Volume 11, No. 6
December 2013
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 by **EDSIG**, the Education Special Interest Group of AITP, the Association of Information Technology Professionals (Chicago, Illinois). Publishing frequency is six times per year. The first year of publication is 2003.

ISEDJ is published online (http://isedjorg) in connection with ISECON, the Information Systems Education Conference, which is also double-blind peer reviewed. Our sister publication, the Proceedings of ISECON (http://isecon.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 about 45%.

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.

INFORMATION SYSTEMS EDUCATION JOURNAL

Security Engineering Lessons Learned for Migrating Independent LANs to an Enterprise Environment

Robert L. Marchant
marchant@psu.edu
Pennsylvania State University
State College, PA 16803

Thomas Bonneau
thomas.bonneau@soteradefense.com
Sotera Defense Solutions
Herndon, VA 20171, USA

Abstract

Transition from small, independent LANs into larger enterprise managed infrastructures is becoming more prominent in academia, business and government. Consolidation of IT resources into larger, more disciplined, and more professionally managed environments has significant advantages however they do bring their own unique issues to solve in order to make the transition for the organizations involved easier. The topics covered under this paper are critical areas of concern organizations and their administrator staff needs to consider and resolve in order that transition and migration can be as painless as possible. Loosely using NIST SP 800.53 controls as a reference, the areas presented within this paper include access control mechanisms, patch management considerations, the need to address difference in hardware and software monitoring, baselines and licensing.

Keywords: LAN migration, data center consolidation, access control, patch management.

1. INTRODUCTION

Large organizations have migrated and are continuing to consolidate independent working group Local Area Networks (LANs) into more formalized hosted environments in hosting platforms ranging from simple migrations of existing LAN equipment into the enterprise network to multi-tenant virtual environments. The reason include, but certainly aren't limited to, economy of scale (e.g. sharing virtual resources, software licensing); reducing cost for space, cooling and power; sharing IT professional maintenance cost (e.g. systems administrations and help desk personnel); increased connectivity (e.g. between previously isolated LANs or to external web service hosting platforms); sharing of resources to handle surges of demand; disaster recovery and long term storage (e.g. archive).

Independent LANs are often created for ad hoc (and sometimes impromptu) purposes. The technical support hired or appointed to support this LAN are very close to the user community and understand how to prioritize the needs of the community (it is very common for a small group lab support LAN organization to anoint one of the researchers as admin who assigns userids, installs software when needed, and configures shared resources). The security controls for these environments are often understandably loose and the "bureaucracy" is typically non-existent; after all, the focus for the support to an independent LAN is the users of the LAN. If the LAN is in place for a long duration, this researcher may even install and maintain anti-virus software, post software upgrades and patches, and check to make sure licenses are up-to-date. For many individuals responsible for standing up and maintaining these independent LANs, connecting to or becoming part of a larger enterprise might be their first exposure to enterprise discipline and to enterprise level security controls.

The initial planning for conducting a migration from an independent support environment to an enterprise environment most often focusses on the network pieces. Usually this discussion involves determining the order the pieces to be transitioned should be migrated but always involves determining what services need to be augmented when the LAN migrates.

The authors of this paper have much experience in assisting organizations in understanding the security implications of migrating to an enterprise environment. This paper presents a few lessons we have learned that, if addressed early in the migration, can ease the process for both the users of the smaller LAN and the enterprise personnel assigned to support the migration. Please note that within the paper we will discuss at a foundational level technical descriptions intended to remind the reader of what data are needed by the enterprise security engineers during a transition.

The NIST Special Publication 800.53 (National Institute of Standards and Technology [NIST], 2009), defines security controls that can be tailored to the needs of an organization. The document, to anyone but a security specialist, is tedious to read and even more tedious to implement. Fortunately, the administrators and technicians involved in the migrations are usually spared the pain of having to work the details of whatever standard the enterprise follows as this is typically the realm of the "security engineer". Although both authors are experience with NIST SP 800.53 (NIST, 2009) and all of our lessons learned relate directly to a subset of these controls, we will spare the reader the tedium of referencing the specific controls that relate to each of our lessons learned (both authors will accept e-mail questions from any adventurous reader who wishes more details on the controls). We will instead discuss our lessons learned in the three topic areas; the large topic of Access Control, the midsize topic of Patch Management, and a small discussion on Systems Monitoring, Licenses and Product Acquisition.

2. ACCESS CONTROL

Our first lesson learned is to never underestimate the complexity of coordinating identity and access control. Issues arising in the access control area almost always involve coordination of directories, authentication mechanisms, and certification authorities. Even in LANs with well administered directory control, differences in directory structures, authentication and certificate structures have to be mitigated. In this section we will briefly discuss directories, authentication services and certification systems as a way providing common ground and to illustrate all the areas where mitigation may be necessary.

Although often believed to be simply the method used to add users to a LANs domain, directory services real function is to manage information about a computer network's users and network resources, and allow network administrators to manage users' access to those network resources. A directory service is intended to interface to a directory that holds the information about named objects contained in the network. The directory service then provides the access to the data contained in one or more directory namespaces. Since directory services can be responsible for authenticating access to network resources, the directory service interface must also be responsible for ensuring secure authentication for any access to the system resources that manage the directory data.

Directory services are almost always a set of applications implemented around a specific standard such as X.500 ("Directory Service," n.d.) or LDAPv3 often provided by the operating system or database vendor. This arrangement often makes sense as a directory service is a shared information infrastructure intended to provide the namespace for the network (a namespace defines the names used to identify objects on the network) and to assist users and applications in locating, managing, administrating, and organizing common items and network resources, to include volumes, folders, files, printers, users information (e.g. ID, Access, location, phone number, picture, etc.), groups, devices. For example, a directory may have a set of objects defined named user-ids, under user-ids may be other objects like: Surname, telephone number, company, nationality, clearance, access, and other identifying information. Administrators will set up the directory namespace using standards that are most convenient for the users they support.

Directories are usually accessed using client/server communications model. Applications read (and write) information with a call to a function or application programming interface (API). The API defines the interface for a particular programming language. The format and content of the messages exchanged between the client and the server must conform to this API and an agreed to message protocol. Obviously, LDAP provides the message protocol, and there are existing industry standards for LDAP APIs for C and Java.

Online services provided within an organization's domain can use one set of security infrastructures for authenticating and authorizing users and propagating their identity attributes (e.g. LDAP server or Windows Active Directory). Security and identity management in an enterprise environment where the entire domain is under a single authority is full of well-established technology and practice. Providing access from external web-based applications, web services, and web users (as is usually the case in an enterprise environment), creates the need to provide cross domain identity management and sharing. Differences in directory services technology, privacy and legal issues related to sharing identity information, differences in controls (and confidence among sharing organization in each other's security practices and controls) make coordinating a federated directory structure difficult (identity is federated when it is shareable across domain and platform boundaries).

As desirable as it is to share identity information, implementation is often difficult. Four technologies are most apparent at proposing solutions to this problem:

- Federated LDAP solutions: These solutions provide security applications coupled to an LDAP architecture (e.g. IBM, Sun, LINUX). Federated LDAP solutions tend to have both the advantage and the disadvantage of being tied to a specific vendor. It is usually easier for a migrating LAN to simply become a participant in the enterprise LDAP.
- Certificate based systems like Kerberos (http://web.mit.edu/kerberos/) and SESAME (http://www.cosic.esat.kuleuven.be/sesame/html/sesame_what.html).
- Public Key Cryptography (asymmetric key systems) such as public key infrastructure - PKI (Adams & Lloyd, 2003).
- XML based standards like the Security Assertion Markup Language- SAML (http://saml.xml.org/). These standards tend to be oriented towards more loosely couple computer to computer communications and tend to be more supportive to one of the three techniques above than as standalone solutions.

Regardless of which technology is used for federated identity management in the enterprise, some method of establishing and maintaining trust is essential to security of the connected systems. Kerberos is an example of an authentication service. Its purpose is to allow users and services to *authenticate* themselves to each other in a manner that is more than just providing a userid and password. In most authentication systems like Kerberos the password is a *shared secret*--something that the user and the service hold in common, and which only the server and the client know. To establishing identity in a Kerberos type system, the shared secret key is used as an encryption key; the user takes something freshly created, a timestamp for example, and encrypts it with the shared secret key. This is then sent on to the service, which decrypts it with the shared key, and recovers the timestamp. If the user used the wrong key, the timestamp won't decrypt properly, and the service can reject the user's authentication attempt.

In Kerberos, both the user and service implicitly trust an entity called the Kerberos authentication server (AS); the AS coordinates user access to all services in the system. Both the user and the service must have a shared secret key registered with the AS.

Kerberos often relies on conventional or symmetric cryptography, in which the keys used for encryption and decryption are the same. As a result, the key must be kept secret and periodically updated. Such a requirement can be circumvented with the use of public-key cryptography, in which there are two separate keys, a public key and a private key. These two keys are asynchronous pairs: Whatever one key encrypts, the other decrypts. As their names suggest, the public key is intended to be known by anyone, whereas the private key is known only by the user.

Public-key cryptography can be integrated into the Kerberos. When the AS generates its response, encapsulating the session key, it encrypts it with a randomly generated key, which is in turn encrypted with the user's public key. The only key that can reverse this public-key encryption is the user's private key, which only he or she knows. The user thus obtains the random key, which is in turn used to decrypt the session key, and the rest of the authentication proceeds as before.

Even though the user and the AS don't have to share a long-term key, they do have to share some kind of association. Otherwise, the AS has no confidence that the public key the user is asking it to use belongs to any given identity. An impostor could easily generate a public and a private key that go together, and assert that they belong to you, and present them to the KDC to impersonate you. To prevent that, public keys have to be *certified*. Some *certification authority*, or CA, must digitally sign the public key. In essence, the CA encrypts the user's public key and identity with its *private* key, which binds the two together. Typically, the CA is someone that is trusted generally to do this very thing. Afterward, anyone can verify that the CA did indeed sign the user's public key and identity by decrypting it with the CA's *public* key. If the migrating LAN has an existing relationship with a CA, care must be taken to preserve this relationship or to carefully migrate to using the enterprise CS(s).

In reality, the CA doesn't encrypt the user's public key with its private key, for the same reasons that the KDC doesn't encrypt the session key with the user's public key. Nor does it encrypt it first with a random key, since the user's public key and identity don't have to be kept confidential. Instead, it passes the public key and identity through a special function called a one-way hash*. The hash (sometimes called a message digest) outputs a random-looking short sequence of bytes, and it's these bytes that are encrypted by the CA's private key. This establishes that only the CA could have bound the public key to the user's identity, since you can't just create any other message that also hashes to those same bytes (that's why the hash is called one-way).*

Public Key Infrastructures can be established to support more than service coordinating and authorizing. The use of PKI enables a secure exchange of digital signatures, encrypted documents, authentication and authorization, and other functions in open networks where many communication partners are involved.

PKI has four parts:

- Certificate Authority (CA)
- Registry Authority (RA) or Local Registry Authorities (LRA)
- Directory Service
- Time Stamping (as an additional service

The Certificate Authority (CA) is the entity responsible for issuing and administering digital certificates. The CA acts as the agent of trust in the PKI. A CA performs the following main functions:

- Issues users with keys/Packet Switching Exchanges (PSEs) (though sometimes users may generate their own key pair)
- Certifies users' public keys
- Publishes users' certificates
- Issues certificate revocation lists (CRLs)

The Registration Authority (RA) is responsible for recording and verifying all information the CA needs. In particular, the RA must check the user's identity to initiate issuing the certificate at the CA. This functionality is neither a network entity nor is it acting online. The RAs will be where users must go to apply for a certificate. Verification of the user identity will be done for example by checking the user's identity card. A RA has two main functions:

- Verify the identity and the statements of the claimant
- Issue and handle the certificate for the claimant

The directory service has two main functions:

- Publish certificates

- Publish a Certificate Revocation List or to make an online certificate available via the Online Certificate Status Protocol (OCSP)

Timestamping is a special service that can be used to confirm the receipt of digital documents at a specific point in time. The service is used for contracts or other important documents for which a receipt needs to be confirmed.

To migrate a LAN into an enterprise, early discussion must resolve how the LAN directory will be transitioned (or assimilated), how to interface with the enterprise's authentication service, what certification authorities are used and how will they be migrated, and how to provide any special access related services to the LAN (e.g. timestamp). If the LAN namespaces and authorities are non-compliant with enterprise standards, ensuring that the changes necessary to directories, authentication services, and certificate authorities are clearly understood and explained to the LAN users will reduce a lot of migration delay.

3. PATCH MANAGEMENT

In today's dangerous cyber world, posting patches to all software as fast as is practical is not just a good idea; it is essential (National Institute of Standards and Technology [NIST], 2005). Most administrators, even admins of the smallest of LANs, are diligent about posting updates and patches as soon as possible. Our second lesson learned is that most independent LAN administrators, especially small LANs, are not prepared for the rigorous process and the automated tools that enterprises use to post patches. Be prepared to patiently walk the LAN admins through the process; be prepared for comments like "well, we can't just post patches whenever we feel like it, our engineers sometimes have process that have been running for days and patching will cause it to crash".

Most enterprise patch managers approach patching with a disciplined process that usually includes evaluating, prioritizing, testing, implementing, and monitoring the patches. As updates are received on products ranging from operating systems to desktop applications, the enterprise process usually involves determining the necessity and priority of a patch distribution. Critical patches will be implemented immediately; others will be scheduled to take advantage of routine maintenance outages. Some application and some products patches will require testing before implementation and most enterprises have some type of test environment to conduct these test (most independent LANs don't). Implementation at the enterprise level is almost always via some automated tool like Microsoft's System Center Configuration Manager (http://www.microsoft.com/en-us/server-cloud/system-center/configuration-manager-2012.aspx) for Windows, one of the many open source or inexpensive commercial update tools for Linux, or vendor specific tools for network devices and database systems.

LAN administrators have to struggle with a couple of issues. First, their privileges will usually be more restricted than what they were used to having (enterprises typical limit "local" administrators to only the level of privilege they need). This often means the LAN administrator is no longer in control of things like what security settings are implemented and when patches are scheduled. Second, enterprises are concerned with maintaining a consistent, enterprise wide, environment. LAN administrators will no longer be in control of when a product or operating environment is upgraded. And finally, LAN administrators will have to be prepared to reassure their users that enterprise patch policy is not intrusive and will not adversely impact their productivity. Spending a little time explaining the enterprise patch management process will help the admin deal with these issues.

4. SYSTEMS MONITORING, LICENSES AND PRODUCT ACQUISTION

Enterprises monitor. Enterprises typically have operations centers that use automated tools to check systems status, collect and analyze logs, and track events. Independent LANs typically do not. Although implementing monitoring very seldom affects the migration of the LAN, it can cause some unexpected resistance if the LAN users feel their privacy is being violated.

Enterprises control licenses and product acquisitions for at least three reasons. First; the penalty for unlicensed products on an enterprise are very expensive and very embarrassing. Second; having enterprise licenses for products applies leverage on the vendor and often leads to much lower cost. Third; standardizing products reducing the maintenance load and increases the efficiency of the patch management process.

Independent LANs however, are used to purchasing what they want, when they want it, often with little regard for registering products and keeping track of licenses. Our final lesson learned to share is that explaining the product acquisition and license maintenance process early, talking it out with the LAN admin will help considerably in diffusing this mostly emotional issue.

5. CONCLUSION

We have discussed some lessons we have learned as security engineers about supporting the migration of independent LANs into an enterprise environment. On the surface these lessons appear to have little to do with security, in reality they are all about security. Although we have spared the reader the details contained within SP 800.53, identity management, patch management, systems monitoring, audit reduction and analysis, change control and configuration management are all security controls and security issues.

We have discussed that meeting with and working with as early as possible; the administrator(s) of a migrating LAN can drastically reduce potential problems relating to directories, authentication and certificate management, patching, monitoring and acquisition. Early meetings can also reduce both the administrators' and the users' anxiety.

The authors have extensive experience in security (combined experience of over 40 years). We are often asked what "things to look out for" in transitioning systems. Each transition is, in reality, different. But almost all transitions can be (at least from the security perspective) simplified by using some form or framework to work with. The best framework is whatever framework the enterprise uses.

The lessons learned we have presented above all can be associated with security controls. The most important lesson we have learned though, is not specifically called out in a security framework. Enterprise security managers must accept risk. They expect risk to be identified and mitigated. They don't like rushed implementations and they don't like surprises. Meeting early, getting security issues addressed early, always reduces the risk that arise when transitions are "rushed", and reduces the delays that are a natural consequence of surprising security managers.

