The Need for Mobile Application Development in IS Curricula: An Innovation and Disruptive Technologies Perspective

Jeffry S. Babb, Jr.
jbabb@wtamu.edu

Amjad Abdullat
aabdullat@wtamu.edu

Computer Information & Decision Management
West Texas A&M University
Canyon, TX 79119

Abstract

Disruptive technologies, such as mobile applications development, will always present a dilemma for Information Systems educators as dominant paradigms in our environment will tend to favor the existing sustaining technologies that we have become known for in our discipline. In light of this friction, we share our approach in investigating and designing a mobile application development which centers on student-faculty partnerships. We discuss a mobile application prototyping strategy and process which has allowed first-hand exploration of the current generation of mobile devices, and associated operating systems (*Android* and *iOS*). The nature of application development for these current-generation devices is discussed. A strategy for investigating and incorporating disruptive technologies, such as mobile applications, is offered for curriculum development. These strategies and the thinking surrounding them are influenced by theories on disruptive technologies and innovation. Of particular interest is the need to keep abreast of innovative technologies without, at the same time, chasing down every "fad" that appears.

Keywords: mobile applications, disruptive technology, curriculum development, innovation

1. INTRODUCTION

To paraphrase Bower and Christensen (1995), a failure to stay on top of technology and market changes remains a persistent problem facing Information Systems (IS) educators in their task of curriculum design in the face of rapid Information Technology (IT) changes. In the case of this change, a primary challenge remains an ability to recognize and separate fad from fundamentals (Noll & Wilkins, 2002; Lightfoot, 1999). As IS educators have witnessed successive "waves" of technological innovation and trends in practice – the micro-computer, end-user computing, object-orientation, the world wide web, etc. – IS educators have been faced with a conundrum: whether an innovation is a "game-changing" development or something more ephemeral? Moreover, the perspectives of the various stakeholders for a given program – students, employers, educators, the public, and the discipline at large – will each exert, at times, contradicting and countermanding demands on the IS curriculum for the sake of relevance. This dilemma is certainly a wicked problem and, arguably, is part and parcel to the very nature our discipline (Denning, et al., 1989). Thus, we assert that what is "core" to our discipline is not always stable; new and disruptive technologies will continually shape the curriculum as it does the discipline. As we seek to develop over-arching models for our curricula – as we think in terms such as theory, abstractions, and design – we must be willing to introduce the new into the tapestry of the old.

There are two principal aims for this paper: first, we consider the inclusion of mobile application development into the IS curriculum; and, second, this paper takes a disruptive technologies theory perspective on how IS educators can recognize important disruptive technologies and incorporate these technologies into the IS curriculum (Bower & Christensen, 1995), and also from an innovation theory perspective (Drucker, 1998). Specifically, we consider the incorporation of the latest generation of mobile devices, and software applications ("apps"), into the IS undergraduate curriculum.

For IS educators and researchers, the latest generation of mobile device presents new horizons for inquiry concerning portability, security, privacy, computing resource management, human-computer interaction, and the social impacts of computing. While these phenomena are now new, the latest generation of mobile devices has newly synthesized these concerns in light of the convergence of technologies manifest in the devices. In this sense, while mobile computing is not new, its impact on cultures and societies has been acute in this latest generation of mobile devices. In fact, mobile computing has further eroded the digital divide as people from various backgrounds and socio-economic persuasions all seem to have embraced mobile computing (Varshney & Vetter, 2000; Lyytinen & Yoo, 2002). For these reasons, we are certain that mobile computing will continue to impact the IS discipline pedagogically, professionally, and intellectually.

In this paper, we take the position that currency gained from embracing innovations will require risk-taking. To this end, we will share our experiences in investigating mobile application development for inclusion into our curriculum. Furthermore, we discuss a process by which IS departments can manage this risk, based on our experience. As such, this paper proceeds as follows. First, we discuss how disruptive technologies influence curriculum design. Next, we discuss the latest generation of mobile devices and characterize what is distinctive, new, and disruptive about these devices. Next, we focus on how and why IS curricula should plan on incorporating mobile computing in accordance with the theory of disruptive technologies. We then propose a course in mobile application development influenced by our experiences and strategies for adopting the technology. Finally, we conclude and discuss how our experiences have reinforced the lessons of both disruptive technology theory and innovation theory, and what implications these hold for curriculum design.

2. A DILEMMA FOR CURRICULUM DESIGN

The challenge for IS educators always has been how to decide when to include a specific new topic or technology into the curriculum, and how to facilitate such implementation without undermining the existing curriculum. Developing a plan and strategy for incorporating mobile application development into IS curricula is not a new or isolated change management problem. It has always been imperative that IS educators remain cognizant of new developments and be vigilant in developing strategies for the research and development of new innovations so as to anticipate demand. In doing so, IS educators may develop a "vision" for which new technologies are suitable for their circumstances (Johannessen, Olsen, & Olaisen, 1999). One utility of this vision is as a means to assess risk tolerance in adopting new technology innovation: is this new technology a right fit and how quickly can we capitalize on this new technology when demand arises? In this sense, we recall a paradox for research and development efforts in the areas of promising technologies: we must develop both the temerity and instinct to act outside of stakeholder demand by investing energy and resources into areas not yet in demand (Bower and Christensen, 1995). Part and parcel to this concept is to develop an "incubator" within our departments for exploring new topics and technology.

