us, consider how these technology developments could be used to advance the goals of a business or organization and think about what challenges the technology developments might pose for an IT organization.

Each student was randomly assigned a class meeting when they would be responsible for a brief in class discussion. To prepare for the discussion, the student had to find a short article from a reputable online source. The student posted the article in an online discussion forum in the Learning Management System (LMS) being used for the course one week before the scheduled class discussion. In addition to posting a link to the article, the students were asked to comment on why they found the article interesting.

Students were also told that they didn't need to understand all of the details of their articles, but "should understand the main concepts and be interested in learning more about the idea." To help students with finding articles, several example articles and possible sources of articles were provided. In addition, students were encouraged to discuss potential articles with the instructor.

To prepare for the in class discussion, all students were asked to read the posted article before class on the day it was scheduled for discussion and prepare at least one question for the discussion. Article discussions took place at the beginning of class, with the student who posted the article providing a brief introduction and helping the instructor start a discussion.

The discussions were held at the beginning of each class meeting with the exception of two exam days and one day used for project presentations. Scheduling one article discussion per class period meant the activity covered most of the semester. The schedule was adjusted a couple of times during the semester to deal with students who dropped the course and also a class cancellation due to weather.

To follow up on the in class discussion, students were asked to visit the LMS discussion forum where the article was posted and follow up with at least one follow up comment. This could be the question they prepared before class, something new they learned during the discussion, an idea for how the topic could affect them personally, a question that was not answered in the discussion or a follow up comment on another student post. To assist with the follow up discussion, one student was randomly assigned to take notes on the in class discussion and post a summary to the LMS discussion forum after class.

In total, the three parts of the activity – posting an article, summarizing one discussion, and participating in the LMS discussion forum for all articles – comprised 8% of the total course grade. The components for selecting an article and summarizing a discussion had a fixed number of points, but the grade for commenting on articles was based on the total number and quality of posts a student made. As an added incentive to students, the total possible points included a bonus so that students could potentially earn additional points equal to 2% of the course point total.

Since this was an introductory course with many first year students, the instructor felt that it was important to provide an example so that students would have a clear idea of what was expected of them. To accomplish this, the first discussion, which took place at the beginning of the second week of class, used an article selected by the instructor. A student earned extra credit by volunteering to post the discussion summary.

The other IT strategy activity was introduced near the midpoint of the semester. The goals of this activity were to introduce students to tools and techniques for undertaking balanced evaluations of options and reaching consensus as a group.

This was done through a group activity to look at technology making a difference in the world. The activity was introduced with an in class discussion on the concept of rubrics. An example of developing a rubric for buying a house was used.

In addition to introducing the concept of a rubric, two specific details were also discussed. The need for objective evaluation criteria to provide consistent scores when used by different people was illustrated by discussing a house buying criteria of "good schools." A discussion of what "good schools" meant showed that different people may focus on different aspects of schools. The students eventually identified that an existing state evaluation of schools could be used for a more objective measure.

The second detail that was discussed was weighting criteria in the rubric. The students' initial instinct was to give all criteria the same weight or the same maximum possible score. Use of different weights was illustrated with the house buying example using a criteria evaluating school quality and whether the yard is fenced for a dog. Both are important criteria, but in discussions, students concluded that school quality was more important because it would have a bigger impact and would be harder to change. The class discussed how to reflect this in the overall rubric and settled on giving the school criteria twice the weight of the fence criteria.

For this activity, students were organized into groups of 3 – 4 students for a total of five groups. Each group then met briefly to pick a topic area. To select topics, students were asked to think about a cause that mattered to them and look for how technology could make a different in that area. Examples of topic areas included helping individuals with chronic medical conditions and supporting the education of young children.

Once a topic area was identified, each group member individually identified four technology projects that were making a difference in the topic area. Each individual also developed four criteria that could be used to evaluate the projects. The projects and evaluation criteria were submitted for review and assessment by the instructor.

After this, each group pooled the individual project ideas and evaluation criteria of the members. This meant that each group had 12 – 16 projects and evaluation criteria to work with. From these, the group worked to develop an evaluation rubric with 4 criteria. This rubric was then used to evaluate all of the group's project ideas.

After evaluating their project ideas, the groups used the evaluation results as a starting point and worked to come to consensus on which project idea was the best. During the period when groups were pooling their individual ideas and evaluation criteria to develop a group rubric, one class period was set aside to allow groups to work together. Other than this time and the initial, brief meeting to pick a group topic, all group work occurred outside of class time. Students were provided group areas within the LMS with discussion forums and other collaboration tools. Students also had access to Google Apps for Education tools.

After reaching consensus, the groups developed a 2 -3 paragraph executive summary discussing their best idea and their evaluation process. The executive summary was posted to a discussion forum in the LMS for review by all students in the course.

A week after the executive summaries were posted, each group gave a 10 minute in class presentation. In the presentation, groups were asked to discuss their best idea and at least one other idea that was considered. They were also asked to discuss their decision process, including their evaluation rubric and any other considerations that factored into their final decision.

Overall, the assignment had three components that were submitted and assessed – the individual project examples and evaluation criteria, the group executive summary, and the group presentation. In total, this assignment comprised 7% of the total course grade. Additionally, after the in class presentations, students were surveyed and asked to order the presentations (other than their own) based on how well they met the goals of the assignment. Students who completed the survey received a small bonus.

6. DISCUSSION

From the instructor's perspective, both activities worked well. As is the norm with a new activity, there was room for some improvement in both activities.

Information on student perspectives were collected in an end of semester survey. For each of the two activities, students were asked Likert scale questions about:

- Whether the activity helped them learn.
- Whether they saw value in the activity.
- How much work the activity was.
- How much the activity helped them understand what IT professionals do
- How much they enjoyed the activity
- Whether they would like to do the activity again.

For the rubric development activity, students were also asked a yes/no question about whether they had developed an evaluation criteria in any of their previous courses.

For each activity students were also provided a free form text question where they could offer other comments or suggestions.

The survey was distributed at the end of the semester, and 10 of the 17 students (59%) responded.

For the article discussion activity, there were a number of excellent articles, but also a few that were challenging to discuss.

One student, selected an article discussing the relative security of operating systems (including mobile OS) with data showing that versions of Windows were among the least vulnerable (Khandelwal, 2015). This allowed discussion about evaluating data sources and the need to rely on hard data rather than received wisdom.

Another interesting article discussed an announcement that a major provider of Electronic Health Records ("Patient records", 2015) software was building a data center and planning to offer cloud hosting of their software. This article was especially relevant since many of the students were Health Information Technology majors. This article also allowed discussion of cloud hosted solutions in use at the university including e-mail and LMS.

Through the course of the semester, two good general discussion questions were identified – "How could a business benefit from using this technology?" and "If you worked for an IT organization, how would you be affected if the company adopted this technology?"

A number of articles selected by students involved consumer technology, especially phone apps. This was not a surprise since students are regular users of consumer technology. These articles presented an opportunity to discuss the consumerization of corporate IT, a topic that is an ongoing challenge to corporate IT and IT education (Law, 2013).

Students had no complaints about selecting an article for discussion, but some students did not post their assigned discussion summary. Also, there were some students with little or no participation in the online discussions following the in class discussion. In general student completion of tasks in this assignment was similar to their completion of other assignments in the course.

The student survey showed that all of the respondents agreed that the discussion activity was helpful in learning about the wide range of technology uses. All respondents also saw the value of discussing how the technologies from the article could impact an IT organization.

In terms of the work required, the survey showed that most of the students found it easy to find and post an article, with one student neutral on the question. Similarly, 80% found the activity helped improve their understanding of what IT professionals do, with the remainder neutral on the question.

Students were also asked whether they enjoyed the article discussion activity and whether they would like to the activity again. The majority enjoyed it (70%) and would like to do it again (90%), with the remainder neutral.

Reaction to the group assignment involving developing and using an evaluation rubric was similar. From the instructor's perspective, the main challenge was that a couple of groups struggled to find a topic that all members found interesting.

The individual project and evaluation details that were submitted met the instructor's expectations. A couple of students did not submit their individual contributions, but this wasn't a surprise given their participation in other course assignments.

The groups were given some in class time to discuss and develop their group evaluation criteria and build the final presentation. All students, including students who had not submitted the individual component actively participated in these discussions.

The executive summaries posted ahead of the final presentation were generally good, but focused more the best project identified by the group and less on the selection process.

All of the final presentations met or exceeded the instructor's expectations, especially given that this was an introductory IT course where no prior IT knowledge is required. The most notable thing about the final presentations was the enthusiasm that groups had for sharing the details of their "best" project with the rest of the class.

Examples of "best" projects selected by the groups included:

- An app that used word images to help people with speech impediments and learning disabilities.
- Language learning software for young children.
- Technology to improve monitoring of blood glucose and reduce associated pain and discomfort in diabetics.
- An app that used gamification to motivate individuals to exercise.
- An app to help farmers access and manage data on crop prices.

