ABSTRACT
This paper reports the results of an ongoing pedagogical process and evaluations being conducted to determine if requiring students to code and get apps accepted into the Windows 8 Store helps develop innovative thinkers. Student participants are primarily non-traditional and none have any prior coding experience.

Keywords: Innovation, Windows 8, student development, experiential learning, programming,

INTRODUCTION
More frequently than not strategists in today’s market believe that the creative thinking process of designers stimulates out-of-the-box thinking and in so doing contributes to innovation and competitiveness of an organization. As academicians, we all strive to find ways to encourage our students to become innovative thinkers who are highly successful once they enter the workforce. We try traditional software development projects (Mills, Hauser, & Pratt, 2008; Mitri, 2008), service learning projects (Wei, Siow, & Burley, 2007), and competitions like Microsoft Imagine Cup (Parrish Jr., Bailey, & Jensen, 2010; Parrish, Bailey, & Jensen, 2009). The greatest difficulty encountered with each of these approaches, while having distinct strengths is that projects by definition are larger, take longer, and require group collaboration which does not always provide the best environment for development of individual innovative thinking skills.

PURPOSE OF THE STUDY
Strategists increasingly are espousing the influence of design thinking on the ability of a firm to be innovative. The design thinking technique starts with developing a thorough understanding of customer needs and leveraging employees’ creative ideas to meet those needs (Wattanasupachoke, 2012). The field of IT has been gathering user requirements and developing systems to meet those needs for decades. As pointed out by Denning (2013), design thinking is already deeply embedded into agile software development while the traditional view of computing design focuses on creating artifacts that are error-free. If the two are blended together, it will surely create a very powerful development environment (Denning, 2013).

However, the difficulty in teaching students how to do this is to find an environment in which they can code, within a single semester, applications that can and will be used by real people. The impetus behind this study was to provide the opportunity to create, implement, get feedback, and make corrections to multiple applications and observe the results.
GETTING STUDENTS ON BOARD

The study was started in an introductory C# course where none of the students had any prior coding experience. Traditionally MIS students dread computer programming classes. For some programming is so intimidating due to lack of confidence and anxiety that retention of computing students can be adversely affected (Connolly, Murphy, & Moore, 2009). Other students object to programming classes because they view programming as “boring” or believe “programming jobs are being outsourced” so learning programming is a waste of time (Ben-Ari, 2011). To broaden participation educational environments have worked to make programming more exciting by incorporating game design and simulation building through drag and drop interfaces (Repenning, 2012). However gaming is better suited for computer science students than it is for MIS students.

The course started out in a traditional manner with the first half of the semester being Windows forms development. However, on the first night of class, the students were informed that by the end of the semester they would be required to have two apps accepted into the Windows 8 Store. Panic ensued. To calm the mass hysteria, the following offerings were made:

- Students were offered the chance to have the Windows 8 exam waived for an additional two published apps
- Ten bonus points were offered for each published app over the first four
- Students were asked to think about what would make more of an impact when potential employers wanted to know what they had done, the statement, “I did assignments in class” or “would you like to see what I have in the Windows 8 Store?”

Halfway through the semester and at the end of the traditional Windows forms coding assignments, class performance was on par for the average class with nearly half the class having a D or less in the course.

MICROSOFT SUPPORT

Microsoft is highly supportive of academic initiatives and this study could not have been accomplished without their assistance. The corporation provides free developer accounts for the students through Dreamspark and free developer tools through Dreamspark Premium. They sent a Microsoft Relationship Manager to hold a weekend workshop to kick-off the Windows 8 app initiative being implemented and Microsoft was offering a $100 incentive for each app a student got accepted into the Store up to ten apps. Students were not required to attend the workshop but most did.

Microsoft provided a template for students to start with that they could customize. Students were required to come up with an idea for an application they believed would be valuable to someone, modify the template inside of Visual Studio with facts on the topic, RSS feeds, youtube links, flicker links, localized weather, and other optional components.

TEACHING STYLE

While the students were not coding from scratch, the theory was that by doing multiple apps which they could complete in a relatively short period of time compared to coding from scratch, they would have a sense of accomplishment and could focus on learning how to navigate in Visual Studio as well as concentrating on what should go into the app. This was quite successful because many of the RSS feeds would cause the app to crash and students would have to debug and replace the feed in the code. From a coding standpoint, they became comfortable with strings, objects, and properties as well as proficient at navigating C#, XAML,
Solution Explorer, and the various aspects of Visual Studio 2012. They also were able to complete numerous applications, receive user feedback in the Microsoft Store, and submit updates. As a side benefit, they learned how to modify code written by another party, which is the most common form of coding in industry.

Sixteen of the 20 students in the class attended the workshop and came both days despite being told they only needed to come one. The workshop was scheduled to end at 5 pm each day and at 11 pm each night students reluctantly agreed to go home. Seeing how successfully the students were learning and working in the workshop environment, the structure of the introductory class was rearranged to make the remainder of the course operated in an open workshop environment.

No lectures were held. Instead the professor worked individually with students on individual problems. Students spoke freely with each other about problems they were encountering and what they’d done to correct them. Since everyone was working on a different topic so their app would get accepted into the Store, there were no duplicate assignments turned in, no cheating that occurred. Students freely learned from and benefited from each other. They also studied each other’s successes and improved their own skills in the process. An interesting observation from the classroom environment was that the students exchanged cell phone numbers and texted each other to discuss issues they were having with their applications.

The submission process to the Windows 8 Store has a major benefit in that it requires the student to write a description that makes a user want to download the app. The student has to create screenshots that are appealing. They have to make decisions as to the age group the app is appropriate for, what countries the app should be published in, pricing, etc. If any of these decisions are not appropriate the Microsoft testers will reject the app and the student must correct and resubmit it. Testers will also, obviously, reject an app if it crashes or fails to perform as it should but the communication component is an aspect that is hard to duplicate elsewhere.