6. REFERENCES

Adams, Carlisle, & Lloyd, Steve (2003). *Understanding PKI: concepts, standards, and deployment considerations*. Addison-Wesley Professional.

Directory Service (n.d.). In Wikipedia. Retrieved July 01, 2012, from
http://en.wikipedia.org/wiki/Directory Services/

National Institute of Standards and Technology (2005). *Creating a Patch and Vulnerability Management Program* (Publication No. SP 800.40 Version 2). Retrieved Jun 19, 2012, from NIST website: http://csrc.nist.gov/publications/PubsSPs.html

National Institute of Standards and Technology (2009). *Recommended Security Controls for Federal Information Systems and Organizations* (Publication No. SP 800.53 revision 3). Retrieved Jun 19, 2012, from NISTwebsite:http://csrc.nist.gov/publications/PubsSPs

Implementing an Integrated Curriculum with an Iterative Process to Support a Capstone Course in Information Systems

Bryan Reinicke
reinickeb@uncw.edu

Thomas Janicki
janickit@uncw.edu

Judith Gebauer
gebauerj@uncw.edu

University of North Carolina Wilmington
Information Systems and Operations Management
Wilmington, NC 28403

Abstract

Learning is enhanced with repetition, either through more exercises in individual courses, or through the integration of concepts in a capstone experience. A well planned and integrated curriculum can utilize a capstone course, not only to provide a service learning component, but also as an opportunity to refresh students on key discipline topics immediately preceding graduation. This article describes the process used at one university to integrate concepts taught in pre-requisite courses into the capstone experience. In addition, it discusses the need to constantly refine all of the courses to integrate the concepts and learning experiences in both directions. The capstone course must provide repetition and hands-on learning of earlier concepts, and the pre-requisite courses must provide the knowledge to enable a successful capstone experience for students. This is a two way integration up and down the chain of courses and instructors must work together to integrate all of the courses in the discipline to enrich the capstone experience and achieve desired learning objectives.

Keywords: Capstone Courses, Curriculum Development, Integrated Curriculum, Service Learning

1. INTRODUCTION

Many schools offer capstone courses as the final requirement in their respective disciplines. The learning goals behind a capstone course are many and vary based on the discipline. They may include: a) to integrate topics from various classes in one discipline, b) to bring rhetorical concepts from previous classes into the real world via a project, and c) to refresh the students on core principles in their discipline before graduation. Regardless of the specific goals, the instructor of the capstone course may be faced with a significant challenge if the pre-requisite courses to the capstone are done in a vacuum and the integrated nature of the capstone is not considered throughout curriculum development.

Tappert and Stix (2012) relate that the goal of a capstone is to familiarize students with how their trade is plied in organizations, so that the curriculum delivers the "practice" part of the promised "theory and practice."

The purpose of this paper is to explore how Information Systems (IS) curriculums can be designed within a single program to support a capstone experience. To do this, the authors will examine the literature in the area and reflect on over five years of experience at their host school in attempting to make the capstone course a positive learning experience for the students.

2. LITERATURE REVIEW

Importance of a Capstone Experience

Over a decade ago, Gupta and Wachter (1998) supported the need for a capstone IS course to "develop student abilities and skills needed in the integrative information systems technology and business areas." Clear, Goldweber, Young, Leidig and Scott (2001) also encouraged the need for a project (capstone) under supervision where students apply what they have learned in their program of study.

Since that time, Abrahams (2010), Umapathy and Wallace (2010), Shih, LeClair and Carden (2010), Stillman and Peslak (2009) and Hashemi and Kellersberger (2009) have all discussed the importance and delivery of capstone or project-driven courses at their institutions. Capstone projects are widely used in business degree programs (Payne, Flynn, and Whitfield, 2008) to provide students with an opportunity to work on a 'real life' project.

The importance of capstone courses may be found in many of the accrediting agencies for colleges and universities. The Southern Association of Colleges and Schools (SACS) determined that an integral part of their Quality Enhancement Program (QEP) is the consideration of capstone experiences, defined as a senior level course that empowers students to evaluate, appreciate and integrate multiple perspectives in a collaborative project (2012 website).

The 2010 Model IS Curriculum (Heikki et al., 2010) lacks a capstone project course in the core model curriculum. However, Schwieger and Surendran (2011) argue for the need to incorporate a capstone course in the IS 2010 Model Curriculum.

Learning Theory

Constructivist learning theory emphasizes the usefulness of combining and building on previous knowledge that typically happens in a capstone course. For example, Brandt (1997) states that learners construct new knowledge by making sense of experiences in terms of what is already known. Learners transfer knowledge through experiences via mental models, which are used to assimilate new information into knowledge, and thus become expanded mental models. This knowledge transfer emphasizes knowledge construction and problem solving in domains.

Rakes (1996) recommends increasing students' success through the addition of practice and through a shift from the traditional theories of learning (cognitive and behavioral) to a resource-based view of learning. The resource-based view of learning involves the role of an instructor changing from an expert dispensing knowledge to a "guide" providing resources, and requires an increase in the number of problems, assignments, and exercises given to the students (Rakes, 1996). Finally, Yadin and Or-Bach (2010) discuss the continuing need for self-assessment and multiple individual exercises in an environment of collaborative learning.

Capstone experiences fit these learning theories well and support the construction of new knowledge through the integration of concepts from prior courses. In addition, the capstone project enables students to have individualized (or team) projects that enable them to self-explore, which is in line with the resource-based view of learning.

Service Learning

Many capstone courses include a service learning component as well. Preiser-Houy and Navarrete (2012) discuss service learning in the context of a teaching strategy that integrates discipline-based learning with relevant community service.

Wei, Siow and Burley (2007) discuss the fact that service-learning is an educational strategy that combines classroom-learning experience with a community service experience. It requires that courses be modified to involve real projects from communities and thus provide students with real-world experiences in a relatively safe academic environment (Wilcox and Zigurs 2003). Service learning has multiple benefits for students by engaging them in the community in which they are learning and allowing them to develop leadership skills as they work through the project (Rose et al. 2005).

3. THE CAPSTONE EXPERIENCE

The IS major at the authors' university is a stand-alone program within the School of Business with approximately 100 majors out of roughly 2,000 undergraduate students. The program has included a capstone experience as a part of the curriculum for approximately eight years. Over the last five years, the faculty has been focusing on continually improving the learning experience, with an emphasis on how to better integrate the learning concepts and theories from the other IS courses so that the capstone course could build (constructivism) on these theories to truly enhance student learning and retention of this knowledge.

The capstone experience is comprised of the following key components:

- Real-world IS development projects with external clients
- Teams of two students work on individual projects
- n-Tier system development environment
- Each project has a back-office SQL database
- Each project generally has a web portal to support the desired functionality of the client
- Teams must develop a presentation layer
- Teams must develop a business logic layer

The following courses are considered pre-requisites for the capstone, as they introduce students to various concepts that are then reinforced during the capstone course:

- Database Management (SQL)
- Business Software Development (VB.Net)
- Systems Analysis and Design (SAD)

The SAD course in particular has become an integral part of the capstone, as this is the course where students meet with the end-clients, develop system specifications to meet client needs, and prototype potential solutions. A contract is signed with the-end client in this course.

As a part of integrating the SAD course with the capstone, many changes needed to be made to the SAD course. These changes necessitated altering the syllabus to include a number of deliverables for the capstone project. One of the problems that had emerged in the capstone course was that the students did not have sufficient time to gather and document requirements and build the system in a single semester. Thus, the tasks for gathering and documenting requirements were moved into the SAD course. In all, the deliverables for the capstone project count for 35% of the SAD course grade. One other change was to move the SAD course to an object-oriented methodology. This more closely matched what was being done in the development courses, as well as reflecting current industry trends towards object-oriented development.

As one of the deliverables in the SAD course, students must develop an interview guide and meeting agenda prior to meeting with the client. This exercise forces the students to think about what information they must gather from the clients ahead of time, and gives them the experience of planning a meeting with a client. After this deliverable has been graded, the students must schedule a meeting with the client and document the meeting as a deliverable for the class. Once the students have met with the client, they have to develop the scope document that was mentioned earlier. They are required to present this scope document to the client in order to get the client's sign-off on the plan for the project. This is a good practice for "real world" projects, and forces the students to think about what the project should actually entail.

The next deliverable is a prototype for the system. While the students do not need to produce a working prototype (this is an exercise for the capstone course), they do need to model every screen that will be included in the system. After a number of iterations, the instructor for the SAD course moved this exercise to earlier in the semester, as creating a prototype forces the students to think about the system as a whole – something they have never had to do before.

The students are then required to develop a set of use case diagrams. This exercise also forces the students to think about the system in detail (which users will perform what functions?), and often prompts changes to the prototypes. This also introduces students to the concept that most projects are iterative in nature. SAD courses tend to emphasize that development is an iterative process, but most students have never actually experienced this. Generally speaking, the students are stunned to find out that: a) their first deliverables were not perfect and b) if they don't correct the earlier deliverables, they will be unable to complete the project next semester.

At this point, the students are required to meet with their clients to present the (now revised) prototype for feedback. The students are again required to plan the meeting and submit meeting notes as deliverables for the class. This mid-point check was included to make sure that the students did not spend the semester working on a system that did not meet the client's needs. Of course, depending on the client's feedback, the students may (again) be required to modify their prototype to meet the client's needs. This is also generally the point at which the students gain their first experience with scope creep, as clients will frequently try to add additional requirements to the system.

Next, the students have to develop descriptive use cases, which forces the students to think about the business logic and information flows for the system. The authors have noted that this assignment is frequently a problem area for students, as they have rarely had to develop their own logic in the past, and tend to be challenged when having to consider "what now" questions. How should the system respond to input? The new understanding can lead to more changes to the prototype, when it triggers the realization that important functions have been left off of one or several of the screens.

The students then must develop an Entity Relationship Diagram (ERD) for the database design. Again, this is frequently challenging for the groups as they have, again, never really had to develop an ERD to support a system. The groups generally discover that a working system requires many more tables than they anticipated. It is not unusual for this activity to prompt the groups to refine their prototype again, as they discover that to make the system work they will need to add in additional detail, which requires that detail to be entered on a screen within the system.

In order to link the use cases and ERD, the students must develop a Create, Read, Update, Delete (CRUD) diagram. Once again, this exercise proves very useful to enhance learning and deepen understanding. Students are often quick to indicate that they fully understand CRUD diagrams following their introduction in lecture, only to find out that actually creating one for a system is significantly more difficult than they had anticipated. This exercise is also useful for forcing the students to think about how the system should actually work, and eventually triggers "aha"-effects when students realize the inherent connection between the ERD and use cases that are conceptualized in the CRUD diagrams.

In order to expose the students to the concepts of project management, and begin to map out the various activities in the capstone course, the students are required to develop a project management plan. This exercise requires the students to map out what has been done in the SAD course and to begin mapping what they think needs to be done in the capstone course. Again, this is generally their first exposure to these concepts, and they discover quickly that developing a detailed project management plan requires a great deal of attention to detail.

Near the end of the SAD course, the students are required to check in with the client one more time at the end of the semester to present their (now extensively refined) prototype to verify the direction the project is taking. The students are again required to document this meeting as a deliverable.

Finally, the students must present their projects to the class at the end of the semester. There are generally multiple projects each semester, so each group could be working on a separate project. The students have to present and explain their prototypes, and describe how they envision the system working once it is completed. The end-of-semester presentations often result in lively discussions and questions from the class on their fellow students' projects, which appears to demonstrate the deep insights that students have gained about the SAD process.

One thing should be noted for the instructors of SAD courses who are considering the approach just described. Asking the students to make multiple changes to their deliverables also requires re-grading the deliverables multiple times. If the students do not make corrections as they go along, they will not have the designs they need going into the capstone course. Thus, to encourage this additional work on their part, the deliverables must generally be re-graded several times. However, in order to encourage good work the first time around, it is a good practice to set a maximum increase for resubmissions. This limit should be high enough to make it worth the students while, but not so high that it diminishes the effort prior to the first submission.

As can be seen from this discussion, the integration of the SAD course with the capstone course has occurred at a significant level, and students appreciate that they are working toward the same end-project in two subsequent courses. The faculty have noted that projects have become more robust as a result of having two semesters to work on the same end-client project. One remaining challenge is to better integrate the database and the business application development courses into the learning outcomes that will be needed in the capstone and SAD courses.

Finally, an additional motivation to integrate the curriculum even more is the fact that the authors' host university is modifying its graduation expectations to include more integration, more service to the community and more experiential learning.

4. LESSONS LEARNED

Adjustments must be made

Developing an integrated curriculum is not a one-off activity. It is a process that requires an annual review by the faculty who are impacted by it. To assist in the course integration, the faculty involved in the four classes mentioned earlier meet periodically to discuss how the courses could be better integrated.

This annual review by faculty also takes into consideration written reviews from students and suggestions from the end-clients. Added to these formal reviews are the faculty's overall evaluations of student projects in both the SAD and capstone courses.

Summary of challenges from annual reviews

Challenges that appeared in both the SAD and capstone courses include: the move from theory and conceptualization (database, development, and SAD courses) to implementation (capstone course) is more difficult for students than they first expect. It is one thing to design a few tables and relationships in a database course, and another one entirely to be in a "production" environment which requires the building of 50 plus tables with the corresponding relationships. Another challenge that students encounter is moving from a logical ERD to a physical database design.

In the SAD arena, students have difficulty when they try to go from use case descriptions and CRUD diagrams in SAD to actually building the proper stored procedures, functions and objects in the capstone course. In addition, a common refrain heard from the students is that they didn't receive all the information needed from the client. Or that the client didn't tell them what they really wanted. Of course, this is a common refrain heard from systems developers as well, which simply emphasizes why this type of learning experience can be so valuable for the students.

From the software development side, the enhanced use of business objects, business logic layers and building the logical flow for the menus, actions and reports can be overwhelming for new developers where the exact steps have not been detailed for them in advance.

When it comes to the replication of real-life experiences, students also get exposed to the critical need for a disciplined approach to managing their projects. As laid out earlier, the SAD course, as well as the subsequent capstone course, requires the students to develop, submit, and possibly revise and resubmit, a sizeable number of deliverables as a team, and also in collaboration with the end-client. While students continuously work on their project deliverables, new concepts and skills are being taught and practiced in the classroom, with the expectation that these will be applied to the project shortly thereafter. So, time management is of critical importance and procrastination can have serious effects on the success of the projects, as a result of the close link between concepts and application in both courses.

An added hurdle results from the fact that students effectively work with two 'bosses' (faculty members) as they move from one course (SAD) to the other (capstone). Deliverables need to be handed over, and possibly revised again once students start working on their capstone projects, which even though not necessarily ideal from the viewpoint of the students is certainly not unusual in real-life projects.

Can you change your syllabus?

When attempting to develop an integrated curriculum, one of the key problems is that what a student learns in one course should be applied in another. This means that all of the faculty involved must make occasional adjustments to their syllabi and assignments to better meet this goal.

The process of developing an integrated curriculum requires discussion among the faculty, and will likely require changing the syllabi for other courses as well. Such changes should not be viewed as an attack by other faculty, but rather an acknowledgement that there is a chance for their course to better support the capstone. Still, some faculty may be inclined not to change their syllabus and learning concepts. The authors have noted that student comments on their previous learning experience (gathered at the end of the capstone course) can help to overcome this resistance. In these cases it can be important to stress that students know the concepts, but are having trouble bridging theory and practice.

One of the revelations of our integrated approach was that achieving buy-in from all faculty involved is not necessarily easy. This is a process, and must be approached as such. This isn't surprising, in retrospect, as it does require that faculty surrender some aspects of their curriculum to support what is needed for the capstone.

Change is the only constant

When using an Integrated Development Environment (IDE), whether it is Microsoft's Visual Studio or Eclipse, the only guarantee is that it will change within the next two years. This presents a special problem for integrated programs, because it is likely that the software will change between the time students take the intro programming course and the capstone. Thus, when moving to a new software platform, special care has to be paid to who is moving to the new platform and when. This can create problems for faculty who may be working with different versions of the same software, as well as for a University's technology department, as they may then have to support multiple versions of the IDE simultaneously.

Sharing work files between classes

While the students' (sub-optimal) habits with regard to saving and naming files may not seem like a problem for faculty, they become one when the curriculum becomes more integrated. If the deliverables from one course feed into another, those files need to be stored such that the students can readily access them. Somewhat surprisingly, file access has been a nearly constant headache, particularly for the instructor of the capstone course, as students tend to misplace and lose files for any number of reasons. Given the extent to which the courses and deliverables build upon one another, the loss of files can be disastrous.