It is interesting to note that several students shared that they were making use of the app identified by their group. As one student said, "I liked the app so much I bought it!"

In reviewing the evaluation rubrics developed by the groups, it was apparent that all of the groups had understood the need for objective evaluation criteria. A couple of the rubrics had different weights for some criteria. For the other groups, there was no way to tell whether they had not understood the idea or not seen the need for it.

The previously discussed end of course survey also included questions about the group activity. 9 of the 10 respondents agreed that it was a helpful way to learn about a method for making choices and all respondents saw the value in the activity. This activity was seen as more difficult than the article discussion activity, with only 70% of the students seeing it as easy and only 70% reporting that they enjoyed the assignment. Again, 80% found that the activity helped improve their understanding of what IT professionals do.

In evaluating prior knowledge, 80% reported that they had not developed an evaluation criteria in any previous courses. In the open ended comments, one student expressed that they don't like group assignments, even though they see the benefits. Another student commented about the lack of participation by the rest of the group, but still thought the method would be valuable with a more active group.

7. CONCLUSION

While technical skills remain important in the IT and IS fields, technology departments continue to shift the emphasis on IT infrastructure to analytics and innovation to improve business efficiency and effectiveness (Khan & Sikes, 2014). Our students not only need technology skills but they also need to learn about IS/IT strategy so that they can work to help organizations use technology to achieve organizational goals.

The successful inclusion of IS/IT strategy learning activities in an introductory IT course offers an additional way to educate students about this important topic. Even if programs are able to add the recommended IS 2010.7 course, adding IS/IT strategy learning activities in multiple courses could benefit students.

8. FUTURE PLANS

Some minor revisions to the two activities are planned. For the article discussion activity, efforts to encourage more online discussion are needed. For the group activity to develop an evaluation rubric and apply it, specifications for the executive summary and presentation will be updated to ask the groups to include more information about the evaluation rubric. Also, a method for assessing individual's contributions to the group activities will be considered.

I also plan to talk with the departmental industry advisory council to get additional ideas for IS/IT strategy learning activities, especially activities that could allow advisory council members to interact directly with students.

9. REFERENCES

Apigan, C. & Gambill, S. (2010). Are We Teaching the IS 2009* Model Curriculum? *Journal of Information Systems Education*, 21(4), 411-420.

Bell, C., Mills, R., & Fadel, K. (2013). An Analysis of Undergraduate Information Systems Curricula: Adoption of IS 2010 Curriculum Guidelines. *Communications of the Association for Information Systems*, 32, 73-94.

CIO Magazine Staff (2015). *2015 State of the CIO.* Retrieved from http://www.cio.com/article/2862760/cio-role/2015-state-of-the-cio.html

Feinstein, D., Longenecker, H, & Shrestha, D. (2014). A Study of Information Systems Programs Accredited by ABET In Relation to IS 2010, *Information Systems Education Journal*, 12(3), 76-84.

Hwang, D., Ma, Z, & Wang, M. (2015). The Information Systems Core: A Study from the Perspective of IS Core Curricula in the U.S. *Information Systems Education Journal*, 13(6), 27-34.

Information Technology 2008: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. Retrieved on May 8, 2014 from http://www.acm.org//education/curricula/IT2008%20Curriculum.pdf

IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. Retrieved on May 8, 2014 from http://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf

Khan, N. & Sikes, J. (2014). IT under pressure: McKinsey Global Survey results. March 2014. Retrieved on May 7, 2014 from http://www.mckinsey.com/insights/business_technology/it_under_pressure_mckinsey_global_survey_results (free registration required).

Khandelwal, S. (2015). Windows? NO, Linus and Mac OS X Most Vulnerable Operating System in 2014. Retrieved from http://thehackernews.com/2015/02/vulnerable-operating-system.html

Krigsman, M. (2010). Understanding Marin County's $30 million ERP failure. [Blog post]. Retrieved from http://www.zdnet.com/article/understanding-marin-countys-30-million-erp-failure/

Krigsman, M. (2013). Rogue IT: Sad truths and unfortunate stories [Blog post]. Retrieved from http://www.zdnet.com/article/rogue-it-sad-truths-and-unfortunate-stories/

Law, W. (2013). "Consumerization of IT" – Challenges for IS Education. *Information Systems Education Journal*, 11(4), 4-9.

Longenecker, H, Feinstein, D., & Clark, J. (2013). Information Systems Curricula: A Fifty Year Journey. *Information Systems Education Journal*, 11(6), 71-95.

Mills, R., Velasquez, N.F., & Fadel, K. (2012). Examining IS Curriculum Profiles and the IS 2010 Model Curriculum Guidelines in AACSB-Accredited Schools. *Journal of Information System Educators,* 23(4), 417-428.

Patient records giant Epic Systems will take a big step into the cloud in 2015. (n.d.) Retrieved from http://venturebeat.com/2014/11/07/patient-records-giant-epic-systems-will-take-a-big-step-into-the-cloud-in-2015/

Woods, D. & Howard, E. (2015). An IT Strategy Course: Why and How. *Information Systems Education Journal*, 13(4), 4-11.

Appendix 1

Table 1: Expected Skills Depth for strategy related skills in model IS curricula

Skill Name	Skill Keywords (select)	Year Introduced	Depth of Knowledge Expected	
			Initial	Current
Strategic Utilization of IT	Use of IT to support business processes	1973	1	4
IT Planning	Value of IT, end user advocacy	1981	3	4
IT and Organizational Systems	Relationship of business process and IT	1981	2	3

Data from (Longenecker et. al, 2013).

Depth of skill: 1 = recognize, 2 = differentiate, 3 = use (or translate, explain), 4 = apply (without direction or hints)

Information Systems Education: The Case for the Academic Cloud

Lionel Mew
lmew@richmond.edu
School of Professional and Continuing Studies
University of Richmond
Richmond, VA, 23173

Abstract

This paper discusses how cloud computing can be leveraged to add value to academic programs in information systems and other fields by improving financial sustainment models for institutional technology and academic departments, relieving the strain on overworked technology support resources, while adding richness and improving pedagogical delivery of course content. A literature review on cloud definitions and how cloud paradigms are being implemented in academia is conducted. The author suggests that for smaller programs and institutions, cloud hosting of applications, services and platforms in support of information systems programs may be the only financially viable solution to course technology requirements. The impact of transitioning core information systems courses to a cloud paradigm is discussed, and examples of how the transition can improve course content and delivery are provided. Finally, details are presented on how a transition to the cloud is being accomplished in the information systems program of the school of continuing studies at the author's small liberal arts university.

Keywords: Cloud Computing, Academic Computing, Information Systems Education, Emerging Technologies, Cloud Virtual Machine

1. INTRODUCTION

The field of information systems is undergoing a paradigm shift, with cloud computing significantly changing enterprise business processes. This paradigm shift also affects university technology department and information systems program financial sustainment models and pedagogical delivery of course materiel.

The ramifications of cloud computing for higher education institution technology departments are significant. Decreasing levels of financial support from government and increased competition from for-profit institutions, coupled with falling enrollments, have a deleterious effect on the ability of college and university information technology (IT) departments to provide support and services. Even as IT requirements are increasing due to the impact of technology and collaboration on daily life, institutional IT budgets are decreasing. Smaller schools or departments are impacted to an even greater extent, since they do not have the size to garner sufficient support in the face of decreasing institutional resources.

A bright spot in this scenario is the emerging cloud computing technology. This allows smaller programs to develop models with increased capability over legacy hosted servers, at lower cost. If a university lacks resources to host dedicated servers for small departments, cloud computing provides these technologies at the application, platform and infrastructure levels. The cloud provider is able to deliver Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). These services allow departments to reap the benefits of hosted servers, applications and infrastructure, without the cost of local maintenance. The cost is no greater than, and often less than legacy tuition and course fee based school financial models.

This paper reviews literature on how emerging cloud computing technology is being implemented in the academic environment. It discusses how cloud computing may be the only viable option for providing necessary academic support to smaller information systems programs. The work argues the case that regardless of size, a switch to the cloud is likely to enhance delivery of content in information systems core courses. It discusses how cloud computing could potentially enhance courses in other fields. Finally, it details how the author's information systems program in the school of continuing studies at a small liberal arts university is transitioning from locally hosted to cloud virtualized resources.

2. LITERATURE REVIEW

Cloud Computing

There are many differing definitions of cloud computing. For example, Gartner, Inc. defines cloud computing as "a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service to external customers using Internet technologies." However, the National Institute of Standards and Technology (NIST) definition is most cited, and is arguably the most widely accepted definition.