A second benefit to the Store is that there is an external validation above and beyond the faculty member that the app is accurate and viable. It is much easier for a student to argue with their professor than a Microsoft tester.

A third benefit is that students have the opportunity to learn that apps that are appropriate in one country may not be viewed as appropriate in another thus assisting with understanding of global issues to some degree. For example, one student learned that a “Walking Dead” app would not be accepted until certain countries were removed as places he wanted the app published to.

A fourth benefit is that students can put a price tag on their app and the more innovative and appealing their app is, the more likely they are to profit from it. Several students have made over $10K off of a single app that they created in their introductory class.

THE RESULTS

The results of this pedagogical experiment can only be described as phenomenal. Once the students started seeing their apps being downloaded by people they didn’t know in countries around the world and they started getting five start ratings, the excitement was unlike anything this researcher has ever witnessed in a classroom before. That excitement culminated in the following findings from the first semester:

- The 20 students in the class had an average of six accepted apps in the Marketplace in a 3 week period instead of the required two applications
Exam scores soared. Students who had been soundly failing the hands-on exams made A’s or B’s on their final two tests.

Student excitement was been so high that at any given time, as many as four to six students can be found voluntarily working together outside of class time on their apps.

Comments such as “who knew programming could be so much fun??”, “when can I take my next programming course?”, and “are we going to have any Windows 8 development workshops this summer?” are common.

Sixteen of the twenty voluntarily spent 24 hours one weekend at a Windows 8 App Development workshop that was only scheduled for a total of 15 hours because it took that long to get them to go home.

Students NOT in the programming class and who have not yet taken a programming class are now getting apps accepted into the Marketplace because their friends in the class taught them how.

Twelve of the twenty volunteered to develop two-hundred and fifty apps during their summer vacation to raise money for their college department after one of our industry partners asked them to do so. These twelve students were not satisfied with only getting 250 apps into the Store. They published 500 apps in Windows 8 and 250 in Windows Phone 8 instead.

Eight of the twenty students came to the first night of the introductory programming class in the fall semester to share how exciting programming was.

In semester two a new and significantly more complex template was introduced which allowed for working with SQL Azure, XML storage, incorporating social media, and numerous other opportunities. Students from the Spring semester immediately started demonstrating their innovative thinking by adding capabilities to the template which could then be used by students across the country. One example of this is a search feature that was added to the template that allows the user of the app to search for any word and the app will locate and show only the tiles that contain that word regardless of where the word is found.

Additional observations from the second semester were as follows:

- By the end of the fall semester a group of about 20 students, some were the original 12, had collectively made approximately $55,000 in sales and Microsoft incentives from the apps they have published in the Store.
- Students have created apps that have upwards of 200,000+ downloads.
- Two students have been awarded Microsoft Bizspark grants worth $500,000 in software and services over a three year period of time.
- One student is making approximately $10,000 per month from his apps. He is one of the students who was failing half way through the introductory course in the Spring semester.
- As previously mentioned, two students in the fall introductory class each made approximately $10K during the semester from a single app they wrote in the class.
- Five of the original 20 students were offered jobs at $12,000-$15,000 more than average starting salary a full semester before they graduated. Two of those students couldn’t even start the job for six months after accepting.
- Three of the students were hired as business analysts by a Fortune 100 company.
- Participating students continued recruiting other students into the major and bringing them to the weekend workshops.
Findings from the third semester (Spring 2014):

- Most recent estimates are that students have collectively now topped the $100,000 earnings mark
- At least a dozen students are complaining that they are only making $400-$500 a month.
- Approximately a half dozen are making a minimum of $1,000 a month.
- Students who are now in their second or third semester are working directly with a Microsoft Technical Evangelist on complex apps.
- Student teams are working with the American Cancer Society, the Willi-Prader Syndrome Association, the 4-H Association, Arkansas Childrens’ Hospital and other non-profits to develop apps for these organizations.

CONCLUSION

This has been a fluid pedagogical process with continual assessments and adjustments being made to the process. Formalized data still remains to be collected to determine what part of the process is the most effective on innovative thinking development but based on observations and comments, the authors believe the phenomenal results of excitement and levels of innovative thinking are because the students have been able to quickly see the results of their efforts and because they are motivated by downloads, five-star ratings, and perhaps most of all, the ability to make money while working on class assignments. Each of these motivating factors has encouraged them to reach for new heights. If an app is not appealing to users or it isn’t marketed well in the description or screenshots in an innovative and enticing manner, then the download and sales will not happen. The students appear to be very competitive with each other and yet supportive at the same time working collectively to help with ideas on how their classmates can improve their designs and marketing.

These levels of excitement and collaboration translate into the development of applications outside of the classroom with most students having upwards of 50 of more apps in the Store at any given time. They continually monitor and improve their apps thereby experiencing a shortened continuous improvement cycle as they work to make their apps more and more innovative and appealing to their target audience.

From a faculty time perspective, this approach requires additional up front prep time in learning the template(s) and leading a one or two-day workshop outside of class to get the momentum going. Additionally, it demands a mental shift away from the traditional “all assignments look the same if they are correctly done” approach. Ultimately, we found the necessary time investment on the part of the faculty to be equal if not less than traditional methods for several reasons. One, students mentor each other relieving the pressure on the faculty member. Second, since apps are based off the same template but customized by student, no two are alike which eliminates the problem with “duplicate” assignments being submitted. Third, once an app is accepted by Microsoft testers (who are very thorough), there is external validation that the app functions correctly at all levels which removes the need for detailed testing on the part of the faculty member. In short, our experience was a win-win for students and faculty.

REFERENCES