One solution to this problem could be a robust class management system, such as Blackboard, as long as it is set up to allow students access to deliverables from a previous semester. Another solution could be a network or cloud-based drive that is backed up regularly where all of the group members have access to the files. The authors would recommend that such a solution not be hosted by the faculty concerned, but rather by the university or an outside entity (such as Google).

A group is only as strong as its weakest link

The suggested high level of integration also requires that all faculty members possess comparable practical skills which probably requires more frequent updates than a less integrated situation would. Ours is not a field that stands still, and if there are faculty members who are not committed to updating their skill sets on a regular basis it can be incredibly difficult for students to obtain the skills and knowledge needed to be successful in the later courses, and especially the capstone.

These observations are based on the authors' experience with developing both IS capstones and an integrated IS curriculum at their university. However, these are issues that any program will encounter when moving toward a more integrated curriculum.

Examples of changes made to the courses

Earlier, a list of student difficulties in transferring theory to practice were detailed. In this section, we summarize some specific changes made to four courses in the IS curriculum as a result of the integration.

Database Management: Originally, stored procedures were mentioned, but not stressed in the database management course. Now, a more significant portion of the class is dedicated to the value and building of stored procedures

Business Software Development: In the context of an integrated curriculum, the course has been changed to put more emphasis on the use of business objects (e.g., person, invoice, inventory item). Now, students must construct multiple objects and understand the purpose of inheritance and isolation. In addition, this course has been changed to include more database applications. This is done so the concepts learned in the Database Management course are reinforced earlier. Items of SQL connections, strings, and an entire CRUD process is developed and practiced.

Software Analysis and Design: A number of adjustments to this course have been described earlier in this paper. In addition, this course has also been changed to include the use of SQL software for ERDs and DFDs. Students now also build a small segment of their database for the end client. Use case descriptions and use case diagrams have been enhanced to emphasize the use of menus by role which helps students build the logical flow in the capstone courses. Taking the use case diagrams with the depiction of different actors and translating them to menus and menu items in a prototype system has really enhanced students' grasp of applying the concepts to reality.

Capstone Course: For a smooth transition between the prerequisites and the final capstone course, the instructor of the capstone needs to be aware and considerate of the concepts and skills introduced and practiced in the previous courses. Before beginning the work on the actual capstone projects, it has proven useful to reinforce previous learning and bring everyone up-to-date. The first month of the class has, thus, been modified to include assignments on building various stored procedures for a common table (i.e. valid states). Having the entire class work on a common assignment and build four important procedures (read, update, insert, delete) has made their development easier later in the semester. In addition, more time is now spent on use cases to help build the business logic layer, especially in the area of 'exceptions' noted in the use cases.

5. CONCLUSIONS

We believe the annual reviews of the involved faculty has increased the level of integration in the IS curriculum and has enhanced student learning. As a result of the integration of the courses, more of the projects reach the 100% completion rate, making the end clients much happier. In some respects the capstone course has become a great refresher on many IS skills needed by employers right before students graduate, as well as a valuable integrative educational experience. The project itself is just a vehicle to help translate theory to practice.

6. REFERENCES

Abrahams, A. (2010). Creating e-Commerce Start-ups with Information Systems Students: Lessons Learned from New Venture Successes and Failures. *Information Systems Education Journal*, 8 (35).

Brandt, S. (1997). Constructivism: Teaching for Understanding of the Internet *Communications of the ACM*, 40 (10), 112-117.

Clear, T., Goldweber, M., Young, F.H., Leidig, P., Scott, K. (2001) Resources for instructors of capstone courses in computing, *ACM SIGCSE Bulletin*, 33(4).

Gupta, J. N. D. Wachter, R. M. (1998). A capstone course in the information systems curriculum. *International Journal of Information Management*, 18 (6), 427-441.

Hashemi, S., Kellersberger, G., (2009). The Pedagogy of Utilizing Lengthy and Multifaceted Projects in Capstone Experiences*. Information Systems Education Journa*l, 7 (17).

Heikki, T., Valacich, J., Wright, R., Kaiser, K., Nunamker, J., Sipior, J., deVreede, G., (2010). Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. *Association for*

Computer Machinery (ACM) and Association for Information Systems (AIS).

Payne, M., Flynn, J., Whitfield, J.M., (2010). Capstone Business Course Assessment: Exploring Student Readiness Perspectives. *Journal of Education for Business*. 83(3).

Preiser-Houy, L., Navarrete, C. (2011). A Community-Based Research Approach to Develop an Educational Web Portal *Information Systems Education Journal*, 9(1) 4-13.

Rakes, G. (1996). "Using the Internet as a tool in resource based learning environment". *Educational Technology, 6* (2), 52-29.

Rose, D. Meyer, A., Hitchcock, C., (2005). The Universally Designed Classroom: Accessible Curriculum and Digital Technologies. *Harvard Education Press*, Cambridge.

Schwieger, D., Surendran, K. (2011). Incorporating Capstone Courses in Programs Based on the IS2010 Model Curriculum. *Information Systems Education Journal*, 9(2) 65-74

Shih, L., LeClair, J., Varden, S., (2010). The Integrated Technology Assessment: A Portfolio-based Capstone Experience. *Information Systems Education Journal*, 8 (63).

Stillman, R., Peslak, A., (2009). Teaching Software Engineering Including Integration with Other Disciplines. *Information Systems Education Journal,* 7 (40).

Tappert, C., Stix, A. (2012). Adapting to Change in a Masters-Level Real-World-Projects Capstone Course. *Journal of Information Systems Educators*, 10(6) 25-37

Umapathy, K., and Wallace, F.L. (2010). The Role of the Web Server in a Capstone Web Application Course. *Information Systems Education Journa*l, 8 (62)

Wei, K., Siow, J., Burley, D., (2007). Implementing Service-Learning to the Information Systems and Technology Management Program: A Study of an Undergraduate Capstone Course. *Journal of Information Systems Education*, 18(1), 125-136.

Wilcox, E., Zigurs, I., (2003). A Method for Enhancing the Success of Service-Learning Projects in Information Systems Curricula. *Information Systems Education Journal*, 1(17).

Yadin, A., Or-Bach, R., (2010). The Importance of Empahsizing Individual Learning in the "Collaborative Learning Era". *Journal of Information Systems Education*, 21(2).

A Pedagogical Approach Toward Teaching An Information Systems Student How To Conduct A Web Usability Study For An Honors Project: A Case Study.

Gayle Jesse
gjesse@thiel.edu
Thiel College
Greenville, PA 16125

Abstract

The purpose of this paper is to provide educators with a course model and pedagogy to teach a computer information systems usability course. This paper offers a case study based on an honors student project titled "Web Usability: Phases of Developing an Interactive Event Database." Each individual phase—creating a prototype along with usability testing, defining a technical structure, and designing a usable interface—is equally valuable to the entire process of interactive web development. A distinct significance is present within each phase, which emphasizes the importance of completing every step in the development process. Unlike businesses that suffered when the Dot-com bubble burst, businesses that acknowledge the diverse levels of understanding and recognize that implementation of each phase directly affects the success of the business will prosper in this age of technology. This paper focuses primarily on the prototype and usability testing phase. With that in mind, an extensive background and explanation of phase one in developing an interactive event database is presented for the reader; the honors student paper did, however, present an all-encompassing understanding of web technologies. Additionally, this paper provides a method for developing the requirements to conduct and evaluate an honors project. Finally, this paper concludes by considering the study's limitations and suggestions for further research.

Keywords: Pedagogy, Prototype, Usability, Case Study, Project-Based Learning, Teaching A Usability Study Course

1. INTRODUCTION

Creating a successful pedagogical approach for a course can sometimes be a difficult task. Teachers often seek input from fellow teachers or additional research to aid in the development of a course they have not taught before. In this case study, the teacher was a mentor for an Information Systems honors student. The student wanted to conduct a web usability study on the current events calendar of a community website and design a new "interactive" events calendar that was database driven. With website technologies growing at a phenomenal pace, web usability testing is a key to creating a website that is easy and pleasant to use. In the world of e-commerce marketing and communication, an easy-to-use website creates the "stickiness" (duration) needed to keep consumers on a site, and usability testing aids increased "stickiness" (Laudon, 2010).

The student grounded this project in the presupposition that the Internet has provided businesses with a potentially beneficial marketing opportunity. Moreover, the student felt that in today's society the benefits of technology are often overlooked or misused. A prime example was the Dot-com bubble burst; countless businesses failed to prosper because they did not effectively make use of web technologies. These organizations did not understand the increased demands of conducting business over the Web and failed to consider many elements that differed between Web and storefront business interactions. Additionally, the student stated a strong interest in providing ways in which technology can aid in communication effectiveness and efficiency. The interest in utilizing technology toward effective communication made the student feel that online interactive databases are an intriguing and powerful tool. Furthermore, the student cited the capabilities of a database for outputting effective and efficient results, which was observed from prior classroom and personal experiences. For example, an individual can obtain database results immediately based on specific and customizable searches rather than browsing an entire site or webpage fruitlessly. The student felt that a website is more usable if narrowed options or categories can be employed such as: name, price, date, or keywords, and then clicking a "Search" button tool. The returned results are normally valued because the results are typically an organized list with specific relevance associated with one's original browsing interest. This, in return, informs the individual and results in effective communication between the client and the business/organization.

2. LITERATURE REVIEW

The following literature review defines pedagogy, case studies, using case studies as educational applications, and the Thiel College honors program, student honors project requirements, and teacher/mentor requirements.

2.1 Pedagogy - Define

The Oxford English Dictionary defines *pedagogy* as "the method and practice of teaching, especially as an academic subject or theoretical concept" (Pedagogy, n.d.). World Bank (n.d.) research indicates that the way to teach is changing. New pedagogies must be utilized because the old pedagogies based on teacher-controlled learning that is highly formal and standardized is no longer relevant. In the 21st Century, students are learning by critical thinking, active learning, problem solving skills, communication (making connections and expressing oneself), and contextualized knowledge (Kharbach, n.d.). Further research by the National Training Laboratory shows that the amount of new information learners retain depends on how the information is presented (World Bank, n.d.). Summarizing the Learning Pyramid (World Bank, n.d.), students learn best when they are actively engaged in their own learning. Confucius said nearly 2,500 years ago, "I hear and I forget. I see and I remember. I do and I understand." (Moncur, n.d.). For the purpose of this research paper, the learning pyramid proves that the honors project in this paper follows the 21st Century pedagogy.

2.2 Case Study - Defined

Simply stated, the purpose of writing cases and sharing them with others is to share experience without all of us actually having to be in the same place (GTTP, 2012). Technically stated, a case study can be defined as the collection and presentation of detailed information about a particular participant or small group, frequently including the accounts of subjects themselves (Case Studies, n.d.). The pedagogy for writing this case study was based on Robert K. Yin's (1993) research. Additionally, this paper is considered a "Critical Instance Case Study" (Case Studies, n.d.). A critical instance case study occurs where one or more sites are examined for either the purpose of examining a situation of unique interest with little to no interest in generalizability or to call into question or challenge a highly generalized or universal assertion. This method was chosen because it is useful for answering cause and effect questions. The research presented here will determine if the pedagogy utilized with the honors student resulted in an effective final project. Ultimately, in designing the study, researchers need to make explicit the questions to be explored and the theoretical perspective(s) from which they will approach the case. There are three most commonly adopted theories: Organizational Theories, Social Theories, and Individual Theories. Individual theories focus primarily on the individual's development, cognitive behavior, personality, learning and disability, and interpersonal interactions of a particular subject (Case Studies, n.d.). Finally, this paper undertakes an approach rooted in the Individual Theories.

2.3 Case Studies as Educational Appliances

What types of educational appliances do case studies provide for teachers? The 1950s marked the dawn of a new era in case study research, namely the utilization of the case study as a teaching method. "Instituted at Harvard Business School in the 1950s as a primary method of teaching, cases have since been used in classrooms and lecture halls alike, either as part of a course of study or as the main focus of the course to which other teaching material is added" (Armisted 1984). Armisted (1984) looked at eight types of case studies, offered pros and cons of using case studies in the classroom, and offered suggestions for successfully writing and using case studies. If using case studies in the classroom is of interest, further supporting research can found by reviewing Merseth (1991) "*The Case for Cases in Teacher Education",* Boehrer (1990) "Teaching With Cases: Learning to Question. *New Directions for Teaching and Learning*", and Boyce (1993) "*The Case Study Approach for Pedagogists*".

2.4 Honors Program - Defined

The Thiel College Academic Catalog (2011-12), pages 177 and 178, summarized the Honors Program as follows: The goals of the Honors Program at Thiel College are to provide an integrative education designed to enhance critical thinking, to enable students to make connections among disciplines and to promote a world view grounded in the exploration of ideas, ideologies and values. These goals are fostered in an environment of small classes, free intellectual inquiry and close association with professors.

In order to provide an appropriate and challenging educational structure, the Honors Program provides a core set of courses that is separate from the College's general education requirement. This core is described in the Honors Program course offerings. It includes some freshman-level classes that are variations of college-wide courses, and a unique sequence of courses during the sophomore and junior years designed specifically for Honors Program members. Participation in the Honors Program permits a combination of the Honors requirements with any academic major with the exception of education.

The honors student course requirements can be found in the appendix (section "Honors Program Requirements").

2.5 Honors Project Requirements

The capstone research project for Honors Program students is **HON 322—Interdisciplinary Course IV: Independent Project** and is worth two credit hours. Students apply the work done in Honors Interdisciplinary Course III by choosing a project related to their own special interests and work independently with the course professor and a mentor. The project integrates library research with a student's own original contributions, which culminates in a public presentation of the project.

The student(s) final project is evaluated using four different criteria:

1. Mentor's evaluation of paper.
2. Mentor's evaluation of project process.
3. Second reader's evaluation of paper.
4. Mentor's or Director's evaluation of oral presentation.

Finally, if the student needs to purchase supplies, the Thiel College Honors Program reimburses students up to $150.

2.6 Teacher Requirements

Per a discussion with the director of the Honors Program, mentors have four major responsibilities: grade the final paper, teach the student how to conduct a research project, teach any other required curriculum, and create a meeting schedule between the mentor and the student. Thus, the teacher/mentor graded the paper according to the supplied grading rubric discussed in section 5 of this paper. Additionally, the teacher/mentor created a meeting schedule that is also discussed in section 3.4. Also, the teacher/mentor instructed courses on all three phases of this project; however, only one phase (Phase I – Usability Testing) is discussed in detail. Finally, the teacher/mentor assisted the student with conducting the research project. To do this, the teacher supplied the student with three required readings:

1. Theoretical Frameworks in Qualitative Research by Anfara & Mertz (2006). Only the introduction was assigned.
2. The literature review of a step-by-step guide for students by Ridley (2008). Only Chapters 2, 3, 6, and 8 were assigned.
3. Reading and understanding research by Locke & Silverman (2004). Chapter 3 and Appendices B and C were assigned.

3. METHODOLOGY

The field of Web Design and Usability has a step-by-step approach that has been identified as a method to successfully solve a given problem.

The following section encompasses the erudite process of teaching students how to conduct a web usability study.

3.1 Course Organization

To teach the honors student, the teacher explained that there is a process to be followed in order to provide the best experience for an Internet user and increase business on websites. The process of developing a web application was explained to the student and the student was advised to gather a brief background on the factors that make an interactive web experience: the Internet, databases, and HTML. The teacher also explained that a usability project was divided into three distinct phases of developing a web application: creating a prototype, defining a technical structure, and designing the interface. To aid the student's research, the teacher provided a listing of suggested readings, which included insights from Steve Krug, a web usability author, and Carolyn Snyder, a paper prototyping author, because this project focused heavily on web usability testing and the prototyping process. Due to the scope of the project, the student only created a paper prototype instead of a digital version. Finally, to ensure that the project would be completed on time, the teacher created a course meeting and due date schedule (section 3.4 and appendix).

3.2 Teaching Usability Study Courses

Phase I of the student project consisted of three stages: Analysis, Prototyping, and Usability Testing. The analysis stage is very important because it involves making an evaluation of the current site. To perform this stage, a site visit is normally essential. The objective of the site visit is to learn how the users interact with the interface. Specifically, this student wanted to learn if a user can effectively conduct a search within the current Community Calendar page. In order to reach the objective of the site visit, the student scheduled a time for the site visit. Conducting a site visit can done by interviews, surveys, video, or a think-out-loud session. This student chose to conduct an interview. Once the site visit was completed, the student formulated a site visit report with the collected data. By summarizing the site visit report, the student found that calendar event search results were simply a long list of text and only allowed the user to search for event by date. The current design did not give the user the option to customize and narrow the search or choose how to display the results (ex: monthly calendar form).