NIST defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." (Mell & Grance, 2011, p.3)

NIST further defines cloud computing in terms of characteristics, and service and deployment models. Service models include Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Deployment models include private, hybrid, community and public models. Private clouds are owned by a single organization. Public clouds provide services to multiple clients using shared infrastructure. Hybrid clouds are often used to leverage the security of the private cloud with the scalability and other advantages provided by the public cloud. Community clouds are those used by a group of users with similar interests, and are thus advantageous to the community.

Cloud Computing in Higher education

Early uses of the cloud in higher education included collaboration, data storage, library science repositories, messaging, flexibility, and computing power.

Mircea and Andreescu (2011) note that the service oriented architecture (SOA) of cloud computing makes it a viable option for academia when financial conditions are poor. They propose a strategy for academic cloud adoption using a stepped migration model. Sultan (2010) suggests that cloud computing may be effective in reducing costs in difficult economic times. Hignite, Katz & Yanosky (2010) discuss opportunities and challenges facing implementation, and suggest that cloud computing leads to a sustainable business model. González-Martínez, Bote-Lorenzo, Gómez-Sánchez, & Cano-Parra (2015) survey 112 articles on cloud computing in education, and note advantages in cost and functionality, tempered by limitations and risks in security, performance and licensing.

Virtualization, simulating a computer in software, allows users to run multiple systems and share resources, in a flexible, scalable environment. Murphy, & McClelland (2008) discuss their experience in developing a high performance virtualized lab, extolling the benefits of virtualization, and demonstrating a scalable solution. Erkoç, & Kert (2010) suggest an infrastructure prototype for a complete cloud distributed campus.

Mohktar, Ali, Al-Serafi and Aborujilah discuss cloud computing in academia, noting the ability to create virtual environments providing the capability to have concurrently managed parallel environments, with an instance for each student. O'Donnell & O'Donnell (2014) provide a detailed discourse on how virtualization enables students to use remotely stored resources transparently.

The literature suggests that cloud paradigms provide improved flexibility, scalability and reliability over local systems, with virtualization making changes in resources transparent to users. These benefits come with significant cost savings.

3. CLOUD AS THE ONLY OPTION

For a small institution or program, moving to the cloud may be the only choice. At the authors' small liberal arts university, the information systems program is part of the school of continuing studies. Bachelor's degrees and post-bachelor's certificates are offered in information systems and information security. Students are local, and most tend to stay in the area. Course enrollments are capped at 15 students, with courses offered annually, and some biennially. As a boutique school, with courses consisting of 15-20 students per course per year, there is no sustainment model which works. Fifteen students in a specific course each year is too few to amortize hardware or software investment.

The Continuing Studies Example

As a school within the university, continuing studies is required to purchase hardware and software specific to school needs. To support project management courses, the school paid to have a public lab outfitted with Microsoft Project, and pays for new version upgrades. The support infrastructure does not support school specific hardware acquisition. The school has access to university wide software licenses, using Oracle running on a university server, as well as Adobe Dreamweaver for web design. The information systems program has a Microsoft DreamSpark subscription allowing students and faculty free licenses to Microsoft software products.

The current level of support is not sufficient to provide students with experience with emerging technologies that will help them on the job market. Developing courses using cloud resources enables instructors to raise the level of training and realism, while lowering cost and providing scalability.

Piloting a Private Cloud

An information systems professor at a local community college designed and implemented a private cloud using a Microsoft grant. The cost to establish the solution was over $100K, not including his time and that of IT support personnel helping to facilitate the project. The project successfully implemented a virtualized private cloud, which was tremendously successful in proving the value of cloud computing. However, even with the grant, this private cloud was not financially sustainable. Although the community college has a larger student population than the authors' institution, it is still small, and several factors contribute to making the system impracticable. First, the annual cost of software licensing is over $10K, which cannot be supported by students taking cloud courses. Second, the IT department is not resourced to support the system. Finally, there is no budget for recapitalization. Still, this pilot project demonstrated how cloud computing can add value to information systems courses.

Hybrid Cloud not the Answer

While a hybrid cloud makes sense for some applications in academia, it does not offer any advantages for delivering information systems courses. Hybrid clouds are appealing to enterprises that want the cost savings, flexibility and scalability that are strong points of public clouds, but are forced by regulatory constraints to protect data by hosting it on a private cloud. Therefore, academic departments dealing with student information, and subject to the Family Educational Rights and Privacy Act (FERPA), may consider use of a hybrid cloud, with data protected under FERPA hosted locally, and other data and applications on the cloud.

FERPA concerns have been a barrier towards full acceptance of cloud computing in meeting all use cases in academia. However, even these concerns are being recognized by cloud and third party services providers. As the cloud paradigm continues to emerge, provisions to accommodate FERPA requirements and other regulatory and business compliance issues continue to evolve.

Due to the limited ability of academic IT departments to control and monitor cloud security, institutions are reliant on controls and compliance by cloud providers. Many institutions lack the resources to evaluate or audit cloud provider resources. Third party providers may have the lead in addressing these issues. One third party provider suggests that they can comply based on their provision of a dedicated disk controller and storage media owned by the institution, and serviced by the third party provider. Emerging paradigms may result in third party providers addressing compliance issues by adopting substantially equivalent alternative standards, implementing alternative compliance schema, or by demonstrating compliance themselves, thus transferring compliance responsibility from institution to provider. The final solution remains unclear, as cloud technology and business cases for academia continue to emerge.

In the case of course delivery, there is no FERPA protected data present, so a hybrid cloud has all the disadvantages of a private cloud, with none of the advantages of a public cloud.

The Public Cloud Paradigm

Using the resources of a public cloud for course delivery offers advantages over locally hosted systems. All of the required resources for software, platform and infrastructure are hosted on the cloud, with cost savings, flexibility, scalability and reliability. For information systems courses, a complete environment can be provided to students, with little consequence for mistakes – individual images can easily be reset by the instructor.

As currently implemented, use of the cloud for course delivery is free for instructors, and grants for student course use are easy to obtain. Even without grants, the cost for the level of service required is trivial. Costs do not mount significantly until systems are deployed.

As cloud computing is still an emerging technology, cloud paradigms continue to change. A risk associated with a switch to a cloud paradigm is the lack of technical support for academic cloud users. As cloud paradigms mature, these risks continue to be mitigated. For example, Microsoft Azure now supports Oracle databases. Using such a database involves provisioning the database in Azure – users are not required to configure a virtual server for the database. Making this portion of the task transparent to users mitigates the need for tech support in server administrator tasks.

In summary, use of the public cloud is efficient from a preparation/user standpoint, is easy for students to use, provides required functionality, and has potential to improve the pedagogical delivery.

4. CLOUD IMPROVES DELIVERY

Cloud hosting makes it economically feasible to attain the required functionality, but also better facilitates course content delivery. Cloud infrastructure, services and applications enable courses to provide unparalleled real-world experiences. This is especially true for information systems courses, where cloud technology capitalizes on the core competencies of leveraging technology to solve business problems.

Security Courses

The cloud's ability to improve course realism is especially true for security courses. For example, target and attack servers may be independently and completely configured within an encapsulated environment, and students are provided with a unique opportunity to use current applications and methods when studying offensive tactics. This experience gives them immediate credibility when seeking employment as penetration testers or enterprise security architects.

Common applications in the penetration tester's toolkit may be used with relative abandon in a contained environment. Penetration testers use an application called John the Ripper to attack local passwords. Similarly, the Cain & Abel tool is commonly used for password recovery. To use either John the Ripper or Cain & Abel proficiently, it is essential that practitioners practice by actually using the application.

The use of cloud computing enables students to obtain first-hand knowledge and experience in use of the penetration tester's tools. This gives students an advantage as they enter the workforce. If they have a job interview, and are asked about their experience in penetration testing, they can reply that they have actually used these and other tools in a real world environment.

If students were to attempt to use these tools outside of the encapsulated cloud environment, the university's IT department would likely be up in arms. Practicing and learning to use these tools on the institutional network may not only be non-compliant with acceptable use policies, but may also be illegal. One option would be to use a closed network for this work, but it would not be feasible to build a lab environment due to the aforementioned financial and support issues.

If dedicated lab resources were somehow made available, the cloud computing paradigm would still be preferable to the physical hardware environment. Envision the confusion and damage which could be caused by a section of students independently attacking intricately configured target computers in the lab. At best, the target computers would have to be reset and re-imaged following an attack event. Using a virtual cloud environment, target servers could be reset anytime, merely by clicking a mouse. This encapsulated environment can be replicated for each student.

The only remote requirement for students to access the virtual environment is internet connectivity. Another advantage of the cloud solution is that students are able to access virtual environments from off-campus home or work computers. In the legacy university hosted paradigm, anyone accessing the server would have had to use a Virtual Private Network (VPN).