In the context of mobile computing in the Information Systems curricula, only recently, in the IS2010 model curriculum, does ubiquitous mobile computing command serious mention (Topi, et al., 2010). Moreover, the IS2010 model curriculum mentions ubiquitous mobile computing in the context of an extant concern, that of the Human Computer Interaction concentration (Topi, et al., 2010). Evidently, a natural tendency exists such that we classify new phenomenon in the context of extant phenomenon (Christansen, 1997). When new areas of concern arise, such as mobile applications, social network computing, and even games and "gamification," it is not likely that our peers, our stakeholders, or our constituents will "green light" the need for these innovations early enough such that an IS program is ready to "hit the ground running" when demand quickly thrusts the relevance of the innovation into our concern. In many cases, it will rest on IS faculty to provide the insight and leadership to recognize and assume these risks. Moreover, we could argue that this is exactly the role faculty are meant to take – to develop new ideas, new knowledge, and strategies for incorporating innovation into the curriculum. In fact, championship of the research and design of new curriculum areas is no different than the championship required for new information systems implementations and other change-management concerns in industry (Bower & Christensen, 1995, p. 44). Thus, change management for the nurture of a disruptive technology requires a unique approach.

Imperative to adopting a disruptive technologies perspective is an understanding of what Bower and Christensen (1995) call *sustaining technologies* and *disruptive technologies*. From a curriculum-development perspective, there are certain *sustaining technologies*, or subjects, that we rely upon to deliver the indisputable core of the information systems curriculum. These are the skills, knowledge, and techniques that the majority of our stakeholders and constituents expect of our programs. Typically, these will be: applications development in a modern programming language, databases and data management, systems analysis and design, data communications and networks, and grounding in the role of IT in organizations. As our going concerns, these sustaining technologies define our discipline and frame how we engage our processes of curriculum assessment and continuous improvement. To borrow from Kuhn (1962), our pedagogical, basic, and applied research and development efforts in this area constitute our *normal science*. As with Kuhn's *paradigms*, we are often careful to structure a curriculum, and models for curricula, in close alignment with the current paradigm; our understanding of and expectations of our programs hinges on this paradigmatic activity. Granted, our curricula, and models thereon, certainly allow for tailoring programs through free electives; however, it is less clear that we are willing (or able) to fundamentally change the paradigm.

We contend that thinking of curriculum development from a paradigmatic perspective supports Bower and Christensen's (1995) assertion that reconciling between the demands of a *sustaining technology* and the uncertain promise of a *disruptive technology* requires finesse and determination. A *disruptive technology* introduces a distinct set of qualities and capabilities which are not readily and apparently relevant to the demands of the existing paradigm. For instance, it has taken time to develop frameworks and infrastructure such that web application development has merged with aspects of traditional systems and software development practices. In fact, one could argue that the advent and maturation of agile software development methods may be somewhat related to the push to incorporate web development methods into the body of knowledge and practice of "conventional" modes of systems development (Abrahamsson et al., 2003). Thus, as with web application development, markets, employers, and educators may not adopt a disruptive technology at the same rate and at the same time. The challenge for IS educators is to develop strategies for exploring new curriculum areas without negatively affecting extant and sustaining technologies, in the short term; in the long term, given the eventual relevance of the innovation, the IS educator must then incorporate the innovation.

Fundamental to the dilemma of adopting a disruptive technology is that the new innovation may not make full "sense" in the existing market as the need for the improvements of the disruptive technology are not yet apparent. As Bower and Christensen (1995) put it, the apparent performance advantages, particularly in light of the prevailing paradigm, often present a small advantage as the market often doesn't "see" the value. For instance, for a previous generation of mobile devices, Research in Motion's *BlackBerry* smart-phone was considered the "gold standard" for enterprise and corporate users such that earlier generations of Nokia and Windows smart-phone devices were not considered serious corporate-use contenders. Now, as the latest generation of mobile devices (Apple's *iPhone* and *iOS*, Google's *Android OS*, and Microsoft's *Windows 7 Mobile OS*) proliferate, the conventional paradigm that the *BlackBerry* is the only serious enterprise device has been challenged (Cozza, 2011). This is not due to the inferiority of the *BlackBerry* so much as the expectations of the market had changed, driven largely by a collection of new capabilities intrinsic to the new devices. In other words, the market was now ready for the disruptive innovation.

According to Bower and Christensen (1995), we can assess the potential of a disruptive technology by understanding the performance trajectory offered by both the extant sustaining technologies and the new and disruptive technology (see Figure 1).