Steve Krug's (2006) book titled, *Don't Make me Think!,* honors Amazon.com for their search approach. He highlighted how most book sites prompt the user to choose from a keyword category (title, author, keyword), while Amazon.com simply allows the user to type any keyword and the search generates results. After analyzing the current web page, undertaking the site visit, researching other community calendar event web site pages, and reading Krug, the student realized that the current web page did not meet usability standards. This prompted the student to design a clearer and more professionally visual site to redesign the current Community Calendar page. Next, the student submitted a plan to redesign the Community Events page, which reduced the user time needed to find an event and resulted in increased efficiency and user confidence.

When working with the web development process, a significant term is "prototype." A prototype is an illustration of a project concept (Houde, 3). A prototype allows the user to put his or her ideas in a form that can be viewed tangibly. Many industries have varying ideas from which prototypes are formed. Materials such as paper, pencils, scissors, glue, website screenshots, Post-its, index cards, manila folders, printed objects, design software, and HTML code are used to produce a prototype in the computer science discipline (Medero, 2007).

Troy Janisch, president and founder of Icon Interactive, lays out four qualities that can be evaluated through designing prototypes. These qualities are: navigation and flow, content, layout, and functionality or interactivity. Navigation and flow is observed through user input that is based on how organized and natural the site and labels are. The effectiveness of a site's content and layout, such as the writing approach used and scarcity of information or too much clutter, can be measured through prototypes. Prototypes are also valuable to determine what functions are beneficial and enhance the interactive experience (Janisch, 2007). As represented by its qualities, a prototype helps to define a professional, appealing site.

Following the analysis stage, the student began the prototyping process. It should be noted that it is common to produce multiple prototypes, and in fact, this is often necessary in order to fully meet user or client needs. This student project only included one prototype, which was simply created on notebook paper by using colored pencils. Medero (2007), an interface designer for the Linguistic Data Consortium at the University of Pennsylvania, addressed the idea that many designers think hand prototypes are not taken seriously. He explained that designers should take advantage of the simplicity of hand prototypes because it is less intimidating than a formal, high-tech sample. Medero (2007) recommended paper prototyping to "lighten the mood and engage a more diverse group," especially during usability testing.

Krug (2006) offered many recommendations for usability and effective web page design. For example, Krug discussed the label for a search button. He stated that it should simply be labeled "Search." He emphasized how easy it is to confuse a user with the label of any button, but especially the search button. The word "Search" is more effective than "Find," "Quick Find," "Quick Search," or "Keyword Search" (Krug, 67). Krug (2006) also provided further recommendations for making the experience easier for a user on keyword searches and suggested that keyword search should not demand case-sensitive words or insist on punctuation. Krug referred to this as "punishing me for not doing things your way." For example, with credit card and social security entries, the user should never have to worry about following a certain format (Krug, 164).

Drawing on these and other recommendations from Krug, the student revised the notebook prototype. The teacher reviewed the new hand-drawn notebook-paper-sized prototype and gave the student permission to create a poster-sized prototype. The student made the poster-sized prototype using poster board and removable screenshots of every individual element of the current homepage and Community Calendar page. The elements of the two pages were adhered to the poster board using Velcro, which allows all elements to be moved or arranged, as the client desires. Finally, paper prototypes have a great influence on improving the final product. Within different usability testing stages, the prototypes must be rearranged or adapted. Paper prototypes allow for more time on actually moving the improvement process along rather than spending hours editing code.

Throughout the explanation of a prototype (second stage), the term "usability testing" was used frequently and is done periodically throughout designing prototypes. Krug (2006) defined *usability* as the process of being certain that "…something works well: That a person of average (or even below average) ability and experience can use the thing…for its intended purpose without getting hopelessly frustrated" (5). Carolyn Snyder, a well-known author of prototyping, stated, "For much of its history, paper prototyping has been a tool clenched firmly in the hand of the academic researcher or usability specialist. Like any useful tool, though, its greatest potential can be reached by placing it in the hands of the non-specialist along with instructions for its proper use" (Janisch, 2004). Although Snyder focused on paper prototyping, all prototyping can be considered from this viewpoint. The purpose of creating a prototype is to follow through with a process that consists of collecting feedback from people who are prompted to perform tasks by using the given prototype (whatever prototype that may be—paper or digital). It is certain that a prototype is most useful during the usability testing stage of the development process.

According to the Guide to Planning and Conducting Usability Tests (University, 2008), there are four types of tests to choose from and they can be completed at any point of the development process. These tests are: explanatory, assessment, validation, and comparison. The explanatory test is executed by providing the user with a simulation of a webpage and then the user is asked to explain his or her thoughts on what the page elements do and what he or she would like to gain from the page. The assessment test is where the user is prompted to complete a task after the prototype is basically perfected and is only tested for effective implementation. The validation test is used when certain timing standards are desired to be reached in order to measure how well all the website features merge. As for the comparison test, this can be conducted at any stage of the design process and is used to compare multiple design ideas by conducting the same task using each design. In this case study, the prototype was ultimately tested using a combination of the explanatory and assessment methods. Also, the usability testing procedures conducted in this project replicated the suggestions by Jeffery Rubin, author of the *Handbook of Usability Testing* (Rubin, 2011).

The first step in usability testing is to professionally and effectively introduce the usability test to the test participants. This means that the proctor of the test needs to explain the purpose of the test using a pre-written script and reassure the participants that they were not being tested (University, 2008). Next, similar to Krug's script example printed in *Don't Make me Think!,* the participants should be asked to simply describe their initial thoughts of the site (150). Next, the participants should be asked to search for certain things on the site that the client desires. This can be any number of elements resulting in one or many different tests for the user to complete. A key aspect to remember during the testing is for the proctor to remain only as an observer until the participant asks for assistance, and then the proctor should encourage the participants to think out loud (University, 2008). The proctor should take notes on the notes sheet and write down his or her observations. If the proctor has questions for the participant that need clarification, the questions should be noted in a questions page left for after the test is completed. Finally, the proctor should debrief the participant by obtaining a clear understanding of the participant's overall experience and take notes of any other comments that the participant wants to share.

3.3 Student Project Requirements

The director of the Honors Program only stated one requirement for the report, which was the report needed to be at least 20 pages in length. All other student project requirements were set by the teacher/mentor. Therefore, the teacher set the minimum length to the required 20 pages. Additionally, the teacher also required the student to write in APA style versus the typical MLA style used in a bachelor's degree program. This change was implemented so the teacher could reinforce how to write research papers to prepare the student for writing in a master's degree program. Finally, the teacher required the student to follow a sample outline and expand the paper as needed. Below is the outline provided:

Introduction
Business Case
Site Visit
Prototype
Usability Test
Recommendations
References
Appendix

3.4 Schedule

As advised by the director of the Honors Program, the teacher/mentor created a meeting schedule for the student. The schedule played a major role in ensuring that the project would be completed on time. The appendix (section "Project Schedule") includes a detailed spreadsheet of meetings, lectures, and due dates.

4. CASE STUDY OF STUDENT USABILITY PROJECT

A case study can be defined as the collection and presentation of detailed information about a particular participant or small group, frequently including the accounts of subjects themselves (Case Studies, 2012).

4.1 Usability Testing

To begin the usability testing, the teacher had the student create the **Usability Test Script**; the script for this project follows.

Hi, my name is [Name], and I'm going to be walking you through this session.

Let me explain why I have asked you to do this session today. I am testing a website of which I am redesigning a page and adding a database feature and I would like to see what it is like for an average person to use, rather than relying solely on my perception since I am working close to the project.

I want to make it clear right away that we are testing the site, not you. You can't do anything wrong during this session. I want to hear exactly what you are thinking, so please don't worry that you're going to hurt my feelings. I want to improve the site, so I need to know honestly what you think. As we go along, I am going to ask you to think out loud, to tell me what's going through your mind so I can take note of an average user's thinking process and perception.

If you have questions, just ask. I may not be able to answer them right away, since I am interested in how people do when they don't have someone sitting next to them, but I will try to answer any questions you still have when we're done.

If you would, I am going to ask you to sign something. It simply says that we have your permission to use the results from your session for this project. The information will only be seen by me and my honors project mentor.

Do you have any questions before we begin?

DEMOGRAPHICS
Before we look at the site, I'd like to ask you just a few quick questions to get to know you and how you currently use the Web.
Q1: First, what is your academic field of interest?
Q2: What exactly does your field of interest do?
Q3: Now, roughly how many hours a week would you say you spend using the Internet?
Q4: How do you spend the time you pass on the Internet?
Q5: Do you have any favorite websites?
Q6: What is the purpose of the site?
Q7: What do you like about this site?
Q8: On a scale of one to five (one being not often at all and five being very often), how often do you use search engines to browse the Internet for what you want?
Q9A: What search engine do you prefer?
Q9B: Why do you prefer the search engine you mentioned in the previous question?
Q10: Have you ever used Bing to search for events?
Okay, awesome, thanks! We are done with the background questions and we can start looking at the site.

This test will be performed using paper prototypes rather than digital so the interaction is limited. If at any time you would naturally type something please use this sheet of paper to write down what you would type. And, anything below this (point) footer is an element that would replace another element upon interaction like a click.

HOMEPAGE TESTING
Q11A: First I'm just going to ask you to look at this page and tell me what you think it is?
Q11B: What strikes you about it?
Q11C: What you think you would click on first? And, again as much as possible, it will help me if you can try to think out loud.
Q12: What would you do to find local community events?
Q13: Would you see it as beneficial to swap the search for local events button with the events box?

PROMPTS TESTING
P1: Okay, moving onto actually using the site. Now that we are on the events page, search for events in the category of education for the month of March and to print the results in calendar view
Q14: If you were to make the calendar feature better, what would you suggest?
Q15: Having the Print Results button at the bottom or top of the page, would you prefer it to be left, center, or right aligned?

P2: Okay, great! Next, what would you do to search for local bowling opportunities for the day of March 31st?
Q16: Awesome! Now, just a few questions to sum everything up…What is your overall satisfaction with the search experience on a scale of one to five? Lastly, do you have any questions that I couldn't answer during the test that you would like answered now?
Q17: And, what would you suggest to improve the homepage?
Q18: What would you suggest to improve the events page?

When the usability testing is complete, an analysis of the participant data collected is to be conducted. The easiest was to organize the data is to base the organization on the format of the notes sheet that was used to write down observations during testing. The gathered data from the one to many tests conducted should be analyzed for patterns of design dissatisfaction, wording and labeling confusion, button placement, and in this project case - overall effectiveness of the customized search database. At this point, the data gathered though the conduction of the usability test should provide significant insights to the designer/developer to create a redesigned site that the client will benefit from.

4.2 Reporting Usability Testing Results

The student analyzed each question asked during the usability test, an example of how the student organized and then wrote up the collected data follows.

Q18: What would you suggest to improve the events page?

Figure 1: Prompts Testing - Question 18 – Events Page Suggestions

Participant	Q18
1	display recent searches
2	NONE
3	bigger font in footer
4	make drop-down calendar bigger; add select all-print all option
5	add Select All option
6	option to search with Calendar or List View results
7	move select box to left under photo
8	make March 2011 a link; move Search button

Many of the suggestions for this question were already mentioned through responses to a previous question. For example, adding a Select All printing option, viewing the results in calendar view, moving the Search button, and making the "March 2011" calendar title a link were all repetitive concerns. Others addressed displaying recent searches made by other users, increasing the size of the drop-down calendar and the font of the footer, and moving the select boxes to the left, under the event thumbnail image.

4.3 Case Study - Student Paper

The table of contents created by the student is below.

5. COURSE ANALYSIS

The final research project required of the honors students is part of **HON 322—Interdisciplinary Course IV: Independent Project.** The director of the Honors Program distributed the grading rubric to the Honors Project Students, Mentors, and Second Readers. The grading rubric also clearly defined the grading scale for the Four Evaluation Areas mentioned in section 2.5 ("Honors Project Requirements"), and it properly assessed the student for the Honors IV Course. The grading rubric document can be found in the appendix (section "Grading Rubric").

6. CONCLUDING REMARKS

Two empirical research issues limit this study. The first issue is that the study was conducted on and by a single student; further studies could be done involving a greater number of students and on a diverse population. Secondly, the experience of a single researcher may raise concerns because a single interpretation may be subjective and possibly different outcomes could have resulted if conducted by several or different researchers.

By reflecting on this case study, the researcher learned that teaching pedagogy to one student in the manner presented in the paper proved successful. The student successfully completed the research project with an "A" grade and far surpassed the expectations of the teacher. The student effectively applied the research theories and computer information systems curriculum taught in the final project.

The researcher suggests that further research should be conducted with a group of students. Additionally, the researcher would conduct a follow-up survey at the end of the course. A follow-up survey would add the quantitative elements, making the study more thorough and complete.

In closing, the purpose of writing cases and sharing them with others is to share experience beyond the confines or limits of geography (GTTP, 2012). This case study paper presented more than just a description of sharing one experience; it shared applicable research and effective pedagogy to use in the 21st Century classroom.

7. REFERENCES

Anfara, V. A., & Mertz, N. T. (2006). Introduction. *Theoretical Frameworks in Qualitative research* (pp. xiv - xxxii). London: SAGE.

Armisted, C. (1984). How useful are case studies. *Training and Development Journal, 38 (2),* 75-77.

Boehrer, J. (1990). Teaching with cases: Learning to question. *New Directions for Teaching and Learning, 42* 41-57.

Boyce, A. (1993) *The Case Study Approach for Pedagogists.* Annual Meeting of the American Alliance for Health, Physical Education, Recreation and Dance. (Address). Washington DC.

Case Studies. (n.d.). *Welcome to Writing@CSU*. Retrieved July 12, 2012, from http://writing.colostate.edu/guides/research

GTTP. (2012). *HowToWriteAGoodCase.pdf*. Retrieved June 8, 2012, from www.gttp.org/docs/HowToWriteAGoodCase

Houde, S. & Hill C. *What do prototypes prototype?* Retrieved February 12, 2011 from http://www.sics.se/fal/kurser/winograd-2004/Prototypes.pdf

Janisch, T. (2004, June 1). *How good does your web site look on paper*? Retrieved February 12, 2011 from http://evolt.org/node/60331/

Kharbach, M. (n.d.). The 21st century pedagogy teachers should be aware of. *Educational Technology and Mobile Learning*. Retrieved July 12, 2012, from http://www.educatorstechnology.com/2011/01/21st-century-pedagogy-teachers-should.html

Krug, S. (2006). Don’t make me think!: A common approach to web usability. Berkley, California: New Riders Publishing.

Laudon, K., & Traver, C. (2010). E-Commerce 2011: business, technology, society. (7th Edition ed.). Upper Saddle River, NJ: Pearson - Prentice Hall.

Locke, L. F., & Silverman, S. J. (2004). *Reading and understanding research* (2nd ed., pp. 29-58). Thousand Oaks, Calif.: Sage Publications.

Medero, S. (2007, January 23). *Paper prototyping*. Retrieved February 12, 2011 from http://www.alistapart.com/articles/paperprototyping/

Merseth, K. K. (1991). *The Case for Cases in Teacher Education.* RIE. 42p. (ERIC).

Moncur, L. (n.d.). *Quotation details.* (Quotation #25848). Retrieved from http://www.quotationspage.com/quote/25848.html

Pedagogy. (n.d.). In *Oxford English Dictionary*. Retrieved from http://dictionary.oed.com

Ridley, D. (2008). *The literature review a step-by-step guide for students* (Repr. ed., pp. 16-27, 28-33, 80-88, 117-129). Los Angeles [u.a.: SAGE.

Rubin, J., & Chisnell, D. (2011). *Handbook of Usability Testing How to Plan, Design, and Conduct Effective Tests.* (2nd ed.). Hoboken: John Wiley & Sons, Inc..

Thiel College. (2011). Academic-Catalog-2011-12.pdf. *Thiel College*. Retrieved July 9, 2012, from www.thiel.edu/academics/academic_catalog/pdf/Academic-Catalog-2011-12.pdf

World Bank. (n.d.). *The learning pyramid*. Retrieved from the World Bank Web sitesiteresources.worldbank.org/DEVMARKETPLACE/Resources/Handout_The-LearningPyramid.pdf

Yin, R. K. (1993). Advancing Rigorous Methodologies: A Review of 'Towards Rigor in Reviews of Multivocal Literatures.' *Review of Educational Research, 61,* (3).