Database Courses

There are many other scenarios and courses where cloud computing contributes value beyond that of a remote host for provision of content. During a recent database course, it took some students until the fourth week of class to establish a remote connection to the database on a university server. Students were expected to connect using a thin client on their laptop computers. Other students, particularly those working in the information systems field, were able to immediately connect. There were several obstacles keeping students from connecting. First, the university firewall prevented students from using client software. Second, the connection methodology was technically demanding. Finally, students experiencing difficulty were typically weaker or inexperienced and were technically challenged and daunted by the process. An academically weak student is at a tremendous disadvantage when they are unable to complete the first weeks of coursework. Those who need help the most are a quarter of the semester behind, without accessing the database. Additionally, the process of connecting to the database is not one of the course goals or objectives.

For the same course using the virtualized cloud environment, a database and application to connect to it (SaaS) would be delivered on a virtualized cloud server (PaaS) in a virtual environment without physical hardware (IaaS). The instructor could then develop an image with a server, database and application for each student to interface with the database remotely. The connection between database and application would already be configured, and the image copied into a virtual environment for each student. Students would only have to log into the cloud provider site to access their virtual environments, negating the requirement to use a VPN and other facilitating software. The students are able to use the application to run SQL commands on their virtual database. The ability to access the database via the application is immediate upon logging into the provider system. Therefore, the cloud environment actually improves course delivery by reengineering the process to allow students to focus on activities reinforcing course goals from the outset.

Networking Courses

Cloud Computing is an outstanding vehicle to host networking courses. In the legacy environment, networking was offered using physical hardware. Now, the cloud based networking course uses a virtual environment based on a private cloud, and features a project with students working to develop a notional network architecture. The course begins with the first half of the course concentrating on traditional (Open Systems Interconnection model, architecture and topology, etc.) networking theory.

Following the midterm, students are introduced to network configuration in a virtual cloud environment on Microsoft Azure. Students are quickly able to register, configure their accounts, then define specifications for and implement a Server 2012 R2 server in their virtual environments. Students then further refine their networks, designing architecture, and specifying routers, switches, etc. The ability to configure networks virtually is a huge advantage in convenience and time.

Development and Web Design Courses

Similar to the database courses, instructors for application development and web design courses are able to create virtual images with development tools, connected databases and web servers to publish pages and deploy applications. The outcome of a potential switch to the cloud for Web design courses is still emerging. In the case of the author's institution, the course is taught using Adobe Dreamweaver, for which there is an institutional license. Although Dreamweaver developed sites can be deployed to Azure, Adobe is marketing its software as a service on its proprietary Adobe Creative Cloud platform, which runs on Amazon Web Services. The ability to run Adobe projects on Azure in the future is unknown. The significance to the program is that if other core courses are standardized on Azure, how will the deviant conditions of the web design course be handled? Will the Adobe product continue to be used as it is still considered the industry standard? Will Microsoft Expression Web (currently being integrated into MS Visual Studio) be adopted for standardization purposes? As Visual Studio is currently the standard for database application development courses, it may be easier if Visual Studio was used for both web and application development courses. These issues have not yet resolved themselves, particularly with continued evolution and emergence of cloud provider offerings.

Courses in Other Fields

Although the focus of this paper is on the information systems field, the philosophy, advantages and justifications for using a cloud paradigm apply to other fields as well. The usages range from simple to complex. A professor for a Chaucer class may have a distributed application which takes a Middle English passage and converts it into modern English. Making this application available on the cloud negates the need for local hosting. Students will be able to access the application from off-campus without using a VPN. Any maintenance or troubleshooting can be done by the cloud provider, rather than the overtaxed university technical support staff.

Perhaps a more compelling case would be that of a recent graduate music student, at another institution, working on building her dissertation with a project collecting and indexing instrumental harp music. Her intent is to create a searchable database where instructors can run queries on search terms including such topics as difficulty, category, length, composer, etc., and provide a listing of music. The music instructor could then select and purchase the appropriate music. If this project was developed using a cloud paradigm, it could easily be operationalized into an online database, and monetized with linked e-commerce site to enable a complete cloud solution.

Institutional Issues

In any business process paradigm shift, there are barriers to change. This is no different in implementation of cloud computing. Institutional leaders are often concerned that new implementations will negatively affect regulatory compliance or information security. Administrators and IT staff worry that cloud computing cannot be accommodated by current information security infrastructures. Sometimes these concerns are through ignorance of cloud computing details, but many of them have solid foundations. However, cloud and service providers generally recognize deficiencies and are devising ways to overcome these difficulties as cloud computing continues to emerge.

In any case, interaction with and support from institutional IT departments is critical to successful adoption of cloud resources. As an example, accessing the cloud environment on wireless or wired institution networks requires use of the Remote Desktop port (RDP). The problem is that RDP is typically constrained by the institutional firewall. Most conscientious IT departments prohibit outgoing TCP connections on port 3389, the default RDP port. Most professionals know that this port is the default port, and scans of this port are becoming more common. There are several ways around this. If a static IP address can be obtained for the cloud environment, the IT department may consent to whitelist it. Alternatively, the IT department may make an exception and open port 3389 during class time. Yet another option would be to collaborate with the IT Department to expose port 3389 to port 443, which is typically configured to allow outgoing TCP connections to access secured websites. Due to encryption required to access secured sites, firewall scrutiny is reduced, and RDP may be possible, depending on other variables. The point of this discussion is not to address the details of RDP resolution, but to suggest that there are many ways to provide an environment where cloud computing can work. This example also illustrates that cloud implementation requires close collaboration with the institution IT department.

5. TRANSITION TO THE CLOUD

As discussed in a previous section, the information systems program at the author's institution is currently in the midst of transitioning technology resources for core information systems courses to the cloud. Based on decreasing resources and increasing costs, but faced with the need to ensure that graduates receive sufficient experience to make them viable in the workforce, the program developed a plan to transition to a public cloud model. The benefits of this transition would be twofold; the program would have a cloud-based solution, and students would gain hands-on experience in cloud technologies.

The plan to transition to the cloud paradigm did not arise from a strategic vision for cloud computing from program leaders. When a professor teaching a networking course received a Microsoft grant to allow his students access to the Microsoft Azure cloud for the semester, the grant approval process at the university proved to be so cumbersome that not only was the grant not approved in time to give students access to Azure that semester, it was never approved. That non-event forced information systems program leaders to take a hard look at cloud computing from a holistic view. It was determined that a transition to a cloud based solution for the program would yield tremendous benefits. It was also seen that several continuing students working in the field had effectively doubled their salaries when they acquired cloud competencies. Since information systems is a practitioner's field, it was determined that providing students with competencies in this emerging technology was essential.

The plan to transition from the legacy courses to those supported by the cloud began with a group of three students enrolled in an independent study course during the spring of 2015. This group reviewed literature regarding cloud basics, determined requirements for offering a summer course supported by cloud resources, examined institutional policies and procedures affecting cloud implementation, and recommended a cloud provider for the summer course.

It was decided that the summer networking course would be offered with public cloud support. Although the students in the spring independent study course made a recommendation regarding cloud providers, cost and functionality provided by each provider changed dramatically during semester break, rendering the recommendation invalid. The instructor selected Amazon Web Services for the functionality it provided, along with low cost to students. The week before the hands-on portion of the course was to start, the instructor switched to Microsoft Azure for two reasons. First, it was available on DreamSpark, and second, he was able to secure the grant funding student use.

Following the networking course, a special topics course on virtualization was offered in the fall, 2015 semester. In this course, students worked on projects virtualizing lab environments to support core information systems courses. This initial offering of the course resulted in completion of a cloud database environment to support the traditional database course, complete with Oracle database schema. Teams of students developed solutions based on both Microsoft Azure and Amazon AWS environments. These prototype solutions were developed in collaboration with the instructor and students enrolled in the current database course. The success of this course resulted in the course being changed from a special topics course to a standard offering, listed in the school catalog with permanent course number. A similar course was then adopted by a local community college, and it is expected that the community college course will be accepted in transfer.

Finally, during the spring of 2016, a capstone course is offered which has changed from the traditional applied systems analysis capstone to one designed to enhance cloud competencies from a holistic, managerial view. Consisting of four parts, this course first has students conduct directed reading designed to provide them with basic knowledge on the cloud, followed by a quantitative assessment. Second, students conduct library research on current developments on cloud technology, and write a short paper on the state of the technology. Third, students collaborate on a team project to conduct an analysis of alternatives recommending a cloud solution to support information systems courses. Finally, after presenting their findings, students develop a prototype cloud based system to meet the requirements of a use case examined during the analysis of alternatives.

This sequence of courses provides students with a cloud competency which will stand them in good stead as they enter the workforce. As the cloud paradigm continues to evolve, program managers must continue to evaluate and make changes to ensure that the course sequence remains relevant to industry and student needs.

6. FUTURE RESEARCH

Following completion of the cloud computing course sequence implementing cloud paradigm, program leaders must assess the results from a holistic view, and decide how next to proceed to improve the program, experience and outcomes.