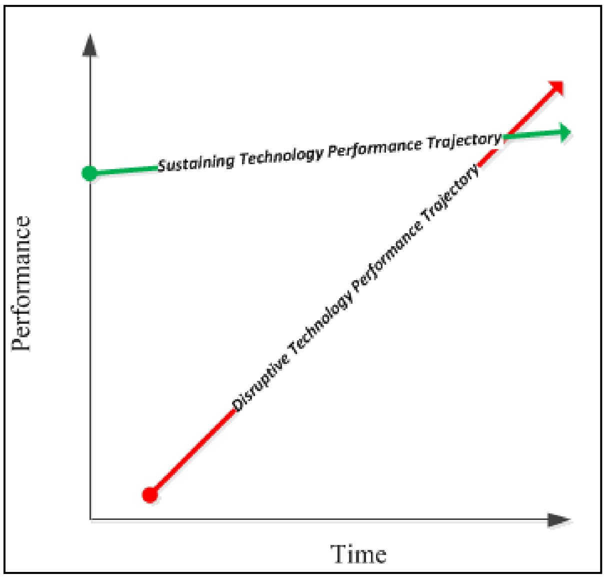

Figure 1 Assessing Disruptive Technologies (Bower and Christensen, 1995)

Whereas the performance trajectory for a sustaining technology is fairly predictable in support of dominant paradigms, the disruptive technology performance trajectory is initially estimated, but not guaranteed. However, if and when the trajectory intersects with, and then surpasses, the sustaining technology trajectory, it can be assumed that the implications for the dominant paradigm will be fairly certain: the disruptive technology is incorporated. Few expected that *Android*, *Windows Mobile*, and *iPhone* devices would threaten *BlackBerry*, but this has been the case and is expected to continue (see Table 1).

In the case of mobile devices, a convergence of technologies and capabilities has impacted the dominant paradigm. The incorporation of several previously-independent features – a PDA, a phone, a GPS, an audio player, a camera, etc. – into the latest generation of mobile devices has challenged the wired paradigm in a compelling way. We see other parallels in the advent of multi-core processors, in the increase in Internet subscribers brought about by the World Wide Web, and in the rediscovery of cloud computing. With these examples, we see that the potential of some technologies is not fully reached until other environmental conditions allow for the benefits of the technology to impact the marketplace. This convergence of device and software capabilities have created new user empowerment and has positioned mobile computing, and accordingly, mobile application development, as among the key strategic technologies for 2011 and beyond (Pettey, 2010).

To punctuate the influence of Bower and Christensen (1995) on our own mobile application development strategy for our curriculum, we reiterate that simply accommodating the curricular needs of the existing and dominant paradigm will likely stifle efforts to explore disruptive technologies. However, programs likely have very little room for experimentation as both the core curriculum and electives are suited to the dominant paradigm and any other concomitant institutional imperatives, such as AACSB accreditation, etc. Therefore, while we must continue to explore new technologies in order to sustain and broaden the appeal and viability of our IS programs, traditional methods may not yield the desired results. Typically, the exploration of new topics for incorporation into the curriculum transpires thusly: we carve out elective space (perhaps even within an existing course), we select a textbook, and we then "guinea pig" a group of students in order to ascertain suitability. As it is often the case that even electives are delivered and received within the auspices of the dominant paradigm of sustaining technologies, the normative approaches may not yield results. Rather, the research & development and "vision" necessary to fully explore the potential of the disruptive technology may be best realized when these efforts are sequestered away from the normal curriculum.

It is important that we do not underestimate the potential that a disruptive technology holds for reinvigorating and revitalizing our IS programs. Historically, as we have continued to explore and incorporate new technologies, we have also broadened the appeal of the major. For instance, the World Wide Web likely brought in an entirely new crowd to IS who may not have been attracted by our previous concerns. Furthermore, whereas our traditional concerns, perhaps from earlier technology waves, were programming, and analysis and design, a focus on new concerns may bring other interests, such communications, marketing, security, etc., as newer technology "waves" move through our discipline.

3. MOBILE COMPUTING BACKGROUND

As exemplary of a disruptive technology, the current generation of mobile applications and devices has not arisen in a void; rather there exist precursors to present mobile computing which goes back for nearly 20 years. Particularly in the 1990s, both the marketplace for mobile devices (particularly mobile phones and Personal Data Assistants) and research on mobile computing phenomena, quickly rose to prominence (Satyanarayanan, Fundamental Challenges in Mobile Computing, 1996; Forman & Zahorjan, 1994; Satyanarayanan, Kistler, Mummert, Elbling, Kumar, & Lu, 1996; Spreitzer & Theimer, 1993; Chess, Grosof, Harrison, Levine, Parris, & Tsudik, 1995).

An ongoing aspect of each generation of mobile device remains the fact that the device can connect wirelessly to the network and that the device offers open-ended computing capabilities (Forman & Zahorjan, 1994, p. 38). That these devices increasingly also incorporate aspects of mobile telephony constituted an early theme that has not only persisted, but has also come to acutely characterize the current generation of mobile devices: that characteristic is feature convergence.

The Current Generation of Mobile Computing Devices

For the purposes of this article, a mobile computing device primarily connects to the network via IEEE 802.11 Wireless LAN networks and/or the International Mobile Communications 2000 (IMT-2000) mobile telecommunications networks. Accordingly, it is often easier to associate these latest generation mobile computing devices by the sales and marketing classification they are most commonly given: Smart Phones. While these devices provide uses beyond that of simple telephones, in this paper, we focus primarily on the phone, the PDA, and the tablet devices most associated with the IMT-2000 3rd Generation (3G) and 4th Generation (4G)-capable devices.

What is distinctly characteristic of these devices is that they foster and flourish a software application "ecosystem," typically characterized by "apps" and "app stores." Additionally, this current generation of mobile computing devices is generally represented by two the two major and competing operating systems for the devices: Apple's *iOS* and Google's *Android*. Thus, we have limited our examination of current-generation mobile devices to *Android* and *iOS* devices for the following reasons: 1) both Apple's and Google's operating systems are fairly representative of the capabilities and features of the current generation devices; and, 2) the devices powered by Apple and Google represent a sizeable portion of market share (both current and projected).