Appendices

Honors Program Requirements

The core of required courses, which substitutes for the general College Integrative Requirement, consists of the following:

Course Number	Course Name	Course Credit Hours
HON 115	History of Western Humanities I	4 CH
HON 111	Oral and Written Expression I	3 CH
HON 112	Oral and Written Expression II	3 CH
HON 125	History of Western Humanities II	4 CH
HON 132	Interpreting the Jewish and Christian Scriptures	3 CH
HON 212	Interdisciplinary Courses I	3 CH
HON 222	Interdisciplinary Course II	3 CH
INDS 210	Science and Our Global Heritage I	4 CH Choose 1
INDS 220	Science and Our Global Heritage II	
One	natural or physical science laboratory course	4 CH
HON 312	Interdisciplinary Course III	2 CH
HON 322	Interdisciplinary Courses IV	2 CH
Foreign Language competency:	Two semesters (check for possible exemption)	0-6 CH
Mathematics competency	See Below	0-4 CH

Mathematics competency:
For the BA degree: pass the mathematics placement test at the pre-calculus level or earn a grade of C- or higher in any math course except MATH 011 or MATH 121. 0-4 CH
For the BS degree: pass the mathematics placement test at the calculus entry level or earn a grade of C- or higher in MATH 141 or any calculus course. 0-4 CH

Writing Intensive Course (WIC) requirement: Satisfactory completion of five WIC courses, not more than three of which can be in the major.

What are Writing Intensive Courses? A student at Thiel College must completed 5 WIC courses to graduate. WIC courses can be completed in any combination of major, minor, core and elective courses that are designated as WIC. However, to fulfill the requirement no more than three courses can be in the same discipline.

Course Offerings
HON 111—Oral and Written Expression I (3 CH) This course for freshman Honors Program students integrates fundamental components of oral and written expression by focusing on similarities and differences between the two forms, emphasizes an introduction to learning in the liberal arts tradition, a comparison of academic and professional disciplines, critical thinking skills, ways of identifying and testing evidence and hypotheses, and the use of primary sources in writing and speech production. Offered every fall.

HON 112—Oral and Written Expression II (3 CH) This course is a continuation of HON 111. It refines the skills introduced in HON 111 and provides further opportunities for formalizing the components of oral and written expression and multidisciplinary learning. The course enables students to refine their critical thinking and problem solving skills in their oral and written analyses of the various subjects and styles of academic writing and oral expression. Students master documentation of sources and extend their knowledge of research skills and oral and written delivery modes. Offered every spring.

HON 115-125—History of Western Humanities I & II (4 CH) This two-semester sequence surveys material and cultural history from antiquity through post-modernism. The interdisciplinary approach encourages students to discover connections between historical periods and artistic style periods in the areas of philosophy, religion, art, architecture, music, literature and theater. Students are encouraged to reflect critically on the connections they discover and find relationships to their own lives and experiences. This discovery/reflection model helps provide students with a context by which to understand the values of humanity both as they are expressed in the past and as they are expressed in their own lives. HON 115 offered every fall and HON 125 Offered every spring. (HON 125: WIC)

HON 132—Interpreting the Jewish-Christian Scriptures/Honors (3 CH) The purpose of this course is to assimilate the content, understand the structure and wrestle with the meanings of the writings included in the Judeo-Christian Scriptures. As an Honors course, a minimum amount of time will be spent on lectures that rehash either the content of the text or the biblical material. Class sessions will focus on discussion, centering upon questions, problems and insights precipitated by the readings. A basic assumption of the course is that participants will take responsibility for a thorough reading of the text and related biblical material in preparation for class. (WIC) Offered every spring.

HON 212—Interdisciplinary Course I: Identity (3 CH) The first semester of a year-long integrative course. Through a consideration of the concept of identity, students will participate in a variety of ways to gain skills in problem-solving, speaking, receptiveness to critical discussion of ideas, value centered decision-making, self-reflection and self-discovery. Offered every fall.

HON 222—Interdisciplinary Course II: Identity (3 CH) A continuation of HON 212. Offered every spring.

HON 312—Interdisciplinary Course III: Creativity (2 CH) This course focuses on the topic of creativity in its broadest sense, as a concept relating to an overall approach to life experience, and also its specific applications to the arts, sciences and humanities. Offered every fall.

HON 322—Interdisciplinary Course IV: Independent Project (2 CH) In this course students apply the work of Honors Interdisciplinary Course III by choosing a project related to their own special interests and working independently with the course professor and a mentor. The project integrates library research with students' own original contributions, culminating in a public presentation of the project. (Students who study abroad may fulfill these requirements by completing a project following their international experience.) Offered every spring.

Project Schedule

Week #	Due at Meeting Time	Mon	Wed	Fri	Sat
1 = 1/9 - 1/15					
2 = 1/16 - 1/22					
3 = 1/23 - 1/29	Discuss New Project	Meeting 11:00		Meetin g 11:00	
4 = 1/30 - 2/5	Title Thesis Current Site Prototype -Tables -Field Names	Meeting 11:00			
5 = 2/6 - 2/12	Introduction	Meeting 11:00 Teach Usability Pro Mgt			
6 = 2/13 - 2/19	Revised Intro	Meeting 11:00 Teach Phase I Phase II			
7 = 2/20 - 2/26					
8 = 2/27 = 3/5	Intro, Phase I & II Due by Midterm break				
9 = 3/6 - 3/12					
10 = 3/13 - 3/19	1. Review Paper - Intro, Phase I & II 2.Teach - Phase III	Meeting 11:00 Teach Phase III			
11 = 3/20 - 3/26					
12 = 3/27 - 4/2					
13 = 4/3 - 4/9	Conclusion				
14 = 4/10 - 4/16	Presentation				Present
15 = 4/17 - 4/23	Revisions				
16 = 4/24 - 4/30	Paper Due		Paper Due		

<u>Grading Rubric</u>

Designed by Dr. Beth Parkinson

TO: Honors Project Students, Mentors, and Second Readers
RE: Grading for Honors IV Course

I. General Grading Scale

A certain percentage of the total grade will be allotted to each of four areas of evaluation.

1. Mentor's evaluation of paper = 45%
2. Mentor's evaluation of project process = 25%
3. Second reader's evaluation of paper = 15%
4. Evaluation of oral presentation = 15%

200 points represents the total points available for the course. Converting these points into percentages:

1. Mentor's evaluation of paper = 90 total possible points
   81-90 = A range 63-71 = C range
   72-80 = B range 54-62 = D range

2. Mentor's evaluation of project process = 50 total possible points
   45-50 = A range 35-39 = C range
   40-44 = B range 30-34 = D range

3. Second reader's evaluation of paper = 30 total possible points
   27-30 = A range 21-23 = C range
   24-26 = B range 18-20 = D range

4. My evaluation of oral presentation = 30 total possible points
   27-30 = A range 21-23 = C range
   24-26 = B range 18-20 = D range

Grades for each of the four areas will be given as <u>points</u>. The total points will be added to determine the final letter grade.
180-200 = A range 140-159 = C range
160-179 = B range 120-139 = D range

<u>IMPORTANT NOTE TO STUDENTS</u>: Attending all class meetings is a course requirement. If you cannot attend a class meeting, call or leave me a voice mail prior to the meeting. You are responsible for knowing all information given out at all meetings. You are allowed one absence without penalty. Five points will be deducted from your semester point total for each absence after the first one. Only <u>dire</u> circumstances will be granted an exception.

ALL PROJECTS WILL BE ASSEMBLED IN A BINDER AND KEPT ON DISPLAY IN THE HONORS PROGRAM CENTER.

II. Grading Criteria for the Four Evaluation Areas

1. <u>Mentor's evaluation of paper - evaluation criteria</u>

   a. Format in accordance with agreed-upon discipline style (e.g., MLA, APA, journal publication style of a particular discipline). Includes general format, citations in the text, references, correct Internet citations, etc.
   b. Well-written introductory section, appropriate lead-in to topic.

c. Clearly explained topic or thesis.
d. Good general organization of material - logical flow of ideas.
e. Adequate review of relevant literature or related research findings.
f. Adequate transitions between various ideas and sections of the paper.
g. Good explanations of terminology used.
h. If charts, graphs, tables, or figures are used, are they easily understood and well set-up?
i. Spelling, grammar.

"a" through "i" are criteria which are applicable throughout the entire paper. "j" through "n" are applicable specifically to the section of the paper presenting a creative, original, "hands on" element.

j. Is your position, argument, or solution clearly stated?
k. Is there a clear distinction between factual information and opinion?
l. Use of information cited earlier in the paper to support ideas - supportive evidence.
m. Are the positions, arguments, data analyses, and/or solutions logical, feasible, workable, realistic?
n. Do you present your viewpoint in a convincing manner?
o. Ending of paper - Whether the ending is a summary, conclusion, or overview, is the ending well integrated into the paper, and not just a few sentences "tacked on?"

2. Mentor's evaluation of project process - evaluation criteria

One of the purposes of this course is to give students the opportunity to work with a faculty mentor. This is intended to be an integral part of the research process, providing a valuable learning experience.

This portion of the grade is not dependent upon the ease or difficulty of executing the project, since it is expected that everyone encountered some difficult periods throughout the semester. The evaluation is based on the following:

a. Appointments made and kept by the student, or rescheduled if he/she could not keep the appointment.
b. The student turned in promised work on time, or made satisfactory arrangements for an extension.
c. *The student and mentor interacted in a real working relationship, rather than the student doing an independent project (* an especially important criterion).
d. Student and mentor maintained a dialogue concerning suggestions and possible improvements for the paper.
e. The student took the initiative in reviewing the relevant literature, using the mentor as a guide, rather than a source for material.
f. The student took the initiative in writing the creative, original, "hands on" portion of the paper, using the mentor as a guide, rather than a source for ideas.

3. Second reader evaluation of the paper - evaluation criteria

This evaluation will use all the criteria used by the mentor (p. 2 "a" through "o") in evaluating the paper.

4. Evaluation of the oral presentation - evaluation criteria

a. Length of presentation close to the 12-15 minute time limit.
b. Good introduction to the topic.
c. Clear explanation of theme or topic.
d. Citation of relevant literature related to the topic.
e. Good transitions between ideas.
f. Selective use of materials - inclusion of important information, exclusion of less important material not needed for the oral presentation.

g. If audio-visual aids were used, were they well integrated into the presentation, a useful addition? Were charts or tables easily understood?
h. Was the creative, original, "hands on" portion clearly presented and easily understood? Were your own ideas clearly delineated from theoretical or empirical data?
i. Good eye contact with audience.
j. Good delivery style - not too fast or too slow, good vocal variety, clear speaking.
k. Notes used as guidelines only, rather than reading from them continuously.
l. Good ending of presentation.
m. Answering questions from the audience thoughtfully and competently.

An Exploratory Study of the use of Video as an Instructional Tool in an Introductory C# Programming Course

Jason H. Sharp
jsharp@tarleton.edu

Leah A. Schultz
lschult@tarleton.edu

Computer Information Systems
Tarleton State University
Stephenville, TX 76402, USA

Abstract

This study examines the background of introductory programming concepts and the use of video as an instructional tool. Thirty-five students in an introductory C# class were administered a survey to report data on demographics, usage on video, and opinions about the video. Students were in online and face to face sections of the class. Data were analyzed to determine how students used the videos and to determine if there were differences between the two groups. Multiple aspects analyzed show no difference in use of the online video between face to face and online students.

Keywords: programming, video, instructional tool, video lectures, e-lectures, online learning

1. INTRODUCTION

Programming, regardless of the specific language utilized, how the programming course is taught, whether the language is procedural or object-oriented, has traditionally been one of the more difficult courses for undergraduate students in an information systems (IS) degree program (Fincher 1999; Jenkins, 2002; Lahtinen, Ala-Mutka, & Jarvinen, 2005; Milne & Rowe, 2002; Robins, Rountree, & Rountree, 2003). Couple this fact with the move toward online education and the problem is exacerbated (Butler & Morgan, 2007). The purpose of the following study is to examine the use of video as an instructional tool to illustrate key concepts in an introductory course in C# programming delivered both face-to-face and online and to compare and contrast findings between the two modes of delivery.

2. LITERATURE REVIEW

There has been an ongoing debate regarding the place of programming in the curriculum of undergraduate IS programs (Topi, Valacich, Wright, Kaiser, Nunamaker, Sipior, & de Vreede, 2010), the primary language that should be used to teach programming (Jenkins, 2002; Russell, Russell, Pollacia, & Tastle, 2010), and how to best teach programming (Fincher, 1999; Jenkins, 2002) and a considerable amount of related research. Similarly, there is an abundance of research related to the use of video, in some form or fashion, for teaching and learning in both face-to-face and online modalities across multiple disciplines in both education and industry. However, a search of the literature yielded no specific research related to teaching programming in an online environment or to providing support for traditional face-to-face courses using video as an instructional tool to assist students in this traditionally difficult subject area.

Programming in the Curriculum

The issue of whether or not programming should be included in the core curriculum and what type of programming it should be is not a new debate (Cain, 1991; Gotwals & Smith, Jr., 1995; Gotwals & Smith, 1993). Recently, IS 2010, Curriculum Guidelines for Undergraduate Degree Programs in Information Systems (Topi et al., 2010) has removed application development from the set of required core courses. The authors, however, assert that "it is important to understand that although application development is not included in the core, it has not been removed from the IS program, and the task force acknowledges that a strong case can be made for inclusion of programming, computational thinking, data structures, and related material in an IS program" (p. 27). The suggestion is to offer application development as an elective and that programs that want to implement a sequence of programming courses can do so.

Programming Language to Use

In regard to what programming language to use, Jenkins (2002) asserts that "there is scant solid evidence that any language is any better or any worse than any other, and the choice continues to be driven largely by the 'flavour of the month' in industry" (p. 55). There appears to be wide agreement among IS educators that the purpose of an introductory programming course, at least, is to teach students to program more so than to teach them a specific language. It is, however, difficult for students to make such a differentiation. Many get caught up in the details of the syntax, while missing the more important higher level concepts. Another variable in this discussion is that languages created to teach programming, but not currently used in industry are avoided in the attempt to recruit students based on the fact that the program teaches languages currently used in the "real world" (Jenkins, 2002).

Russell et al. (2010) conducted a study of the programming languages used in information systems and computer science curricula. Their goal was to determine if a particular language was better suited for the sequence in which the programming course was offered (first-course, second-course, third-course) based upon curriculum type (CIS, MIS/IS, CS, IT). In their study they examined Visual Basic.Net, Java, and C#. Their results indicated that "only for the second programming course did program type seem to influence the programming language used" (p. 10) and that a Windows-based interface was primarily used for the first-course and second-course, while a Web-based interface was primarily used for the third-course.

Teaching and Learning Programming

Jenkins (2002) asserts that "at the moment the way in which programming is taught and learned is fundamentally broken" (p. 53). He goes on to say that "few computing educators of any experience would argue that students find learning to program easy" (p. 53). In light of these statements, Jenkins summarizes several potential reasons that students have such a difficult time learning to program. These include commonly cited reasons such as lack of aptitude or cognitive factors including learning styles and motivation. Rountree et al. (2003) also provides an excellent review of the literature in the general context of cognitive psychology as it relates to learning how to program, specifically: the task, mental models and processes, and novice capabilities and behaviors.

Jenkins (2002) goes on to cite another reason why students find programming difficult which he calls "life skills". These particular skills are not as commonly cited by IS educators, but may include transitioning to college life and such intangibles as being away from home for the first time, struggling to develop new friendships, and having to manage finances and personal and study time for the first time. It is within this time of "transition" that many students encounter their first experience with programming. Jenkins asserts, "this is difficult enough material to master when a student is well settled, but departments' insistence on teaching this during a period of transition can only increase the difficulty" (p. 55).

Jenkins (2002) also suggests that adding to the difficulty of programming is the fact that it consists of multiple skills and multiple processes. This "hierarchy" of skills begins with lower level skills such as the basics of syntax and progresses to higher level skills such as semantics, structure, and style. Related to processes, the programming student must be able to translate specifications into an algorithm, consider if these specifications resemble something from past experience that can drawn upon, and finally must convert the algorithm to actual code. It is essential, therefore, that a student master all three processes. As Jenkins put it, "there is little point in lecturing to students on syntax when they have no idea of where and how to apply it" (p. 55).

IS educators are all too familiar with the student who attends every class meetings, appears to follow the lectures, seems to grasp the program examples, but is "incapable of writing their own program. They have not mastered all the processes; they can code, but they cannot produce an algorithm" (Jenkins, p. 55). Lahtinen et al. (2005) echo these sentiments by stating, "the biggest problem of novice programmers does not seem to be understanding of basic concepts but rather learning to apply them" (p. 17).