One means of determining how well the project has succeeded is to apply technology acceptance models, which have been well documented and validated in previous research. A recent model is the Unified Theory of Acceptance and Use of Technology (UTAUT), where dependent variables are usage intention and behavior. It is planned that a quantitative study using a survey instrument be used to validate relationships suggested by the UTAUT model.

Another measure of how well the implementation process has succeeded is what students take away from the course sequence. When exposed to the cloud paradigm in the classroom, a quantitative analysis of student self-efficacy in cloud competencies would yield insight into the relationships between implementation and competency assimilation. In addition to a multi-variate regression analysis, Structural Equation Modeling could be used to identify latent variables accounting for some portion of the variance.

A third opportunity to gauge the effects of cloud implementation would be a qualitative analysis of the process. This could be from inductive analysis, ethnographic approach or a case study methodology.

7. SUMMARY AND CONCLUSIONS

Cloud computing is an attractive alternative to traditional, locally hosted technologies for supporting information systems courses. This work suggests that public cloud computing presents a viable model to provide resources necessary for information systems core courses well within the financial sustainment models of smaller institutions or programs.

The literature supports the assertion that cloud computing is financially viable for support of information systems courses, and also yields value due to the flexibility, scalability and reliability of the cloud paradigm. It is suggested that for smaller institutions and programs, the cloud model is currently the only financially viable option.

Despite the lower costs, examples of how cloud computing pedagogical delivery improves content provision in the information systems program areas of information security, database, networking and application/web development depict a case of added value at reduced cost. Examples of courses in other fields are also provided, and demonstrate the value that cloud computing is anticipated to add across the board.

This work discussed a transition to a cloud paradigm at the author's institution that is currently in progress. Although not complete, efforts to date have been successful and on-time, and the expectation is that continued implementation will go smoothly.

In summary:

1. Cloud paradigms are particularly attractive where the student population is too small to support amortization costs of technology on locally hosted systems

2. The cost of private clouds is not sustainable for small institutions and programs where lifecycle costs must be amortized over a small student population.

3. Public cloud paradigms are currently the only sustainable options for smaller institutions and programs.

4. Cloud paradigms improve pedagogical delivery of course content for information systems and other courses.

5. Emerging technologies and provider strategies must be constantly monitored for change.

6. Offering cloud courses at smaller institutions or small programs is contingent on cloud provider support and keeping costs to students affordable.

7. With current levels of provider support, switching a small information systems program to a cloud-based paradigm is achievable and yields many benefits.

8. ACKNOWLEDGMENTS

The author acknowledges the work of Associate Professor Anthony "Dirk" Burress of J. Sargeant Reynolds Community College for having the foresight, vision, motivation and plain hard work to develop and implement a private cloud pilot and transition to a public cloud paradigm for the University of Richmond and J. Sargeant Reynolds information systems programs. Through his efforts, students are also receiving an outstanding education on cloud computing competencies.

9. REFERENCES

Erkoç, M. F., & Kert, S. B. (2011, June). Cloud computing for distributed university campus: A prototype suggestion. In *International Conference on Future of Education, Firenze*.

Hignite, K., Katz, R. N., & Yanosky, R. (2010). Shaping the higher education cloud. *EDUCAUSE White Paper, May*, *21*.

González-Martínez, J. A., Bote-Lorenzo, M. L., Gómez-Sánchez, E., & Cano-Parra, R. (2015). Cloud computing and education: A state-of-the-art survey. *Computers & Education, 80*, 132-151.

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology (NIST), Special Publication 800–145 (January 2011).

Mircea, M., & Andreescu, A. I. (2011). Using cloud computing in higher education: A strategy to improve agility in the current financial crisis. *Communications of the IBIMA*, *2011*, 1-15.

Mokhtar, S. A., Ali, S. H. S., Al-Sharafi, A., & Aborujilah, A. (2013, January). Cloud computing in academic institutions. In *Proceedings of the 7th international Conference on Ubiquitous information Management and Communication* (p. 2). ACM.

Murphy, M. C., & McClelland, M. (2008). Computer Lab to Go: A "Cloud" Computing Implementation. In *Proc ISECON* (Vol. 2008, No. 11, pp. 1-10).

O'Donnell, E., & O'Donnell, L. (2014). A review of virtualization and cloud computing in higher education. In M. Khosrow-Pour (Ed.), Encyclopedia of Information Science and Technology: IGI Global, 10, 7485-7495, doi: 10.401018/978-1-4666-5888-2.ch737

Plummer, D. C., Smith, D., Bittman, T. J., Cearley, D. W., Cappuccio, D. J., Scott, D., Kumar, R. & Robertson, B. (2009). Gartner highlights five attributes of cloud computing. *Gartner Report*, 167182, 1À5.

Sultan, N. (2010). Cloud computing for education: A new dawn? *International Journal of Information Management*, 30(2), 109-116.

Organizing an App Inventor Summer Camp for Middle School Girls: What the Experts Don't Tell You

Nancy L. Martin
nlmartin@siu.edu

Andrey Soares
asoares@siu.edu

School of Information Systems and Applied Technologies
Southern Illinois University
Carbondale, IL, 62901, U.S.A.

Abstract

In this paper, we report on our experience as rookies organizing, funding, and running a summer computing camp for middle school girls. The focus of the camp was building mobile applications using App Inventor. The three day/two night camp targeted girls in rural, high poverty school districts and was funded through an award from the National Center for Women & Information Technology and Microsoft Research. The award allowed girls to attend the camp at no cost. Although a number of institutions and organizations run successful summer camp programs, our experience as first-time organizers should provide other novices with guidance and potential pitfalls for organizing their own camps.

Keywords: summer camps, women in computing, recruiting, mobile application development, App Inventor, experience report

1. INTRODUCTION

The U.S. Bureau of Labor Statistics reports that in 2014, less than 26% of computing jobs were held by women (2014). Although enrollment in computing degree programs has been on the rise in recent years, the number of female students in undergraduate programs is still less than 15 percent and projections of future enrollment levels signal an ongoing struggle to attract women to the computing discipline (Computing Research Association, 2015). It is important to increase the number of women in computing because gender diversity has the potential to increase innovation and improve product and service design, among other benefits (National Center for Women & Information Technology, n.d.-a).

The shortage of women in computing has not gone unnoticed. Organizations such as the National Center for Women & Information Technology and Girls Who Code help bring attention to the dilemma. An abundance of research shows that introducing women to computing before college is a crucial step in the recruiting process. Moreover, research has shown that by high school age, many young women show no interest in computing or consider it "nerdy" or otherwise unattractive (e.g., Gurer & Camp, 2002; Margolis & Fisher, 2002). Therefore, targeting middle school-aged girls with opportunities to learn computing is necessary.

In reality, there is a growing need for more workers, both female and male, in all technology fields. Feeding the pipeline of technology workers is difficult because middle and high schools often do not offer computing courses or do not cover computing in any structured manner. Educators have been trying to fill the void through a variety of after school programs, weekend enrichment programs, and camps for middle school and high school students. K-12 teachers have also been targeted through workshops and other train-the-trainer events. Recent efforts by the nonprofit organization, Code.org, are making headway into pushing coding education into public schools at all levels, but much work is still needed.

Computing, and programming in particular, is perceived as having a relatively high barrier to learning. Computer scientists have attempted to address this barrier through the development of visual block languages such as Alice, Scratch, and others that lower the barrier of initial learning for programming. Another approach to lower the barrier in computing education is through the use of smartphones and mobile application development. App Inventor is a visual block language that is designed for mobile application development on Android devices. Mobile application development is a rapidly exploding field, and some universities now use App Inventor to introduce computer science courses (Abelson, Morelli, Kakavouli, Mustafaraj, & Turbak, 2012; AlHumoud, Al-Khalifa, Al-Razgan, & Alfaries, 2014; Uludag, Karakus, & Turner, 2011). In our own university, faculty have found success using App Inventor both as an introductory tool and in more advanced programming courses (Soares, 2014; Soares & Martin, 2015).

Addressing the lack of females in computing, the authors of this paper organized a summer camp for middle school girls in an effort to initiate an outreach program in the rural setting of southern Illinois. To address the high barrier to learning in computing, mobile application development with App Inventor was the focus of the camp. Following is our experience report which describes organizing, funding, and running a three-day/two-night summer camp for middle school girls focused on Android application development.

2. MOTIVATION

We are faculty in a Bachelor of Science degree program in Information Systems Technologies in the College of Applied Sciences and Arts at Southern Illinois University Carbondale (SIUC). SIUC is the only public four-year university in the southern quarter of the state. It is located in a rural area with the nearest major city more than 100 miles away. Many counties in this region are considered low income and suffer from higher than average poverty rates and lower high school graduation rates (Mid-America Institute on Poverty, 2003). Many students from small school districts in the area do not attend college, and many have never visited a college campus. This lack of access, especially for young women, was the major impetus for organizing the camp.