From a curricular standpoint, what also distinguishes these current-generation devices is the nature of the methods used for the delivery and maintenance of software. These devices are characterized by always-on access to the Internet and World Wide Web, access to a software shopping, purchasing, distribution, and maintenance infrastructure commonly and colloquially known as application stores, or "app stores." Additionally, these devices have considerable processing power, memory, video acceleration, and audio processing capabilities such that a variety of software can be written to utilize these capabilities. In short, these are more hand-held computers than mobile phones.

Mobile Application Development Concerns

While the aim of this paper is to discuss an approach to incorporating mobile application development into the IS curriculum, some discussion of the logistics and particulars of creating mobile applications is in order. First, we will characterize and classify the software development concerns and particulars for developing for both platforms. Then, we will relate our own experiences and how we incorporated a Bower and Christensen (1995) approach to researching and developing a disruptive technology for incorporation into our curriculum.

Eric Raymond, in his classic tome on software engineering methods, *The Cathedral and the*

Bazaar (1999), likened certain development processes as being akin to either "Cathedrals" or "Bazaars," depending on the openness communication and the availability of source code. When considering Apple's *iOS* and Google's *Android*, we will illustrate how working with Apple's *iOS* SDK is more akin to the "Cathedral," whereas working with Google's *Android* SDK is more akin to the "Bazaar." While this characterization may be somewhat hyperbolic, we find it useful based on our experiences.

Working with the iOS SDK – The Cathedral

During our research and development work, Apple's *iOS* Software Development Kit (SDK) is the disruptive technology we spent most of our development, prototyping, and piloting efforts on. Therefore, our experience with Apple's *iOS* is much more extensive than with Google's *Android*.

The *iOS* Software Development Kit (SDK) is available to *iOS Developer Program* members or those who have enrolled their department into the freely-available *iOS Developer University Program*. The advantage of this program is that there are no licensing fees and the program allows designated faculty, and up to 200 students, to develop, test, and deploy mobile applications. While it is not possible to upload applications developed under the *iOS Developer University Program* to Apple's *App Store*, it is possible to individually provision each faculty and student mobile device with a license which allows downloading to the device from the development environment.

iOS SDK - Language, Library, and Tools

As the majority of IS programs use either Java, C#/.NET, or VB.NET as a programming language, the steepest learning curve in learning *iOS* development will be in the tools and language area. The development language for creating *iOS* applications is Objective-C. While Objective-C is a wrapper around C, there are idioms in the syntax and object interaction of Objective-C which will be unfamiliar regardless of familiarity with C, C++, Java, or C#. As Objective-C is not a managed language, direct memory allocation and management will likely be foreign to most IS students.

The Integrated Development Environment used for *iOS* development is *XCode*. XCode's workflows will be fairly unfamiliar to students who are used to *Visual Studio*, *Eclipse*, or *Netbeans*. The Cocoa-based SDK libraries used in *iOS* development are straight-forward maps to the Cocoa Touch, Media, Networking and OS/Kernel-level features of the *iOS*. Faculty and students familiar with the .NET *Framework Class Library* documentation or the *Java Class Library* documentation should do well with the documentation and examples for the *iOS* SDK. In general, the *iOS* development experience is certainly closed-source and "Cathedral"-like. Information regarding the tools and resources mentioned in this section are available online at http://developer.apple.com.

The Android SDK – The Bazaar

Whilst arriving to the market after Apple's *iOS*, the *Android* OS, designed to run in a number of hardware environments, is the fastest-growing OS environment for smart phones and other current-generation mobile devices. Being that *Android* is based on a Linux kernel, it is at its heart, a more "Bazaar"-like endeavor. In our testing, we concluded that *Android* development should be easier for most existing Information Systems (and Computer Science) faculty and students. Thus, we selected *iOS* as it would afford greater research and development value.

Android SDK - Language, Library, and Tools

Perhaps the most welcome news for many IS educators is that the learning curve for the tools and language supporting the *Android* SDK is no steeper than what is already the case in the majority of IS programs. This is so as the programming language for *Android* Development is primarily Java, which is still widely used by many IS and CS programs around the world. This also means that there are no development restrictions, such as the requirement for a particular hardware and system architecture for the development machine.

Comparison of the SDKs

Hopefully, Raymond's (1999) "Cathedral" and "Bazaar" conceptualization was a useful metaphoric device in understanding the different premises and assumptions when working with both toolsets. While there are many reasons to go with *Android* only, the largest players in the "App store" model will likely remain Apple and Google/*Android*. It remains to be seen what will happen with Nokia as they plan on moving forward with *Windows Mobile* solutions (Elop & Ballmer, 2011). The *Windows Mobile* option represents a "middle way" between the "Bazaar" of the *Android* ecosystem and the "Cathedral" of the Apple approach. It should be noted that a .NET environment would be easier for an IS program to transition to as opposed to the Apple *iOS* environment. Table **3** presents a side-by-side comparison of the *Android* and *iOS* environments (Chikkala, 2011).