Fincher (1999) states that the approach to learning to program prior to the emergence of Computer Science (CS) and Computer Information Systems (CIS) as distinct disciplines was geared toward learning the "languages and techniques of programming for a specific purpose" (p. 12a4-1). As these disciplines have matured, however, she suggests that "programming is not taught as a process separate from purpose. We no longer teach programming in order to get the computer to *do* something, but as a transferable skill in its own right" (p. 12a4-1). She then goes on to summarize and evaluate four approaches for teaching programming: (1) the "syntax-free" approach, (2) the "literacy" approach, (3) the "problem-solving" approach, and (4) computation as interaction. Fincher suggested that the differences within each approach lies in how it defines what comprises programming. In terms of commonality, she asserts, "all the approaches have in common the idea that coding is separate from programming" (p. 12a4-4). Fincher concludes by stating that the "debate about what we should be teaching undergraduate computer scientists is not particularly new" (p. 12a4-4); however, "what is new is the questioning of what we are aiming to do in the teaching of programming" (p. 12a4-4). Simply put, it is not "what" is taught, but rather "how" and "why" it is taught.

Video Lectures or E-Lectures

A database search of the terms video-based learning, video-based training, and video-based instruction results in myriad ways video has been used in both educational and professional settings ranging from accounting (Martin, Evans, & Foster, 1995), ethics (Sedaghat, Mintz, Wright, 2011), drug education (Dusenbury, Hansen, & Giles, 2003), learning and motivation (Choi & Johnson, 2005), promotion of student-centered learning (Gainsburg, 2009), and acquisition of technical skills such as suturing and knot-typing for medical students (Dubrowski & Xeroulis, 2005) and block-laying and concreting for distance learners (Donkor, 2011), just to name a few. Suffice it to say, the use of video in some form or fashion is not a new phenomena to the teaching and learning discipline.

The use of video, in some form, within business-related education is not without representation in the research literature. For example, Mintu-Wimsatt (2001) conducted a study between two MBA classes, one delivered in a traditional face-to-face mode and the other delivered using interactive video instruction. Their study indicated that students in the face-to-face course consistently rated the course higher than those in the distance learning course. Ellis and Okpala (2004) evaluated the use of digital technology and software use among business education teachers, specifically the use of digital video cameras and multimedia editing software to edit text, sound, video, computer graphics and animation. They found that younger educators had more of an affinity for incorporating digital video technology into their courses than older educators and that there were some "differences in the use and comfort level of these instructional tools among business education teachers of different ethnic groups" (p. 56).

With the rise in online education the use of and study of various information and communication technologies including video, in its various forms, continues to grow (Katz, 2000). With the explosion of available video on the Internet via sites such as YouTube® (Jones & Cuthrell, 2011), through video content providers such NBC Learn®, and various other video streaming technologies (Hartsell & Yuen, 2006), educators have a wealth of content-rich and in many cases, professionally-edited video to provide to students in both face-to-face and online modes of delivery.

Additionally, through screen recording and video editing software packages such as Adobe Captivate® and Camtasia®, educators know have powerful tools readily available to make their own quality videos. Although the availability of technologies which enable the use of video in the classroom continues to grow, it is still relevant to note the importance of instructional design factors related to the creation of learning environments which implement these technologies (Fanning, 2008).

Terms such as "video lectures" (Brecht & Ogilby, 2008; Geri, 2011; Lents & Cifuentes, 2009) and "E-lectures" (Jadin, Gruber, & Batinic, 2009) are becoming quite common-place in the literature. Jadin et al. (2009) defines an e-lecture as "a media based lecture including an audio or video recording, synchronized slides, table of contents, and optional complementary information (e.g., external links)" (p. 282). Based upon availability and affordability of video-based technologies and the rise in online education, Geri (2011) suggests that "in the coming years, the use of video lectures as a means for distance learning, as well as for supporting traditional in-class learning is expected to increase" (p. 225).

In her article entitled, "If We Build It, Will They Come? Adoption of Online Video-Based Distance Learning" Geri (2011) suggest that they will indeed come, noting that "video lectures offer students a rich learning experience, which resembles traditional in-class learning" and may possess the potential to "increase both student retention and achievements in distance and blended learning environments" (p. 225). Although the study found that "the majority of students prefer attending traditional face-to-face class meetings. Nevertheless, the availability of videos may improve the achievements of all the students enrolled in a course" (p. 231).

Brecht and Ogilby (2008) conducted a study to evaluate the feasibility and effectiveness of a comprehensive teaching strategy based upon video lectures. The authors suggests that video lectures serve two major strategic purposes: (1) they provide additional teaching time to students who may not fully understand material presented in the classroom lecture and textbook, and (2) they allow classroom coverage of more complex and challenging subject materials since basic concepts can be provided via the video lectures and watched outside of class. The study indicated several interesting findings: video lectures helped students raise their course grades, there was a 71.9% reduction in failing grades among students for whom the videos were available compared to students for whom the videos were not available, the creation of videos means that the lectures for the entire semester are available for preparing for the final exam.

Not only does the debate continue about inclusion of programming in the curriculum, what language or languages to teach, and how to teach them, but with the rise in online education and availability and affordability of video-based technology, a new debate arises: that of teaching programming, a traditionally difficult subject, in an online environment and the potential use of video as an instructional tool to support both online and face-to-face courses. As such, we present the methodology for our study, results, and discussion and conclusion in the following sections.

3. METHODOLOGY

Videos

Demonstration videos were created to correspond to each chapter in the programming text. Adobe Captivate® was used to create videos that captured the desktop of the instructor as he completed each programming exercise. Each video focused on key concepts from the chapter and showed students how to program concepts in C# while the instructor narrated the video and added additional explanation to the programming concept. Videos were approximately 20 minutes in length. Videos were then placed in Blackboard for students to view at their convenience.

Participants

In order to examine potential differences in delivery format, students in two sections of introduction to programming classes were given a survey to determine the way they used the videos during the course and their preferences in relation to the videos. Of the two sections, one section was taught entirely online and the other section was taught face to face and was supplemented with material online. Most students in the classes were in the 18-40 age range and most were male. All students were majoring in computer information systems.

Fifteen of the students were in the face to face section and the other twenty were in the online section for a total of thirty-five participants.

Data Collection

A fifteen question survey was administered to both sections of students. Online students were encouraged to take the survey online. Upon completion, students were provided with a code to turn in for extra credit being offered to participants. Students in the face to face section completed the survey on paper and turned the survey in to the instructor during class. They also were offered extra credit for their participation.

In addition to demographic information, students were asked to report their preferences and opinions on the video. Students were also asked to report how much time they spent using the videos and the textbook as well as their opinions on various aspects of the videos such as length, topic coverage, and usefulness. (Appendix A).

4. RESULTS

Hours watching videos and reading the course textbook

Students self-reported the amount of time per week they spent on average watching the online video demonstrations. For all students, the average time spent watching videos was 2.11 hours per week. For online students, the average time spent watching video was 2.46 hours per week and for face to face students, the average time watching videos was 1.58. In order to determine if a difference exists between online and face to face students, an independent t-test was conducted and found no significant difference between the hours spent watching videos by the two different groups of students; $t(31)=1.676$, $p=.070$.

Overall, students spent more time per week watching videos than they did reading their textbook. The mean time spent reading was 1.81 hours compared to the 2.11 hours spent watching videos.

Usefulness of videos

On the question regarding usefulness of the video, 94% of all students rated the usefulness of the videos at 7 or above on a scale from 1 to 10. The two students that reported lower scores of 1 and 2 were in the online and face to face class respectively. A Mann-Whitney U test was conducted between the online and face to face students. The test showed that there was no statistically significant difference between the reported usefulness of the videos between the two groups; $U=118.00$, $p > .05$.

Preference of textbook vs. online videos

Overall, 82.9% of all students surveyed, rated their preference for the videos over the textbook at a level of 7 or above. By group, the preference of textbook versus video was examined using a Mann-Whitney U test. No significant difference was found in the preference of one instructional media over the other; $U=129.00$, $p=.458$.

Knew Instructor

When asked if the videos helped students know their instructor better, 80% of students in both sections indicated at a level of 7 or higher that they did feel as if they knew their instructor through the videos. In order to determine if a difference exists between online and face to face students, a Mann-Whitney test was conducted and revealed there was no significant difference between the responses of the two groups; $U=131.00$, $p=.511$.

5. DISCUSSION AND CONCLUSION

The transition to teaching programming online included many concerns about how to interact with students and to emulate the demonstration aspects of a face to face class in the online environment. To mitigate this problem, the instructor recorded demonstration videos specifically for use in the online sections of the class, never intending to use them in the face to face class. Students who were in the face to face class had access to the demo portion of the class through class lecture and it seemed unnecessary or redundant to include the videos in the online component of the face to face class. In addition, there were concerns that posting demonstration videos online would encourage students to miss class and watch the videos online instead.

However, the results show that the videos were as important to the face to face students as the online students. Students in both sections watched the videos in similar amounts and rated them equally useful between groups.

Surprisingly, the students who saw the instructor for three hours a week were just as likely to report that the videos helped them "know" the instructor as much as online students, many of which had never met the instructor face to face.
When looking at the overall impact of the video on both students, it is evident that the videos were a valuable addition to both sections of the class. Granted, the creation of videos were time consuming for the faculty member, but the high levels of reported usefulness and the fact that many students depended on the video more than their textbook to understand the concepts in the class, seem to indicate that the time was well spent.

Should the instructor spend the time required to create their own video or use video that accompanies the textbook or found on the internet? Although this was not the focus of this study, the fact that students in the online section were able to make some connection with the instructor is demonstrated in the responses to the question covering that subject. In addition to that, many of the students commented on how important it was that the videos contained all of the concepts necessary to complete the lab assignments and that students found them more useful towards the end of the class when they felt that the book did not adequately cover some subjects.

This exploratory study demonstrated the usefulness of including video in the online as well as face to face sections of introductory programming classes. Future areas of study in this area include the use of video in other CIS topics areas, differences among student learning types, and the difference of prepackaged versus instructor created video. In addition to data on student's self-reporting responses, more quantifiable data from usage statistics as well as some correlation to student success in the course would be useful.

6. REFERENCES

Butler, M., & Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. In *Proceedings of the 24th ascilite Conference*, December 2-5, 99-107.

Brecht, H. D., & Ogilby, S. M. (2008). Enabling a comprehensive teaching strategy: Video lectures. *Journal of Information Technology Education: Innovations in Practice*, 7, IIP 71-IIP 86.

Cain, W. P. (1991). Object-oriented programming and the CIS curriculum. *Journal of Information Systems Education*, 3(1), 2-7.

Choi, H. J., & Johnson, S. D. (2005). The effect of context-based video instruction on learning and motivation in online courses. *The American Journal of Distance Education*, 19(4), 215-227.

Donkor, F. (2011). Assessment of learner acceptance and satisfaction with video-based instructional materials for teaching practical skills at a distance. *The International Review of Research in Open and Distance Learning*, 12(5), 74-92.

Dubrowski, A., & Xeroulis, G. (2005). Computer-based video instructions for acquisition of technical skills. *Journal of Visual Communication in Medicine*, 28(4), 150-155.

Dusenbury, L. A., Hansen, W. B., & Giles, S. M. (2003). Teacher training in norm setting approaches to drug education: A pilot study comparing standard and video-enhanced methods. *Journal of Drug Education*, 33(3), 325-336.

Ellis, R. S., & Okpala, C. O. (2004). Evaluation of digital technology and software use among business education teachers. *Journal of Instructional Psychology*, 31(1), 53-59.

Fanning, E. (2008). Instructional design factors as they relate to the creation of a virtual learning environment. *Journal of Interactive Instruction Development*, 21(2), 24-42.

Fincher, S. (1999). What are we doing when we teach programming? In *Proceedings of the 29th ASEE/IEEE Frontiers in Education Conference*, November 10-13, 12a4-1-12a4-5.

Gainsburg, J. (2009). Creating effective video to promote student-centered teaching. *Teacher Education Quarterly*, 36(2), 163-178.

Geri, N. (2011). If we build it, will they come? Adoption of online video-based distance learning. *Interdisciplinary Journal of E-Learning and Learning Objects*, 7, 225-234.

Gotwals, J. K., & Smith, Jr., C. R. (1995). Restructuring programming instruction in the computer information systems curriculum: One department's approach. *Journal of Information Systems Education*, 7(2), 67-72.

Gotwals, J. K., & Smith, M. W. (1993). Bringing object-oriented programming into the undergraduate computer information systems curriculum. *Journal of Information Systems Education*, 5(3), 2-8.

Hartsell, T., & Yuen, S. C. (2006). Video streaming in online learning. *AACE Journal*, 14(1), 31-43.

Jadin, T., Gruber, A., & Batinic, B. (2009). Learning with E-lectures: The meaning of learning strategies. *Journal of Educational Technology & Society*, 12(3), 282-288.

Jenkins, T. (2002). On the difficulty of learning to program. In *Proceedings of the 3rd Annual LTSN-ICS Conference*, August 27-29, 53-58.

Jones, T., & Cuthrell, K. (2011). YouTube: Educational potentials and pitfalls. *Computers in the Schools*, 28(1), 75-85.

Katz, Y. J. (2000). The comparative suitability of three ICT distance learning methodologies for college level instruction. *Educational Media International*, 37(1), 25-30.

Lahtinen, E., Ala-Mutka, K., & Jarvinen, H. (2005). A study of the difficulties of novice programmers. In *Proceedings of the 10th ITiCSE*, June 27-29, 14-18.

Lents, N. H., & Cifuentes, O. E. (2009). Web-based learning enhancements: Video lectures through voice-over PowerPoint in a majors-level biology course. *Journal of College Science Teaching*, 38(2), 38-46.

Martin, E., Evans, P., & Foster, E. (1995). The use of videos in the teaching of accounting. *Accounting Education*, 4(1), 77-86.

Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming – Views of students and tutors. *Education and Information Technologies*, 7(1), 55-66.

Mintu-Wimsatt, A. (2001). Traditional vs. technology-mediated learning: A comparison of students' course evaluations. *Marketing Education Review*, 11(2), 63-73.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.

Russell, J., Russell, B., Pollacia, L. F., & Tastle, W. (2010). A study of programming languages used in information systems and in computer science curricula. *Information Systems Education Journal*, 8(56), 1-15.

Sedaghat, A. M., Mintz, S. M., & Wright, G. M. (2011). Using video-based instruction to integrate ethics into the curriculum. *American Journal of Business Education*, 4(9), 57-76.

Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, Jr., J. F., Sipior, J. C., & de Vreede, G. (2010). IS 2010: Curriculum guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems*, 26(18), 360-428.

Building an Effective Interdisciplinary Professional Master's Degree

Douglas M. Kline
klined@uncw.edu
Information Systems and Operations Management,

Ron Vetter
vetterr@uncw.edu
Computer Science Department

Karen Barnhill
barnhillk@uncw.edu
Cameron School of Business Graduate Programs Office

University of North Carolina Wilmington
Wilmington, NC 28403, USA

Abstract

This article describes the creation of the Master of Science of Computer Science and Information Systems at University of North Carolina Wilmington. The creation of this graduate degree was funded by the Sloan Foundation as a new type of program, the Professional Master's. The program was designed with significant industry input, and is truly interdisciplinary, spanning not only departments, but schools and colleges. The planning, start-up, operation, and formal review of the program are reviewed. IS Educators planning or administering graduate programs should benefit from the review of challenges and solutions provided.

Keywords: graduate programs, professional master's degree, curriculum, program planning, program administration

1. MOTIVATION

In the Fall of 2005, the University of North Carolina admitted its first students into the Master of Science of Computer Science and Information Systems degree program. The program was unique for several reasons:

- Interdisciplinary – across colleges
- Professional Science Master's – funded by the Sloan Foundation
- Industry driven

The program has been successful, meeting its enrollment goals, achieving near 100% placement, generating scholarly articles, and providing significant indirect benefits to the university and professional community.

The goal of this paper is to provide a case study reference for the creation of similar programs. To this end, we will describe the program in detail, review the seven years since inception, discuss challenges encountered, and identify keys to the ongoing success of the program.

2. PROGRAM DESCRIPTION

The Master of Science in Computer Science and Information Systems at UNC Wilmington is a 36 hour program designed to prepare students for advanced careers in the information technology field. The program is delivered jointly by the Computer Science department and the Information Systems faculty in the Information Systems and Operations Management department.