Based on experience teaching Android application development in our curriculum, we believed mobile application development with App Inventor was a perfect mechanism as a low barrier and fun way to engage girls with the computing discipline. Others have reported positive experiences using App Inventor for beginners and for younger students, and in a camp environment (Roy, 2012; Urness & Manley, 2013; Wagner, Gray, Corley, & Wolber, 2013).

3. PLANNING THE CAMP

Since our primary motivation for creating the camp was to provide exposure to computing for a group of girls from a rural, low income, high poverty area of the state, costs for the girls to attend would need to be kept to a minimum, if not free.

Initial Planning and Feasibility

SIUC's Continuing Education (CE) department is the central campus source for all community activities offered such as camps and conferences. Our first step was to meet with CE staff to understand what all was required to offer a summer camp.

The CE staff provided examples of other camp brochures, schedules, activities, legal requirements, etc. to help us design an appropriate combination of learning and fun for a middle school-age group. The CE staff was also instrumental in our understanding of the administrative requirements of running a camp. Numerous tasks were necessary; for example, liability waiver forms and photograph release forms had to be prepared for signature by parents or guardians; a process for collecting camper registration information was needed; a means to collect and subsequently refund deposit money had to be established, and many others administrative tasks.

In this stage, in addition to understanding the process, our focus was on preparing a preliminary budget to determine if a low or no cost camp was feasible. It became clear early on that some sort of outside funding would be necessary if we were to offer this camp cost-free. We identified a potential funding source through the National Center for Women & Information Technology (NCWIT) (www.ncwit.org).

NCWIT and the Academic Alliance

NCWIT is a nonprofit organization that works with industry partners, universities, government and other nonprofit organizations to increase the participation of women in computing and technology. The NCWIT Academic Alliance (AA) is one arm of the organization and is focused specifically on "implementing institutional change in higher education" (National Center for Women & Information Technology, n.d.-b) with regard to gender diversity in technology and computing. NCWIT AA provides numerous resources, research, and best practices for recruiting and mentoring women into the technology fields. SIUC is a member of the NCWIT Academic Alliance.

The NCWIT Academic Alliance Seed Fund provides NCWIT's AA members at non-profit, U.S. institutions with start-up funds (up to $10,000 per project) to develop and implement initiatives for recruiting or retaining women in computing and technology. The Seed Fund was initiated in 2007 and is funded by Microsoft Research. As of 2015, 43 AA member organizations have received more than $500,000 to initiate, support, or grow technology-related outreach programs. Funds are awarded each year through a competitive proposal process (www.ncwit.org/programs-campaigns/ncwit-awards/ncwit-aa-seed-fund). Proposals are normally due around November 1st and awards are announced the following February.

In the fall of 2012, with a potential funding source identified, we began detailed preparation of a budget and proposal to initiate a summer camp program.

Preparing the Budget

We again turned to SIUC's CE staff for their expertise in budget preparation, planning for 20 campers. The proposed budget included categories for equipment, personnel, housing, meals, activity fees, recruitment materials, t-shirts for campers, and CE fees for handling administrative processes.

Equipment. As we were planning for 20 campers, only 5 phones needed to be purchased because our department already owned 15 phones for an Android Application Development course offered in the curriculum. An allowance was also made for textbooks purchased for each camper.

Personnel. The target age group of girls was middle school, or 6th through 8th grades. Because this would be an overnight (two nights) camp, supervision outside the daily class scheduled hours was required. We budgeted for two female graduate assistants to serve as Resident Assistants (RA). The RAs would supervise the campers from the end of camp each day until they escorted the girls back to the lab the following morning. We learned that background checks were required for anyone serving as an RA, so those fees were also added to the budget. Additionally, to support up to 20 girls, lab assistants would be needed. We budgeted for two lab assistants for three days. Because these personnel would not be on normal graduate assistant contracts, we were allowed to offer an hourly rate of pay. The authors of this paper served as camp director and camp instructor, respectively. As camp faculty, we did not pay ourselves. Still, personnel was the most costly component of the budget.

Housing and Meals. During the summer, campus dining halls are not in operation, so we had to plan all the meals and snacks for the three day/two night camp, and arrange to have them supplied at appropriate places and times. Additionally, we provided breakfast, snacks, and drinks during the course of the class day. On the last day of camp, light refreshments were provided for parents/guardians who came to see camp projects as they picked up their girls. Meals and food for the campers and camp personnel was the second largest category of expenses.

Housing the campers and RAs for two nights was the third most costly item in the budget. Campus dormitories were utilized with two campers to a room and a room for the RAs. These were charged at a daily rate by University Housing along with a nominal daily charge for sheets. Campers would need to provide their own blankets and towels. Since we were targeting a group of students specifically identified as in need of financial aid, we allowed a small amount of money in the budget in the event that we needed to purchase towels or blankets for some of the campers.

Activities. One of the motivations for offering the camp was to expose girls from the rural region to the college environment and the University campus. Therefore, we planned evening activities at SIUC's Recreation Center. Campers and the supervising RAs had access to various court and exercise activities and swimming. We budgeted the standard daily rate for the attendance fee for the campers. To allow use of the pool, we also had to budget for lifeguards to be on duty during our visit.

Our classroom building is located just across a courtyard from SIUC's Campus Lake. For the last day of camp, we planned for a cookout for the campers and camp personnel (included in the meal budget) at the lakeside pavilion. We also planned for the girls to have access to paddle boats during the extended lunch period. Although the use of the equipment was free, we again had to budget for lifeguards to be on duty while the campers were using the lake.

Recruitment. Based on the number of schools we would contact, a relevant amount was budgeted for printing camp brochures and initial mailings. We chose to do a print mailing because the University had address information for the targeted schools, but not all of schools had email contacts available. A small amount was also budgeted for follow up mailings to each camper's school to acknowledge participation.

Miscellaneous. We also budgeted to provide a camp T-shirt for all campers and camp staff. Additionally, a flat fee was charged by CE for handling the registration process. CE managed all interactions with regard to camper registration and payment of deposits, all coordination of background checks and housing reservations, and check-in/out at the dormitories.

With a complete budget prepared, a full proposal was written and submitted to NCWIT in November 2012. The following February, we received notice we had won one of the awards for 2013. Our full proposed budget was funded and we immediately began preparing the recruitment material.

4. ORGANIZING THE CAMP

The camp was planned for the first week of June, 2013. This date was chosen based on the availability of camp personnel and to coordinate with other camps being offered the same summer. From the award announcement to the first day of camp allowed just under four months to recruit campers, personnel, and organize the camp.

Personnel Recruitment. Potential RAs and lab assistants had already been identified and had verbally committed to working during the camp prior to the award announcement.

Camper Recruitment. Because we were providing the camp completely cost-free for up to 20 girls, we wanted a process to insure that truly deserving girls would be selected. Based on CE's experience with similar recruitments, we established a nomination process that began by mailing a cover letter, along with a camp brochure and nomination packet, to administrators of all middle schools in the southern quarter of the state. The letter explained the camp, the purpose, and asked the administrators to nominate one female student from their school who best met the following criteria:

1. completed 6th, 7th, or 8th grade as of May/June 2013
2. previously demonstrated an interest in computers, technology, math, science, or engineering
3. demonstrated maturity level commensurate with attending an overnight camp
4. demonstrated financial need
5. able to secure transportation to/from Carbondale (no travel funds were provided)
6. not related to the nominator (son, daughter, niece, nephew, etc.)

Administrators could complete the nomination process via a website created specifically for the camp. They were to upload a completed nomination form with information about the student, including contact information for a parent or guardian, and a signed letter of recommendation on school letterhead. The brochure could be given to the student. Once a nomination was submitted, a registration form and camp information were mailed to the student's home address by CE. If an email address was given, we also sent the material via email to the student and parent or guardian.

We did charge a $50 registration deposit per student that would be refunded upon camp completion. This deposit was recommended by CE staff to prevent students from registering for the camp and subsequently not attending, thereby precluding another student the opportunity. In some instances, a school administrator or teacher paid the deposit for a student.

We had anticipated mailing the packets by mid-March, allowing schools about one month to complete the nomination process, and potential campers would be contacted by late April. However, due to numerous unforeseen events, that timeline was extended to the point that we lost potential campers. We did not anticipate the "red tape" involved in receiving such an award and subsequently being able to use the funds. Even though the funds were considered an "award" versus a grant, they still had to be processed by the University's grant office. Additionally, once the funds were available, control transferred to Accounting Services to administer the award account. We continued preparing for the camp while waiting for approval to spend money on the recruitment mailing. By the end of March, we still did not have University approval or access to the award funds.