4. A DISRUPTIVE-TECHNOLOGIES APPROACH

In 2009, we determined that several key technologies, most of which appear in Table 2, required further consideration. We again heeded advice to isolate, incubate, and otherwise nurture a research & development project for our exploration of mobile application development (Christensen, 1997). As of this writing, there are other universities offering courses in *iOS* and *Android* mobile applications development – in fact, there are free online courses at the Massachusetts Institute of Technology, Stanford University, and the University of California at Berkeley for this purpose. Like many institutions, our university lacks the critical mass, both literally in terms of students, and figuratively in terms of resources, renown, and prestige, to absorb risks in the manner that these higher-order institutions can. Table 4 provides a list of some universities offering a course in mobile development that includes either iPhone development, *Android* development, or both, as of the summer of 2011. Our process for finding these schools and programs was simple and straight-forward: we used the search terms "mobile application development" and "*iOS*" and "*Android*," where sites searched were limited to the `.edu` top-level domain. Rather than creating an exhaustive list, we focused on the most relevant findings from the first 50 returns. Results were corroborated between Google's search engine and Microsoft's "Bing" search engine.

Overcoming the Risks and Challenges: A Pilot Approach

Rather than embark on an experimental course, we undertook a pilot/prototype approach whereby the department hired a few of our best, most-capable, and well-rounded undergraduate and graduate students to work on an internal iPhone/*iOS* app for the department under the supervision and direction of a faculty member. We next offer a synopsis of the findings, lessons learned, and emerging concerns from this pilot effort.

The High-risk and low-initial-ROI problem

While it was crucial to allocate a faculty member to this endeavor, this allocation represents a risk. It is never certain that either the professor or the student would retain or convey any material that would be useful to the department and the undergraduate program. Another risk of the experiment is that all knowledge and expertise were allocated to the faculty member and students assigned to the project. Should anyone disengage from the project, there is a danger that these resources would be lost. Also, being a pilot project, the immediate payoff would be less than clear to many stakeholders and constituents.

Standards

While Objective-C is a well-specified and documented language, and while the closed environment of the *iOS* and its SDK have well-documented API and libraries, there is a risk in investing in a technology area largely void of standards. While web technologies are based on standards, many large vendor-driven SDKs and APIs, such as Apple's Cocoa technologies in the *iOS*, are subject to unilateral changes and the deprecation of any portion of the SDK.

Platform and Development Knowledge

While faculty mentor and student were sufficiently skilled in application development, we were concerned for what a beginner's experience might be. Our conclusion is not surprising: like any other innovation, utilizing the IT innovation requires the same hard-skills in application development as would any other innovation related to application development. In short, there is no "magic" in these technologies, they require many of the same skills we've been training our students on for decades now.

Difficulty to Test and Deploy

Another concern arose in that mobile development often requires that you test in an emulator rather than, or at least prior to, testing on an actual device. Also, deployment is very closely controlled by the vendors who control the

App Stores, making ad-hoc distribution especially challenging in the *iOS* case.

Ancillary Concerns

When working with the *iOS* SDK, some faculty may feel as though they are stepping backwards a bit in that resource allocation, management, and de-allocation are once again first-order concerns. Also, these devices present new data and location privacy concerns that have fewer equivalents in a static computing model (Keith et al, 2010).

Outcomes Informing Next Steps

By undertaking a low-risk and low-profile initial pilot application, we were able to "dry run" and "what if" issues that we anticipated and those we did not. Furthermore, by including students, we were also able to discuss potential content for the course from the student's perspective. Through this experience, we have been able to demonstrate to other faculty in the department, in the college, and in the university, our new expertise. Among the evidence of success of our approach is that our student, upon graduation, was subsequently hired by the university to continue a mobile development strategy for the institution.

There are many avenues yet available for incorporating mobile application development into the IS undergraduate curriculum at our institution, and the pilot allowed our department the opportunity to test those waters in an unobtrusive way. We concluded that any effective course would not likely be suited to absolute beginners: the sophistication of the required tools and techniques suggest that upperclassmen and graduate students would be the best target audience. The data in Table 4 implies that several other programs around the country have also made this determination.

5. CONTENT FOR A PROPOSED COURSE

Our next step in the process is to design and offer a course. The initial course would present a survey of topics focusing on both *iOS* and *Android* development. In general, of the other institutions we surveyed, we found the University of Notre Dame's approach to be sound. Specifically, we agree with and share the following aims to prepare students for careers in mobile application development (Laneman, Flynn, & Poellabauer, 2010):

- Increase the number of professionals in mobile application development and related technologies
- To foster the development of skills which address real-world problems with the tools and skills required for mobile application development
- To enhance the ability of students to communicate with a variety of audiences through their applications.

An outline of topics, skills, and techniques, broken down by the weeks of a semester is depicted in Table 5. A short-list of the major topics is as follows: The Objective-C language; The XCode IDE; Project Life Cycle; The Cocoa API; Views; Touch and Animation; Interface Elements; Event-handling; Sub-frameworks: Audio, Video, Photos, Maps, Sensors; Persistent storage; Networking; Multi-threading

Required Resources

As with any other area related to application development, departments considering an mobile application development course would do well to prepare the requisite resources. We have discussed the technical requirements, but our experiences revealed that the most important resources were human.