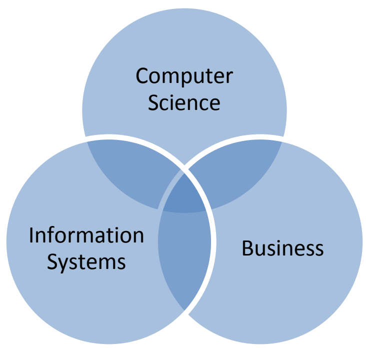


Figure 1

The main strength of the MS CSIS program is its interdisciplinary nature. This was a major motivation in its creation, and is supported by the current feedback from the stakeholders. The Interdisciplinary nature of the program is represented by the Venn diagram in Figure 1.

The three areas have these characteristics:

- Computer Science – technical theory, science, research, academic contributions
- Information Systems – technical application, implementation, professional contributions
- Business – business theory and application

The Venn diagram represents that none of the characteristics are exclusive of any one discipline. Faculty, courses, capstone projects, and students fall in various parts of the Venn diagram. We view this interdisciplinary nature as a strength of the program, creating well-rounded graduates. Traditional single-discipline programs create, for example, business-minded graduates without technical skills, or technical-minded graduates without business skills. This program, furthermore, strives to cover both theory and application, so that graduates are able to apply their knowledge in the workplace.

Curriculum

The program consists of 36 graduate credit hours, in 3-credit hour courses.

Component	Credit hours
Core Courses	18
Capstone	6
Electives	12
	36

The Core is required of all students, and consists of the six 3-credit courses. Each Core class has a specific content pre-requisite that is typically satisfied by some undergraduate MIS or CSC course. The Core and prerequisites consist of:

- CSC 532 Design and Analysis of Algorithms
  - undergraduate Data Structures
- MIS 534 Information Security Management
  - undergraduate Networking or Telecommunications
- CSC 544 Network Programming
  - undergraduate Networking or Telecommunications
- CSC 550 Software Engineering
  - Undergraduate systems analysis OR software engineering
- MIS 555 Database Management Systems
  - Undergraduate database
- MIS 565 Analysis, Modeling, and Design
  - Undergraduate systems analysis OR software engineering

The capstone component can be satisfied with either a Project or Thesis. The capstone is performed under the guidance of a committee of faculty, chaired by a graduate faculty member. The capstone is typically accomplished over two semesters, and represents a significant contribution to the body of knowledge in IS/CSC Academia (Thesis) or IS/CSC Profession (Project). The capstone requires a public proposal and a public final defense. The capstone also requires a significant document in addition to deliverables agreed upon by the committee.

The twelve required elective hours can be made up of:

- CSC/MIS 5XX various offerings
- CSC/MIS 591 Directed Independent Study
- CSC/MIS 592 Topics in Computing
- CSC/MIS 598 Internship

- Other graduate courses aligned with career goals and approved by academic advisor

In addition to specific course prerequisites, there are also significant program prerequisites. These not only give context to the application of core coursework, but also contain the concepts and language used in the core coursework. The program pre-requisites are:

- Introductory Computer Programming
- Intermediate Computer Programming
- Introductory Marketing
- Introductory Management
- Introductory Accounting
- Introductory Finance

The program is considered a Professional Science Master's (PSM) degree and is a member of the UNC Professional Science Master's degrees (http://www.ncsu.edu/grad/psm/). These programs are typically terminal degrees geared towards professional employment, combining Business and Professional curriculum with Science, Technology, Engineering, and Math (STEM) curriculum.

The program is delivered in a manner to serve both part-time local working professionals and full-time students. Courses are offered in the evenings or late afternoons. Where appropriate, courses are offered one evening a week.

The program directly strengthens the Computer Science, Information Systems, and Business degrees at UNC Wilmington:

CSIS Graduate Assistant activities

- assist in CSC undergraduate courses (some serve general campus degrees)
- assist in CSC department technical support
- assist in MIS undergraduate courses (some serve campus and business degrees)
- assist in MIS department, Cameron School of Business technical support
- assist in CSC faculty research (sometimes with other disciplines, e.g. Geography, Psychology)
- assist in MIS faculty research (sometimes with other disciplines, e.g. Finance, Marketing)
- assist in CSC and MIS grants
- co-author research with CSC faculty
- co-author research with MIS faculty

CSIS faculty (CSC and IS) are developed professionally by:

- teaching graduate level courses
- mentoring graduate students
- participating in capstone projects
- developing electives for graduate courses

Less overtly apparent are these benefits to the campus and community:

- CSIS students are employed by other academic departments and non-academic offices on campus
  - Information Technology Services
  - Admissions
  - Housing and Residential Life
  - Continuing Education
  - Graduate School
- CSIS students are employed as interns, part-time employees, and full-time employees by local businesses
  - GE-Hitachi
  - Visionair
  - PPDI
  - Many others
- CSIS students support grants in other disciplines
  - Chemistry
  - Biology
  - Psychology
- CSIS capstone projects have become production systems used by campus agencies
- CSIS students and faculty co-author research with faculty from other disciplines on campus
  - Marketing
  - Psychology
  - Finance
  - Operations Management
  - Mathematics

Organizational Structure

The organization structure is depicted in Appendix 1. In summary, the program is overseen by two main entities: the MS CSIS Program Committee, and the Cameron School of Business Graduate Programs Office. The CSIS Committee is comprised of two Computer Science graduate faculty and two Information Systems graduate faculty. One of the four members is director, with the directorship alternating between CS and IS faculty. The CSIS Committee handles curriculum changes, admissions, and graduate assistantship and scholarship allocations.

The CSB Graduate Programs Office handles operational aspects such as individual degree requirements, payment of graduate assistantships, marketing, reporting, etc. There are synergies of operations among the CSIS program and the other programs within CSB: the International MBA, Professional MBA, and the Master of Science in Accountancy.

Of significant impact to the CSIS program is the organization structure that spans the College of Arts and Sciences and the Cameron School of Business. There are significant differences in funding, procedures, culture, and philosophy that impact the program. As an example of how impactful this is, students are entered into campus information systems as EITHER Arts & Sciences students OR Cameron School of Business students. This determines state formula funding during student coursework and ultimately follows students to commencement, where they attend EITHER Arts & Sciences OR Cameron School of Business commencement.

Another example of the challenges of integrating the two departments is how each views program content. For example, the Computer Algorithms core course was generally viewed as not necessary by the Information Systems faculty, while the Computer Science faculty viewed it as absolutely critical. The Systems Analysis course was championed by the Information Systems faculty, and viewed as low value by the Computer Science Faculty. In the end, the two courses were both included as a compromise.

The organization chart lines represent both formal and informal lines of reporting and/or communications. The purple boxes in Appendix 1 represent the two departments involved. In general, changes to the program are slower and more difficult to achieve than in single-department and/or single-school/college programs. Differences in philosophy or practice between the departments require more discussion to achieve a "meeting-of-the-minds". Fundamental motivations are influenced by differences in funding and compensation between the two departments, for example, teaching load, equipment purchases, department budget, travel compensation, training compensation, publication requirements, etc.

Accreditation
The program falls under UNC Wilmington's accreditation by the Southern Association of Colleges and Schools (SACS, http://uncw.edu/planning/sacs.html). Furthermore, the program falls under the Cameron School of Business accreditation by the Association to Advance Collegiate Schools of Business (AACSB). The program is also a Professional Science Master's degree, and participates in the University of North Carolina System-wide Professional Science Master's Programs.

The Computer Science department's Bachelor of Science degree is accredited by the Computing Accreditation Commission of the Accreditation Board for Engineering and Technology (ABET).

3. HISTORY

The Request for Authorization to Establish a New Degree Program was submitted in October of 2004, after several years of planning. The degree officially accepted students in the Fall of 2005. The CSC department had been offering graduate level courses for several semesters, so some early students entered the program with coursework towards the degree program.
The program was initially staffed with 10 tenure-track CSC graduate faculty, and 4 tenure-track MIS graduate faculty. The program is now staffed with 11 tenure-track CSC graduate faculty, and 6 tenure-track MIS graduate faculty. However, several faculty members are in administrative positions that limit their involvement in the program.

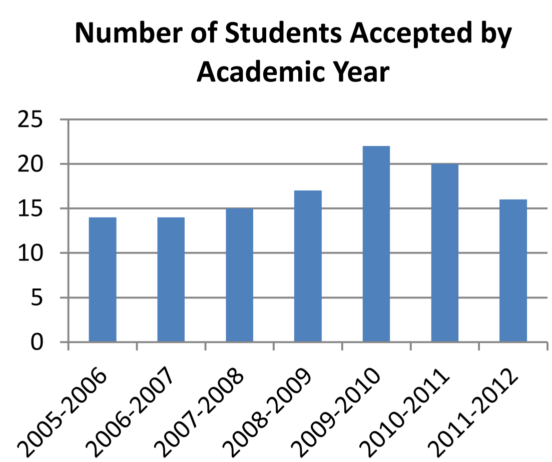


Figure 2

Fourteen(14) students were accepted in the first academic year (refer to Figure 2). Admissions generally increased for the first five years, with a high of 22 admissions in the 2009-2010 academic year. Admissions have somewhat fallen, with 16 admissions in the 2011-2012 academic year. As shown in Figure 3, 45 students have graduated from the program (note that 2011-2012 contains expected graduations).

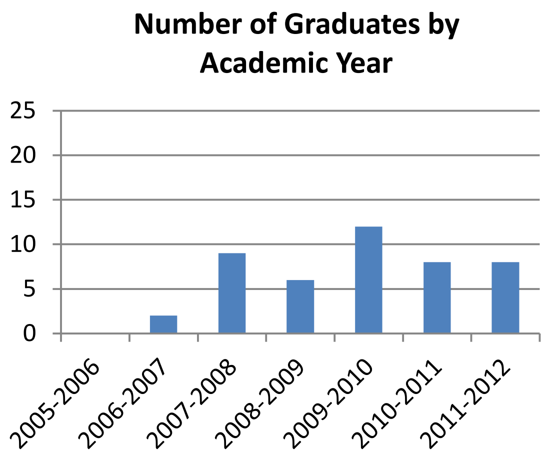


Figure 3

Several minor curriculum changes occurred since program inception, the main one concerning capstone projects. The hour requirement was standardized in the catalog so that both Project and Thesis options required 6 credit hours devoted to the Project/Thesis. This catalog change matched practice and made the requirements clearer.

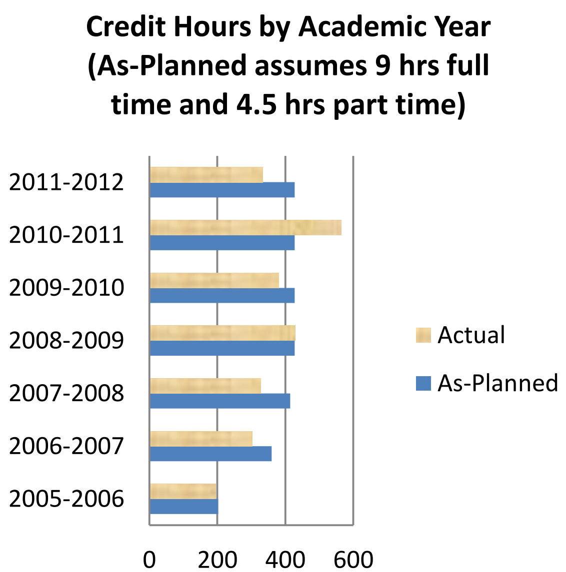


Figure 4

From the Request for Authorization to Establish document, the program was expected to begin with 20 Full Time and 5 Part Time students, and have, by Year 4, 40 Full Time students, and 15 Part Time students. Steady-state was expected to be 40 Full-Time, and 15 Part-Time students. Figure 4 shows the As-Planned versus Actual credit hours for the program, by year. The program currently has 51 students pursuing the degree, but it is difficult to classify them clearly as full-time or part-time (many began as full-time, are now part-time, but may become full-time again). Figure 4 shows the total planned credit hours versus the total actual credit hours enrolled in by students. In short, the program met its enrollment projections, but has since fallen off from the projected enrollment. This appears to be partly due to the national graduate enrollment trends, and the cancellation of multiple electives in the past year due to resource constraints.

4. OPERATIONS

Recruitment

At the inception of the program, the state of North Carolina participated in a twelve-state consortium called the Academic Common Market. In summary, if the desired degree was not offered in a student's home state, the student could pay in-state tuition at one of the consortium schools. For the MS CSIS degree, this was a significant benefit in our marketing – students from twelve states could pay in-state tuition to come to our program. Unfortunately, North Carolina no longer participates in the Academic Common Market consortium.

From the marketing research and experience of our graduate school, incoming students found out about their program in two main ways: the internet and word-of-mouth. Many of the traditional recruiting efforts have shown to be of little value:

- Campus visits
- Graduate Program Fairs
- Corporate Visits

These recruiting methods have worked well for the Master of Science of Accounting and the Professional MBA program, which draw students mainly from the local community.

Our current marketing efforts focus on two items: creating searchable content on our web site, and actively using social media, mainly

LinkedIn. Each capstone project results in a substantial document that we post on our web site. This effectively builds content, but we have not previously received full benefit because of deficiencies in search-ability. Search engine optimization fundamentals such as review and monitoring of web site statistics, proper meta tags, proper sitemaps, and changing content, are essential. Here are some additional initiatives that we are undertaking:

- Each project has a "landing page" with abstract, cross links to faculty, and can be linked to by other sites. This makes it easy for others to link to individual projects.
- Each pdf document will have a link to the main master's program URL. Without this, the project documents are found, but the program may not be.

We have viewed social media as the combination of the two ways students find programs: the internet and word-of-mouth. In addition to creating Facebook and LinkedIn presences, we have begun to integrate LinkedIn into some of our procedures:

- Announcements of public capstone project proposals and defenses must be made by students on their LinkedIn accounts, and shared with the MS CSIS LinkedIn group.
- Announcements of graduations are posted in the LinkedIn MS CSIS group
- Announcements of published papers are posted in the LinkedIn MS CSIS group
- Faculty are reminded to endorse students, as appropriate, via LinkedIn

Although we cannot directly identify results in recruiting from this new effort, we see other positive benefits. This forces students to consider and develop a professional online presence prior to graduation. Faculty have improved their own online professional profiles to be better references to students. Students give each other virtual "pats on the back" as they complete or announce items. In short, this has shown to have a very positive, albeit mostly intangible effect, that we think will pay dividends in the future.

Scheduling of classes

Classes are scheduled by the individual department chairs, with a preference toward offering late afternoon and evening class times. Conflicts are avoided by the simple agreement that IS graduate classes will be offered on Monday and Wednesday, and CS graduate classes will be offered on Tuesday and Thursday. However, there are still coordination issues attempting to ensure that the core classes are offered in the same semester each year, and that a sufficient number and variety of electives are offered.

Capstones

Capstone projects are supervised by a committee comprised of a chairperson, and at least two other members. The committee must contain at least one member from the IS faculty and from the CS faculty. The third member can be from IS/CS, from another academic discipline, or an IT professional.

The capstone is meant to contribute significantly to either the IS/CS academic discipline(s) or the IS/CS profession. The capstone can take the form of a Thesis (academic contribution) or a Project (professional contribution).

Six (6) credit hours of capstone are required, and the capstone is typically completed over the academic year (9 months). The first three (3) credit hours are typically used to research and write a proposal, which is delivered in a public presentation. The second three (3) hours are typically used to execute/implement what was proposed, culminating in a public final defense of the capstone.

The capstone element has been labor-intensive, but also extremely valuable and rewarding, for both students and faculty. Capstones have resulted in:

- Published Journal and Conference Proceedings
- Creation of, and improvement to, real-world public and private production information systems
- Job offers specifically based on work from capstones

5. PROGRAM REVIEW

Student Body

Several promising trends were apparent in the review of the student body from 2005 to 2012 (specific statistics available from authors).

- Average test scores (GMAT/GRE) of incoming students rose
- Average Verbal GRE rose
- The gender diversity increased significantly
- The racial diversity increased significantly

- The percentage of non-UNCW backgrounds increased from 7% to 68%

In general, students who desire graduate assistantships have been able to obtain them, either through the CSIS program, or with other academic or administrative departments on campus. Note that in 2011-2012, 6 additional graduate assistantships were available through grants with the Chemistry and Psychology departments

Students who are admitted attend a one-day Orientation at the beginning of the Fall semester. The Orientation includes program information, research presentations from faculty, a Meyers-Briggs assessment, professional etiquette session, and a team-building low-ropes course. Once each Fall and Spring, a Capstone Orientation is held covering the details of topic selection, committee selection, and administration of capstones. At the end of the Spring semester, a Graduation dinner is held (in addition to the UNCW Commencement) to celebrate graduates along with their families. A variety of other activities are available to them through the departments: IT Career Night, Wilmington IT Exchange and Expo, Business Week, IT Advisory Board Meeting, and ad hoc meetings with IT professionals.