Concerned that we were running out of time before the end of the school year, we learned that we could ask our College to set up an Advance Account. That process was completed and the recruitment packets finally were approved to mail in early April. However, another stumbling block had occurred. A period of several days expired between when we approved the mailings and when they were physically mailed, again slowing the recruitment process. In some cases, schools did not receive the material until early May. These events culminated to severely impact our recruitment process since many schools dismiss for the summer around mid-May and many families already had other events planned for the summer. As a result, only nine campers were nominated and registered.

Lab and Equipment Setup

To run this camp, we needed phones, a wireless router, and computers for the campers and instructor. The computers needed Internet access and a browser to connect with App Inventor. We utilized a lab classroom outfitted with desktop PCs running Windows 7 and used Google Chrome for the browser.

Accounts. To be able to use App Inventor, campers needed a Gmail account. We discovered that most of the girls registered did not have a Gmail account, and some did not meet the minimum age required by Google to open an account. For these campers, we created temporary Gmail accounts that were monitored by the lab assistant. Campers also needed login access for the SIUC lab computers, and these were requested as temporary accounts through the University's Office of Information Technology.

Phones. The girls were not required to have an Android phone to attend the camp because we had several phones previously acquired with a State Farm Technology Grant. The phones were controlled by the instructor who distributed them before class and collected them at the end of the day. For camp or class use, the phones need not be activated for cell service, but should have wireless capability. All phones were configured by the instructor to access a wireless router available during class.

Instructional materials. At the time of the camp, the primary textbook in use was App Inventor: Create Your Own Android Apps (Wolber, Abelson, Spertus, & Looney, 2011). We were able to purchase books at a discount from the publisher. Additionally, we provided copies of some labs that were not part of the book. The girls were allowed to keep the book in hopes they would remain interested in mobile application development.

Figure 1: Sample pictures captured and modified with the PaintPot application

As an example of class work, campers created the PaintPot application which uses the camera component to take pictures and allows users to draw on top of the picture. Campers were allowed to take their phones to various camp activities and to register their environment. Figure 1 displays a few campers' pictures that were captured and modified with the PaintPot application. This was a great opportunity to allow campers to utilize their application with activities inside or outside of class, as well as to cultivate new ideas based on their learning experiences and environment. Many campers took the opportunity to tease camp staff with their drawings which we saw as a positive ice breaker with the campers. The instructor, in particular, was an easy target as he was the only male involved in the camp.

Daily Schedule

Each day allowed several hours of class time, but other activities were also planned for the campers. During extended lunch periods, the girls could interact with camp personnel, other campers, and invited guests. We planned for lunch periods on the first two days to include visits from women in computing professions, and other educational activities such as videos or demonstrations of coding and other websites.

On the first day of camp, the girls arrived at the dormitories for check-in. Although CE handled the check-in process, most camp personnel, including faculty, RAs, and lab assistant were present to meet the girls and their parents or guardians. The schedule allowed for about six hours of class time the first day of camp. Some of this time was spent introducing mobile application development, and acclimating campers to the process of building and running applications with App Inventor. By the afternoon, the campers were completing labs with the assistance of the camp staff.

The second day allowed for a total eight hours of class time. The morning session involved demonstrations of more features and subsequent practice by the campers. In the afternoon session, the campers were instructed to begin planning and building their own application or to customize an application from a tutorial provided by the instructor.

The final day of camp allowed about four hours to complete individual projects. After the lunch cookout, parents or guardians arrived in the classroom where the girls could showcase their accomplishments from the camp.

At the end of each day of the camp, the RAs would meet the campers in the classroom to escort them back to the dormitories and prepare for evening activities. When the RAs arrived, all camp personnel would briefly discuss what went well and what needed improvement for the next day or next camp.

5. RUNNING THE CAMP: PLANS VERSUS REALITY

The popular idiom "the best laid plans of mice and men oft(en) go astray" certainly rang true in our experience with this camp. It is through these learning experiences, we hope to help others in their initial attempts to organize and run a summer camp.

Recruitment

We had tremendous assistance from the CE staff in planning the camp, but they are not involved in the monetary side of award funding so were unable to warn us of the time required to process the award. Given our inexperience and the events described in Section 4 that impacted the timeline, our recruitment process suffered. Although we had budgeted and planned for up to 20 girls, only nine actually attended the camp. We would strongly recommend planning the recruitment procedures early in the process. We were not able to proceed with the printing and mailing processes until we had access to the award funds.

Personnel

Potential camp personnel were identified prior to the award announcement. The RAs required background checks, and as graduate students, their hiring process took several weeks. As a result we already had signed agreements with the two RAs before we knew of our low camp enrollment. Even though it was a lower camper to RA ratio than we planned, it was expense well spent because nine middle school-age girls proved to be a handful to supervise. For the lab assistant, we hired an undergraduate female student who was trained on Android development with App Inventor. The hiring process for undergraduate student workers is much simpler than for graduate students, therefore we were able to reduce the number of lab assistants originally planned to only one. The one lab assistant and two faculty were sufficient classroom support for the nine campers.

Schedule and Activities

While we anticipated that it was unreasonable to expect middle school-aged girls to work in long sessions without getting bored or frustrated, we quickly learned that we had underestimated the energy and interest levels of middle school girls. Additionally, both our planned speakers were unavailable by the time of the actual camp. Therefore, on the first day of camp, we immediately arranged for more diverse extracurricular activities during lunch periods.

Our College recruiter came to the rescue. She arranged a number of games for the girls to play in teams including a scavenger hunt around campus. These activities helped the girls acclimate to each other and to the camp staff, and get to see a bit of campus. This was a no-cost activity.

During the lunch period on the second day, campers enjoyed bowling at the SIU Student Center. Our cost was shoe rental and game fees. Although this amount was not planned in the original budget, we had saved a considerable amount of money by only housing nine campers instead of 20. On the third day, the cookout and lake activities were held as planned.

The original plan when the budget was created was for the campers to visit the SIUC Recreation Center both evenings of the camp. However, between the proposal submission and award dates, and actual camp planning, the Recreation Center became unavailable for the second night. Instead, we scheduled SIUC's Craft Shop to host a session at the dormitory. Craft Shop staff brought supplies for the girls to each decorate a pair of flip flops they would keep. The cost was a flat fee per crafter, and was about equal to the expense we would have incurred at the Recreation Center.

Camp Evaluation

As part of the award proposal, we planned a number of means to evaluate the success of the camp. From the camper perspective, we utilized ICE@Georgia Tech Surveys (Institue for Computing Education @ Georgia Tech, n.d.), available at http://coweb.cc.gatech.edu/ice-gt/1115, to measure interest in computing both pre- and post-camp. Likely due to the small sample size, there were no significant differences in responses to statements such as *computers are fun, computer programming is hard,* or *I am good at computing.* However, in responding to statements such *I liked this camp; I had fun at this camp; this camp made me interested in computing;* and others, all campers either agreed or strongly agreed indicating a successful endeavor. A few weeks after the camp ended, we received an email from a parent saying that her daughter was interested in taking more computing courses as a result of the camp.

From the camp staff perception, although we hit many bumps along the way, all agreed the camp was a success. Camp staff and others were quick to change direction or add additional activities as needed. From our faculty perspective, the purpose of the camp was accomplished. A group of girls from rural, high poverty school districts was able to attend a computing camp at no cost.

6. LESSONS LEARNED

1. Budget. There were many line items in the budget that we had not anticipated. For example, for campers to stay in University housing, we were charged a “risk management fee” per camper. Although it was a nominal daily fee, it was just one of several expenses we had not considered in early planning. Our advice is for faculty to work with their departments that coordinate camps early in the idea-forming stage and ask to see actual budgets from other camps.

2. Timeline. Allow twice as much time as you think is needed for planning and initial organizing. As novices, we now realize that our four-month period was not near enough to effectively handle the roadblocks we encountered. Also talk to other faculty on your campus who have operated camps, specifically asking questions about difficulties and roadblocks they have experienced. Although we did talk to other faculty on our campus, their camps had been running so long that they had all but forgotten all the stumbling blocks faced in the early years.

3. Resources. If seeking an outside funding source, contact your grant and accounting offices prior to proposal submissions. Understand the mechanics of receiving and spending money from outside sources. Ask specifically about lead times; then add several weeks or months to your timeline.

4. Recruitment. Identify target groups or schools and contact them early to alert them to your upcoming camp and formal recruitment materials. Have those recruitment materials ready to distribute as early as possible to insure school administrators and teachers have time to identify campers. If operating a camp that will charge fees, allow sufficient for time schools to distribute camp information to students and parents or guardians and encourage early registration. Make use of your university’s contact databases. If using postal mail, contact those that will be printing and mailing your material and ask specifically about lead and lag times. We chose to use postal mail, but in the future, we will likely rely on electronic communication for the distribution process.

5. Campers. Understand your target campers. Three days and two nights is a long time to entertain middle school girls. In the future, we will better plan for a significant activity during the lunch breaks and add more short breaks. These additional break times can be tied to the class with activities such as sending the campers out to use the GPS feature on the phones. We were fortunate to have the assistance of our College recruiter to organize the scavenger hunt and other games. Seek similar resources to have on stand-by in the event last minute activities are needed.