Perhaps the most important outcome of our efforts in exploring disruptive technologies is the development of faculty expertise. In our department, this experimental approach had the department chair's full blessing, consent, and endorsement. While our in-house "vanity" project had seemingly little direct and upfront value, the department chair was able to isolate us such that we were able to develop familiarity, comfort, and expertise. Whereas many departments might tend to outsource new and upcoming areas to adjunct faculty, to do so is to miss an opportunity to keep full-time, tenure-track, and tenured professors up-to-date and steeped in the technologies that will impact the profession and the curriculum.

Fitting Mobile Application Development into the Curriculum

Of course, the ultimate outcome of our efforts should directly impact curriculum development in the form of new course offerings. Since it is common to take a "wait-and-see" approach in response to disruptive technologies, many run the risk of "missing the boat." Certainly, web application development hasn't turned out to be a passing fad and has, in fact, grown and evolved substantially; so much so that many departments now use web development to fulfill their programming and application development component. We would argue that application development for the web is now a *sustaining technology*. In our experient, we find that many of our graduates who go on to work in application development do so either fully or partially in a web-oriented context. Thus, our prognostication for where mobile application development will lead us is this: incorporation or usurpation. The simple fact that we've classified mobile application development as a subclass of "application development" implies that our principle concerns will be. Logically, disruptive technologies will grow and challenge our existing framework of what application development is. Historically, we have benefitted from these disruptions as they have brought new students to our discipline.

6. CONCLUSION

In this paper, we have outlined the importance of identifying, researching, and developing a curricular response to potentially disruptive technologies. We, as IS educators, recognize that our discipline is about technology-wrought change. However, we are also subject to influences which encourage us to sustain the status quo of the dominant paradigm. Our constituents, stakeholders, and even peers, may not provide the clues and incentives to explore potentially disruptive technologies as their benefits are not yet clear. Often, when the benefits of and demand for a disruptive technology are clear, IS educators are left to play "catch-up" in order to remain relevant. In this paper we have explored a risk-taking R&D approach whereupon we have accepted the Bower and Christensen (1995) advice to isolate and nurture a faculty-student partnership for exploring and developing disruptive technologies for curriculum development. We have also shared our experiences in adopting a mobile applications development strategy for our curriculum and we have outlined our intentions for a recommended course in this area.

We believe that our implementation of the disruptive technology approach is both valid and effective. We found that the keys to the success of our strategy were: 1) creating a team of capable students headed by student-focused faculty; and, 2) selecting faculty who understand curriculum issues and the challenges of developing a new course. For us, this was an effective approach as it developed and tested ideas outside of the normal and traditional process of curriculum development. This research and development approach provided us with an opportunity to 1) explore the mobile-computing topics that we believed would be important; 2) identify the appropriate and needed resources; and, 3) structure a course which could be effectively delivered.

We attribute our disruptive-technologies approach as being a key success factor for understanding how a new technology innovation might incorporate into our curriculum. If we simply outsourced this experimentation, our department would miss a key opportunity to learn new technologies, to understand how these technologies fit, and the occasion to share these experience with our peer IS educators.

7. REFERENCES

Abrahamsson, P., Warsta, J., Siponen, M., & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. *Proceedings of the International Conference on Software Engineering*. Portland, Oregon, USA.

Bower, J. L., & Christensen, C. M. (1995). Disruptive Technologies: Catching the Wave. *Harvard Business Review*, 43-53.

Chess, D., Grosof, B., Harrison, C., Levine, D., Parris, C., & Tsudik, G. (1995, October). Itinerant agents for mobile computing. *IEEE Personal Communication*, 2(5), 34-49.

Chikkala, K. (2011, May 12). Android Vs. Apple IOS. Retrieved July 1, 2011, from http://www.rushk.com: http://www.rushk.com/android-vs-apple-ios/

Christensen, C. M. (1997). The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail. Boston, MA: Harvard Business Press.

Cozza, R. (2011, April). Forecast: Mobile Communications Devices by Open Operating System, Worldwide, 2008-2015. Retrieved July 1, 2011, from Gartner: http://www.gartner.com/DisplayDocument?ref=clientFriendlyUrl&id=1619615

Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., et al. (1989, January). Computing as a Discipline. *Communications of the ACM*, 32(1), 9-23.

Drucker, P. F. (1998). The Discipline of Innovation. *Harvard Business Review*, 3-8.

Elop, S., & Ballmer, S. (2011, February 11). Open Letter from CEO Stephen Elop, Nokia and CEO Steve Ballmer, Microsoft. Retrieved July 10, 2011, from Nokia Conversations: http://conversations.nokia.com/2011/02/11/open-letter-from-ceo-stephen-elop-nokia-and-ceo-steve-ballmer-microsoft/

Forman, G. H., & Zahorjan, J. (1994, April). The Challenges of Mobile Computing. *IEEE Computer*, 38-47.

Johannessen, J.-A., Olsen, B., & Olaisen, J. (1999). Aspects of Innovation theory based on knowledge-management. *International Journal of Information Management*, 19, 121-139.

Keith, M.J., Babb, J.S., Furner, C.P., & Abdullat, A.A. (2010). Privacy Assurance and Network Effects in the Adoption of Location-Based Services: An iPhone Experiment. *ICIS 2010 Proceedings*, Paper 237.

Laneman, J. N., Flynn, P., & Poellabauer, C. (2010, January). A Smartphone Programming Course for Undergraduates. Retrieved July 2011, from Wireless Institute - University of Notre Dame: http://wireless.nd.edu/font-page/wi-nd-mobileapps-white-paper.pdf

Lightfoot, J. M. (1999). Fads versus Fundamentals: The Dilemma for Information Systems Curriculum Design. *Journal of Education for Business*, 75(1), 43-50.

Lyytinen, K., & Yoo, Y. (2002, December). The Next Wave of Nomadic Computing. *Information Systems Research*, 4, 377-388.

Mauro, C. L. (2011, February 6). Why Angry Birds is so successful and popular: a cognitive teardown of the user experience. Retrieved July 10, 2011, from Pulse > UX Blog: http://www.mauronewmedia.com/blog/2011/02/why-angry-birds-is-so-successful-a-cognitive-teardown-of-the-user-experience/

Noll, C. L., & Wilkins, M. (2002). Critical Skills of IS Professionals: A Model for Curriculum Development. *Journal of Information Systems Education*, 1(3), 143-154.

Pettey, C. (2010, October 19). Gartner Identifies the Top 10 Strategic Technologies for 2011. Retrieved July 1, 2011, from Gartner: http://www.gartner.com/it/page.jsp?id=1454221

Raymond, E. S. (1999). The Cathedral and the Bazaar. Cambridge: O'Reilly.

Satyanarayanan, M. (1996). Fundamental Challenges in Mobile Computing. *Proceedings of the Fifteenth ACM Symposium on Principles of Distributed Computing*, (pp. 1-7). Philadelphia, PA.

Satyanarayanan, M., Kistler, J., Mummert, L., Elbling, M., Kumar, P., & Lu, G. (1996). Experience with Disconnected Operation in a Mobile Computing Environment. *The Springer International Series in Engineering and Computer Science*, pp. 537-570.

Spreitzer, M., & Theimer, M. (1993, July). Scalable, secure, mobile computing with location information. *Communications of the ACM*, 36(7).

Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, J. F., Sipior, J. C., et al. (2010). IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information. *Communications of the Association for Information Systems*, 26(18).

Varshney, U., & Vetter, R. (2000, June). Emerging mobile and wireless networks. *Communications of the ACM*, 43(6), 73-81.

Appendices and Annexures

Table 1 Worldwide Mobile Communications Device Open OS Sales to End Users by OS (Thousands of Units) (Cozza, 2011)

OS	2010	2011	2012	2015
Symbian (Nokia)	**111,577**	**89,930**	**32,666**	**661**
Market Share (%)	**37.6**	**19.2**	**5.2**	**0.1**
Android	**67,225**	**179,873**	**310,088**	**539,318**
Market Share (%)	**22.7**	**38.5**	**49.2**	**48.8**
Research In Motion	**47,452**	**62,600**	**79,335**	**122,864**
Market Share (%)	**16**	**13.4**	**12.6**	**11.1**
iOS	**46,598**	**90,560**	**118,848**	**189,924**
Market Share (%)	**15.7**	**19.4**	**18.9**	**17.2**
Microsoft	**12,378**	**26,346**	**68,156**	**215,998**
Market Share (%)	**4.2**	**5.6**	**10.8**	**19.5**
Other Operating Systems	**11,417.40**	**18,392.30**	**21,383.70**	**36,133.90**
Market Share (%)	**3.8**	**3.9**	**3.4**	**3.3**
Total Market	**296,647**	**467,701**	**630,476**	**1,104,898**

Table 2 Top Strategic Technologies for 2011 and Beyond

Strategic Technology	Relevance to Current Generation Mobile Computing
Cloud Computing	This is no longer simply a concern for Enterprise computing, as evidenced by services like *Dropbox*, *MediaFire*, and *Hulu*
Mobile and Tablet Applications	By and large, it is the application ecosystem available for these devices that constitutes the appeal of these mobile platforms
Social Networking	It can be argued that social networking has reinvigorated the web and computing as all aspects, both private and corporate, of life are assimilated into social networks. Mobile devices are a popular access node.
Video	The capture and sharing of this data is also key to the demand for mobile devices
Next-Generation Analytics	Mobile devices, particularly location information, represent important and valuable metrics
Social Analytics	Brining a social network aspect into CRM and market development into the analytics picture. This approach is evident in Social Network Analysis
Context-Aware Computing	It is quite clear that mobile computing is the primary enabler of this concept. The latest generation of mobile devices allows for a full who, what, where, and when picture.
Ubiquitous computing	This is also possible largely through the mobile device and its ability to allow a user to never lose contact of the computing environment.