6. Instruction. The girls picked up on the material much more quickly than we expected. Therefore, we had to swiftly copy additional lab materials and prepare additional lessons on the fly. In the future, we will over-plan, i.e. prepare double what we think might be covered during the camp.

7. SUSTAINABILITY AND FUTURE PLANS

Our goal with the NCWIT AA proposal was to initiate annual or biennial summer camps in an effort to attract middle school girls to computing. In particular, because of our geographical location, we want to target girls that might not otherwise have an opportunity to attend such a camp. Due to the small number of campers recruited, we conducted the first camp with some money left over. This allows us recruit future campers to attend camp at no cost.

Additionally, our sustainability plan includes a number of other components. First, we will seek industry partners that we currently work with to provide donations of money, food, and other camp items. Second, we will allow a portion of camper slots to be paid ones. For example, in a camp for 20 girls, we could recruit 10 for no cost, and allow 10 other girls to attend for a fee. We know there is a demand for computing camps in our area because since our initial camp material was distributed, we have been contacted by several parents offering to pay for camps, and also requesting computing camp for boys.

We plan to expand the type of offerings we provide in an effort to enhance the recruitment of all students, but especially females, to computing. We have participated in our campus Expanding Your Horizons (www.eyhn.org) conference by offering short workshops using App Inventor. We are also planning single-day camps for middle school girls in spring semesters. A one-day venue would cost much less to host, and would perhaps be more attractive to some families who do not want their children away at camps in the summer. We would also like to expand the day camp offerings to include high school girls.

Also, because of the relative simplicity of the equipment required, we envision a camp where instead of students coming to the University, instructors can travel to schools to offer short versions (i.e., half or full day) of the camp using the schools' equipment and our phones. Each state varies in its rules regarding faculty visits to public schools, so others should research this topic relative to their state prior to visiting schools.

Overall, we believe we conducted a successful first summer camp despite our inexperience. Although we encountered various roadblocks, they were met with patience, ingenuity, and the help of many others. While other universities and organizations have established successful long-term summer camp programs (e.g., Ericson & McKlin, 2012), we hope that our experience as summer camp rookies will be helpful to others considering similar endeavors.

8. REFERENCES

Abelson, H., Morelli, R., Kakavouli, S., Mustafaraj, E., & Turbak, F. (2012). Teaching CS0 with mobile apps using App Inventor for Android. *Journal of Computing Sciences in Colleges, 27*(6), 16-18.

AlHumoud, S., Al-Khalifa, H. S., Al-Razgan, M., & Alfaries, A. (2014). Using App Inventor and LEGO mindstorm NXT in a summer camp to attract high school girls to computing fields. *2014 IEEE Global Engineering Education Conference (EDUCON)*, 173.

Computing Research Association. (2015). 2014 Taulbee Survey. *Computing Research News, 27*(5).

Ericson, B., & McKlin, T. (2012). *Effective and and sustainable computing summer camps.* Procceedings of the 43rd SIGCSE Technical Symposium on Computer Science Education, Raleigh, North Carolina.

Gurer, D., & Camp, T. (2002). An ACM-W literature review on women in computing. *SIGCSE Bulletin, 34*(2), 121-127.

Institue for Computing Education @ Georgia Tech. (n.d.). *Pre and post attitude surveys for computing summer camps and workshops*. Retrieved from http://coweb.cc.gatech.edu/ice-gt/1115

Margolis, J., & Fisher, A. (2002). *Unlocking the Clubhouse: Women in Computing*. Cambridge, MA: MIT Press.

Mid-America Institute on Poverty. (2003). Atlas of Illinois Poverty *Illinois Poverty Summit*.

National Center for Women & Information Technology. (n.d.-a). About Us. Retrieved May 15, 2015, from https://www.ncwit.org/about

National Center for Women & Information Technology. (n.d.-b). Academic Alliance. Retrieved March 12, 2015, from https://www.ncwit.org/alliances/aa

Roy, K. (2012). *App inventor for Android: Report from a summer camp*. Proceedings of the 43rd SIGCSE Technical Symposium on Computer Science Education, Raleigh, North Carolina.

Soares, A. (2014). Reflections on teaching App Inventor for non-beginner programmers: Issues, challenges and opportunities. *Information Systems Education Journal, 12*(4), 56-65.

Soares, A., & Martin, N. (2015). Teaching non-beginner programmers with App Inventor: Survey results and implications. *Information Systems Education Journal, 13*(5), 24-36.

U.S. Bureau of Labor Statistics. (2014). Labor Force Statistics from the Current Population Survey. Retrieved March 5, 2015, from http://www.bls.gov/cps/cpsaat11.htm

Uludag, S., Karakus, M., & Turner, S. W. (2011). *Implementing IT0/CS0 with Scratch, App Inventor for Android, and Lego Mindstorms*. Proceedings of the 2011 Conference on Information Technology Education, West Point, New York.

Urness, T., & Manley, E. D. (2013). Generating interest in computer science through middle-school Android summer camps. *Journal of Computing Sciences in Colleges, 28*(5), 211-217.

Wagner, A., Gray, J., Corley, J., & Wolber, D. (2013). *Using App Inventor in a K-12 summer camp*. Proceeding of the 44th SIGCSE Technical Symposium on Computer Science Education, Denver, Colorado.

Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2011). *App Inventor*. Sebastopol, CA: O'Reilly Media.

ISSN: 1545-679X September 2016

APPENDIX A
DAILY SCHEDULE

SIU CARBONDALE SOUTHERN ILLINOIS UNIVERSITY
SCHOOL OF INFORMATION SYSTEMS AND APPLIED TECHNOLOGIES

BUILD YOUR OWN ANDROID APP

SUMMER CAMP FOR MIDDLE SCHOOL GIRLS

Daily Schedule

Wednesday

Time	Activity	Location
8 am – 9 am	Registration	Schneider Hall
10 am – 12 pm	Class Activities - Intro to mobile apps, demo app - Set up phone, computers, & Gmail accounts for camp use - Build first app	ASA 204B
12 pm – 1:30 pm	Lunch Games & Campus Scavenger Hunt	ASA 111
1:30 pm – 5:30 pm (Break 3:30-3:45)	Class Activities & Break - Build apps	ASA 204B
5:30 pm – 6 pm	Return to dorms	Schneider Hall
6 pm – 7:30 pm	Rec Center Activities Swimming (lifeguards provided)	SIU Rec Center
7:30 pm – 10 pm	Dinner at dorm Dorm activities	Schneider Hall
10 pm	Lights out	Schneider Hall

Thursday

Time	Activity	Location
7:30 am – 8 am	Breakfast	ASA 111
8 am – 12 pm (Break 10-10:15)	Class Activities & Break - Build apps	ASA 204B
12 pm – 2 pm	Lunch Bowling at SIU Student Center	ASA 111
2 pm – 6 pm (Break 4-4:15)	Class Activities & Break - Build apps	ASA 204B
6 pm – 7 pm	Dinner at dorm	Schneider Hall
7 pm – 8:30 pm	Decorate flip flops with SIU Craft Shop personnel	Schneider Hall
8:30 – 10 pm	Dorm activities	Schneider Hall
10 pm	Lights out	Schneider Hall

Friday

Time	Activity	Location
7 am – 7:30 am	Pack up & store belongings for check out	Schneider Hall
7:30 am – 8 am	Breakfast	ASA 111
8 am – 12 pm (Break 10-10:15)	Class Activities & Break - Work on app for showcase	ASA 204B
12 pm – 2 pm	Lunch & activities at Campus Lake - Cookout & Paddle boats (lifeguards provided)	Behind ASA Bldg
2 pm – 4 pm **Parents should arrive at 2 pm**	Camp Showcase - Student presentations of their apps	ASA 204B
4 pm – 4:30 pm	Check out of dorms	Schneider Hall

APPENDIX B
SAMPLE TIMELINE

	Month								
Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
FEASIBILITY										
Contact CE office										
Contact experienced faculty										
Contact grant office										
Contract accounting office										
Identify funding sources										
Determine camp schedule & size										
Preliminary budget										
FUNDING										
If outside funding, prepare proposals										
If charging, determine camper fees										
Get approval to spend funds										
If charging, receive registration funds										
RECRUITMENT										
Identify target population										
Prepare contact materials										
Build/prep for web submission										
Determine means of contact										
Acquire contacts										
Consult with campus printing/postal service										
Distribute recruitment materials										
Register/contact campers										
CAMP PLANNING										
Plain daily schedule										
Recruit personnel										
Reserve housing										
Reserve lab space										
Prepare/purchase lab equipment										
Prepare/purchase lab materials										
Plan extra-curricular activities										
Plan for food										
Prepare camp evaluation processes										
Conduct camp										