Table 3 Comparing the features of the Android OS and Apple iOS

Key Criteria	Android	Apple iOS
Definition	Android is a software stack for mobile devices that includes an operating system, middleware and key applications.	*iOS* is Apple's mobile operating system, Originally developed for the iPhone
Ownership	Google open Source.	Apple.
Operating System	*Android* is Google developed open source operating system.	Apple *iOS* is a proprietary operating system.
Ease of use	*Android* does not have the same level of simplicity as *iOS*, we can detach our brain and still manage to work the interface.	*iOS* has turned out to be the easiest mobile operating system.
Hardware Vendors	Samsung, Motorola, LG, Sony Ericsson, Dell, Huawei, HTC etc.	Apple.
Compatible Devices	Compatible with any Devices.	iPad, iPod Touch, iPhones.
Application Store	*Android* Market 200,000.	Apple Store 300,000.
Google Talk	GTalk Specific Client and Video Supported.	Web browser chat.
Gmail Client	Gmail Specific eMail client.	Only Apple general eMail Client.
Web Browser	Open source Webkit layout engine coupled with Chrome's V8 JavaScript engine.	Safari.
Features	*Android*'s biggest advantage over *iOS*, *Android* has features like multitasking, widgets, tethering, Wi-Fi hotspot and Adobe Flash support etc.	*iOS* can have the ability to install applications, multitasking, copy-paste, folders, etc.
Messaging	SMS,MMS,eMail and C2DM.	SMS, MMS, eMail.
Performance	When running on faster hardware, *Android* is never perfectly smooth.	*iOS* ran perfectly even on the modest hardware also.
Connectivity	Wi-Fi, Bluetooth and NFC.	Wi-Fi, Bluetooth.
Adobe Flash	Supported	Not Supported
Email Attachments	Multiple files.	Single file only.
Supports	*Android* supports Hotspot via Wi-Fi.	Apple *iOS* supports internet Tethering via Bluetooth.
Social Network	*Android* supports Social Network contact Sync.	N/A

Table 4 Search Results for Mobile Application Development Courses

University	Course	Platforms
Harding University – Searcy, AK	COMP 475 – Mobile Computing	*Android*
Olin College – Needham, MA	Mobdev 2010 – Experimental Class	*iOS*
University of Notre Dame	CSE40333 - Mobile Application Development	*Android*, *iOS*
Strathmore University – Nairobi Kenya	MIT Africa Information Technology Initiative	*Android*, *iOS*

University of Southern California – Los Angeles, CA	ITP-499 – Mobile Application Development	*Android*
UC San Diego – San Diego, CA	ART40544 – Basics of Programming *Android*; ART40545 – Basics of Programming *iOS*	*Android*, *iOS*
Austin Peay State University – Clarksville, TN	CSCI3010 - Mobile Software Development	*iOS*
Carnegie Mellon University – Heinz College- Pittsburg, PA	95-740 Mobile Software Development	*Android*
Brandeis University – Waltham, MA	COSI153 – Mobile Application Development; COSI153 – Mobile Game Development	*Android*, *iOS*
Northeastern University – Boston, MA	CS4520 - Mobile Application Development	*Android*
MIT – Cambridge, MA	IAP2010 - Introduction to iPhone Application Development	*iOS*
Purdue University – West Lafayette, IN	CNIT355 - Software Development for Mobile Computers	Java (*Android*?)
Boston University – Boston, MA	MET CS 683 Mobile Application Development	*iOS*, *Android*
Dominican University – River Forest, IL	CPSC 446 MOBILE APPLICATIONS DEVELOPMENT	*iOS*, *Android*
DePaul University – Chicago, IL	CSC 471: Mobile Application Development	*iOS*, *Android*
San Diego State University – San Diego, CA	CS 696 Mobile Application Development	*iOS*
Texas A&M University-Corpus Christi	COSC 4590 Special Topics: Mobile Programming	*Android*
Florida State University - Tallahassee, FL	CIS4930-01: Mobile Programming	*Android*
The University of Utah – Salt Lake City, UT	CS4962 - Mobile Application Programming: *iOS*	*iOS*
California State University – Los Angeles – Los Angeles, CA	CIS 454: Mobile Application Development	*iOS*, *Android*
Stony Brook University, Stony Brook, NY	Special Topics in Computer Science - Developing Mobile Applications	*iOS*, *Android*

Table 5 Outline for a Course in Mobile Application Development

Week	Topic/Theme	Concerns, Skills, and Techniques
1	Introduction to mobile computing	Basics of technologies which enable mobile computing
2	*iOS*	Illustrate tools, techniques, background, requirements, etc. Basic aspects of Objective-C and accessing the *iOS* SDK
3	*Android*	Illustrate tools, techniques, background, requirements, etc. Basic aspects of Java and accessing the *Android* SDK
4	Conceptualize a Project; Seek stakeholders	Building a context-driven and purposeful app provides motivation. Students select *Android* or

		iOS
5	Project concept evaluation and selection	Given the nature of the platforms, what is feasible?
6	Design	UI, User Experience, Event model, Application model. Study of a popular app, such as "Angry Birds." (Mauro, 2011)
7	Design	UI Widgets, Themes, and customiziations; 2D graphics; Saving data and state (CoreData/Data Storage)
8	Design Review	Mobile HCI guidelines and principles; modes of user interaction (gestures, etc.); Design guidlines
9	Prototype/Mockup	App deployment, testing, and provisioning
10	Design Revision	SDK Services: Emulator, Background/multi-processes; threads
11	Design Revision	SDK Services: Security, permissions, profiles, Location, Networking, Web
12	Implementation	Frameworks, plug-ins, 3rd party enhancements, App store deployment
13	Testing	Debugging and testing tools
14	Private Demo and Testing	Initial end-user "beta" testing
15	Public Demo / Contest	Public demonstration, feedback, and voting on "best app" for prizes