In general, funding is available for students to give research presentations. However, the MS CSIS program does NOT have a budget to fund student travel. The funding is procedurally difficult to obtain, coming from multiple sources, each with their own requirements and procedures. Sources for student funding include:

- the Cameron School of Business
- the Information Systems and Operations Management Department
- the Computer Science Department
- the UNCW Graduate School
- the UNCW Graduate Student Association
- ad hoc grants

Assessment
The CSC and ISOM departments collaborated to establish the following learning objectives for the program:

1. Discipline Specific Knowledge, Skills, Behavior and Values
2. Critical Thinking
3. Communication

These learning objectives are measured with the following instruments:

- Content Knowledge Assessments for each of the Core Classes
  - 10 questions from each Core class
  - Administered at completion of degree (previously at end of course)

- Oral Communication Assessments
  - In two courses: Systems Analysis and Software Engineering
  - Oral communication rubric completed by instructor for oral presentations

- Capstone Project/Thesis Evaluations
  - Completed by each committee member at final defense

In general, the assessment instruments have been helpful in summarizing information across students and courses. Rather than making decisions on individual faculty's anecdotal observations, decision making is improved with more rigorous analysis.

Program Objectives were established by collaboration among the faculty of the CSC and IS areas. The Program Objectives are also aligned with the UNCW Strategic Goals. The Program Objectives are kept in mind as event, curriculum, and administrative decisions are made.

The Program Objectives are:

- Increase dialogue between industry and the MS CSIS Program
- Provide learning opportunities for faculty
- Improve student recruitment

Stakeholder Feedback
As part of the program review, the following activities were planned and executed:

- IS/MSCSIS Advisory Board Meeting Breakout (Feb 1, 2012)
- Program Review Kickoff Meeting (Feb 3, 2012)
- Current Student Focus Group (Feb 16, 2012)
- ISOM Faculty Meeting Feedback (Feb 24, 2012)
- CSC Faculty Feedback (March 5, 2012)

In particular, the IS/MSCSIS Advisory Board Meeting and the Current Student Focus Group offered well-reasoned, thoughtful, constructive feedback.

The bullet items below summarize the feedback from all these stakeholders.

Strengths

- Interdisciplinary nature of program
- Applied nature of program
- Smaller size of program
- Student Aid (assistantships)
- Placement of graduates / Job opportunities
- Ability to personalize program/flexibility
- Quality of faculty
- Professional networking opportunities
- Friendly and helpful staff
- Wilmington location – tourist destination

Weaknesses

- Lack of Global IT component
- Too few electives
- Faculty resources (too few faculty)
- Lack of an "integrative" class
- Cross-listing of courses; very similar graduate/undergraduate courses
- High Variability in entering student skill sets (technical, business, and applied)
- Lack of discretionary program budget
- Large number of pre-requisites
- Large class sizes (in some cases) / variability of class size

Opportunities

- Foreign student recruitment
- Recruitment outside UNC Wilmington
- Upcoming undergraduate IT major
- Economic recovery
- UNCW's goal of developing graduate programs
- Graduate Business Certificate – (easier to achieve business pre-requisites)

Threats

- Faculty Workload
- Conditional Admits
- Exhaustion of local market
- Lack of involvement from some faculty
- Divide between CSC and IS faculty / departments
- Hardware and technical support
- Changing technologies

Suggestions

- Benchmark more closely with other Prof Science Master's degrees
- Make capstone projects optional (due to too few faculty)
- Include Professional mentoring as part of program
- More teamwork
- More case study methodology
- More Microsoft / .Net – less Java
- Student-to-student mentoring

Many of the positive AND negative items above come from the fact that the program is very broad and serves multiple purposes:

- Full-time students & Part-time students
- PhD-bound students & profession-bound students
- Multiple technology foci: telecommunications, software development, project management, etc.
- Multiple student backgrounds & skillsets: MIS / CSC
- Content from multiple disciplines: MIS/CSC/Business
- Theory & Application

Academic output (research from)

Table 1 lists the research that is attributable to the program. This research mainly comes from capstone projects and theses, but also comes from classroom assignments, or independent studies. The research outlet for most is refereed conference proceedings, but also includes refereed journal articles in journals such as Computer, Journal of Information Systems Applied Research, and Journal of Information Systems Education.

Table 1

Year	Ref'd Jrnl Articles	Ref'd Conf Proc
2012	3	
2011		5
2010		7
2009		4
2008		8
2007	1	5

Placement

Graduates of the program are well-received by employers. Forty five (45) students have graduated since inception, with 98% employment within the IT profession. (During the economic downturn, employment was not necessarily immediate upon graduation.) In fact, many students accept employment offers before graduating. The average starting salary for graduates since 2005 is approximately $60,000. The average starting salary for graduates from Fall 2009 to the present is approximately $63,000. Employers include, but are not limited to:

- Pharmaceutical Product Development, Inc. (PPDI)

- GE-Hitachi
- Corning
- New Hanover Regional Medical Center
- TriTech
- TranS1
- ATMC
- UNC Wilmington
- Construction Imaging
- Speciality Soft
- Guilford Mills
- Engineering Software Solutions
- Cloudwyze
- American Eagle
- Deloitte and Touche
- Price Waterhouse Coopers
- AT & T
- DAK Americas
- SciQuest
- Credit Suisse
- CTG
- LabAnswer
- Lideli (Japan)

The placement covers a wide range of industries and professions. Industries covered include:

- Banking/Finance
- Software Development
- Engineering
- Healthcare
- Manufacturing
- Education
- Retail
- Information Technology
- Consulting
- Telecommunications

6. CONCLUSION

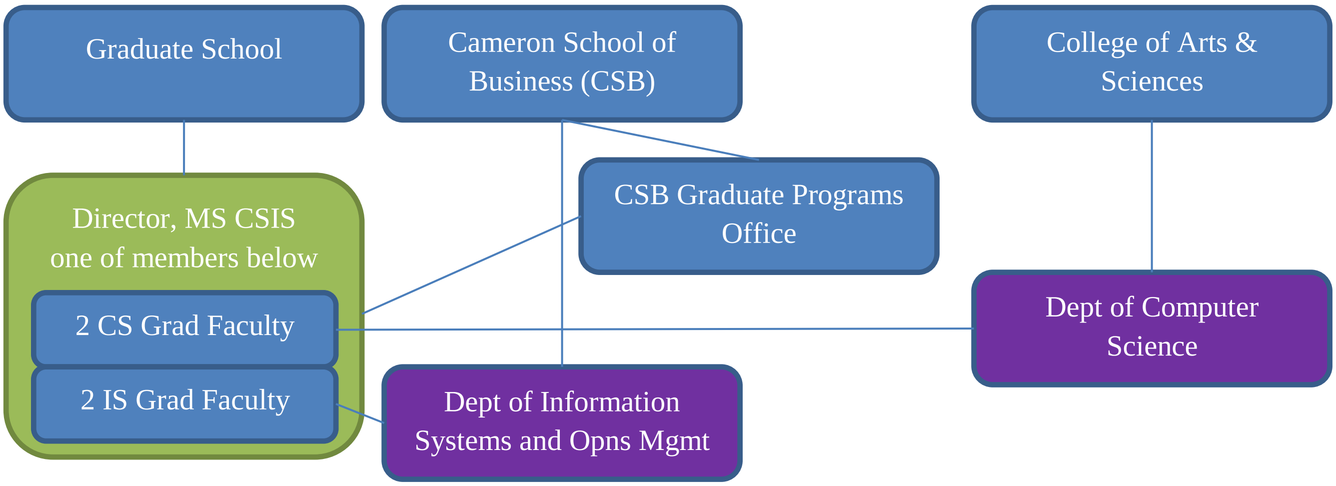
The creation of the Master of Science of Computer Science and Information Science degree has been challenging and rewarding. The program is truly interdisciplinary, which has created many of the challenges, but has also generated many of the rewards.

The program has aided many alumni in their professional careers. In addition the program has had positive impact on the departments, undergraduate students, the university, and local IT professionals and organizations.

The formal review of the program was a positive experience, confirming our judgment that the program was fundamentally well-conceived and implemented. However, numerous potential improvements were identified. Because the program is still relatively young, there are many operational inefficiencies that need to be refined.

At a more strategic level, recruitment is an area that could dramatically improve many aspects of the program. A larger applicant pool will improve program metrics and outcomes, as well as have positive cultural effects on the student body, departments and university.

Appendix 1: Organization Structure



Ten Year Assessment of Learning Outcomes of a Computer Information Systems (CIS) Program

Samuel Abraham
sam@sienaheights.edu
Siena Heights University
Adrian MI 49221

Abstract

In recent years greater attention has been paid to develop learning outcomes for academic programs and then to develop methods to assess these learning outcomes. Generally speaking, there are two kinds of outcomes: course outcomes and program outcomes. Assessments of these learning outcomes in institutions of higher education are mandated by the accrediting organizations. This paper describes a methodology used by a Computer Information Systems program in a small undergraduate institution to develop its learning outcomes, to collect assessment data, and to evaluate or assess its course and program outcomes during a ten year period. The data collection and the subsequent data analysis showed the strengths and weaknesses of the program and we were able to address a number of these weaknesses.

Key words: Course outcomes, Learning outcomes, Learning outcome assessments, Measurement, Outcome based education, Program outcomes, and Programs metrics.

1. INTRODUCTION

In recent years greater attention has been paid to develop learning outcomes for academic programs and then to develop meaningful assessment methods to evaluate these outcomes. Assessment is a systematic and on-going process of collecting, interpreting, and acting on information relating to the goals and outcomes developed to support the mission and purpose of an institution (Osters, 2003). According to Acharya (2003), assessments should help us to answer the following questions: (1) What do we want the students to learn? (2) Why do we want them to learn it? (3) How can we help them to learn it? (4) How do we know what they have learned? Also Osters (2003) pointed out that assessments should help us to improve what we are doing. Assessment begins with the articulation and development of measurable outcomes. Generally speaking, there are two kinds of learning outcomes: course outcomes and program outcomes. The course outcomes should describe what students are expected to learn from an individual course, while program outcomes should describe what a student is expected to accomplish after completing the coursework from the program. Maki (2002) pointed out that learning outcome assessments must be based on institutional curiosity to seek answers to questions about student learning, why they learn, how well they learn, when they learn, and explores how pedagogies and educational experiences develop, and foster student learning. Maki (2002) also pointed out that innovations in pedagogy or integration of diverse methods of teaching and learning into a program of study, redesign of a program, reconceptualizing the role of advising, or establishing stronger connections between curriculum and non-curriculum represents some of the kinds of changes that faculty and staff may undertake to improve student learning and development based on their interpretations of learning outcome assessment results.

2. PROGRAM LEARNING OUTCOME DEVELOPMENT

Learning outcomes should describe what students will be able to demonstrate in terms of knowledge, skills, and values upon completion of a course, a span of several courses, or a degree program (Osters, 2003). Clear statement of learning outcomes serves as the foundation to assess the effectiveness of the teaching and learning process. According to Osters (2003), the three essential components of a measurable learning outcome are: (1) Student learning behaviors, (2) Appropriate assessment methods, and (3) Specific student performance criteria. Student behaviors describe what students are expected to demonstrate by the completion of the course. Action verbs like demonstrate, apply, define, analyze, etc. are used to describe student behaviors. Assessment methods are tools and techniques used to determine the extent to which the stated learning outcomes are achieved. Student performance criteria should be expressed in specific and measurable terms that are acceptable to a specific course or series of courses. A variety of methods, qualitative and quantitative, direct and indirect, should be used to assess the learning outcomes. Keep in mind that a simple letter grade alone does not provide adequate feedback to student's performance, because the letter grade alone does not sufficiently identify with the strengths and weaknesses of individual learning outcomes. If the grading system is accompanied by a rubric where the individual outcome components are addressed, then this tool can be used to pinpoint the weakness and strengths of the student's performance.

It is very important to define the learning outcomes of a program/course in specific and precise manner. Spady & Marshall (1994) wrote:

> "Outcomes are clear, observable demonstrations of student learning that occur after a significant set of learning experiences…Typically these demonstrations, or performances, reflect three things: (1) what the student knows; (2) what the student can actually do with what he or she knows; (3) the student's confidence and motivation in carrying out the demonstration. A well-defined outcome will have clearly defined content or concepts and be demonstrated through a well-defined process beginning with directive or requests such as explain, organize, or produce."

After an exhaustive research the faculty members developed a number of outcomes for the CIS program and from this list we were able to select six measurable outcomes for our program. The American Association of Higher Education's (AAHH) (1996) nine principles of good practices for assessing student learning were used in the selection process. We also used a number of other research documents from the AAHE's assessment web site. Our hope is that the graduates of our program will be able to show that they have accomplished these six outcomes by receiving a degree from the CIS program. The following list shows the learning outcomes developed by the CIS program.

1. Students will demonstrate the skill to write complete, complex programs that are fully tested.
2. Students will demonstrate the skill to develop a complete information system that incorporates feasibility study, analysis, design, systems development, testing, implementation and maintenance.
3. Students will demonstrate the ability to solve problems using the computer as a tool, using either application packages or custom programs.
4. Students will demonstrate the ability to work as a team member in a problem-solving situation.
5. Students will demonstrate the ability to investigate existing literature in Information systems.
6. Students will demonstrate the ability to communicate effectively.

Fig 1

3. COURSE LEARNING OUTCOME DEVELOPMENT

Once these outcomes were developed, we set out to see how these outcomes can be accomplished through our course offerings. We know that we have to develop a set of outcomes for each of our courses, keeping in mind that there must be a match between these course outcomes and the program outcomes. In other words, the stated program outcomes must be accomplished through the course outcomes. Faculty who are teaching the individual courses are asked to take the program outcomes and see how these outcomes can be accomplished through their courses. Also these are the outcomes a faculty would like his/her students to know at the completion of that particular course. Axelsson and Melin (2010) pointed out that when learning outcomes are developed in a transparent and clear way, students will be able to use them before, during and after the course. The importance of measurability and clarity of the course outcomes were emphasized. Faculty members developed a set of learning outcomes for each course from which we selected five or six outcomes for each individual course. We then developed a table to show the relationship between program outcomes and courses offerings. We also agreed that when we develop new courses in the future, we need to pay greater attention to the course outcomes to see how the new course will satisfy the program outcomes. By adding new rows in Table 2 we will be able to get a quick view of the relationship between the course and program outcomes.

Course Number	Course Title
CIS 119	Visual Basic Programming
CIS 218	Introduction to Information Systems
CIS 252	Introduction to C++ Programming
CIS 260	Cobol Programming
CIS 353	Systems Analysis
CIS 363	Data Base Structures
CIS 443	Data Communication
CIS 465	Management Information Systems
CIS 495	Senior Project
Electives	
CIS 352	Data Structures Using C++
CIS 340	Java Programming
CIS 370	Network Operating Systems
CIS 455	Computer Hardware & Software
CIS 460	Web Development
CIS 470	Information Assurance
CIS 480	Internships
CIS 485	Emerging Technology

Table 1

Required Courses	**Learning Outcomes**					
	1	**2**	**3**	**4**	**5**	**6**
CIS 119	X		X			X
CIS 218			X	X	X	X
CIS 252	X		X			X
CIS 260	X		X			X
CIS 353		X	X	X	X	X
CIS 363		X	X	X	X	X
CIS 443			X	X	X	X
CIS 465			X	X	X	X
CIS 495	X	X	X		X	X
Electives						
CIS 352	X		X			X
CIS 340	X		X			X
CIS 370			X	X	X	X
CIS 455			X	X	X	X
CIS 460		X	X	X	X	X
CIS 470			X	X	X	X
CIS 480			X		X	X
CIS 485			X		X	X

Table 2

The current CIS course offerings (course numbers and corresponding course titles) are listed in Table 1 for reference. Table 2 shows a mapping of the courses and the CIS program outcomes.

Upon completion of this course, students will be able to demonstrate proficiency in:

1. A disciplined approach to problem solving methods and algorithm development (CIS-O#1, 3)

2. The syntax and vocabulary of Visual Basic.Net (CIS-O # 1)

3. The usage of Visual Basic.Net Programming Environment (CIS-O #1)

4. Developing complete Visual Basic programs that include specification, design, code, debugging, testing, and documentation. (CIS-O #1)

5. Using computers as a tool in problem solving (CIS-O # 3)

6. Communicating the program development process in a predetermined format (CIS-O #6)

Fig 2

As mentioned earlier, we have developed a number of outcomes for each course. Faculty members also developed a number of rubrics for each course to assess the achievement of each student.

The outcomes developed for CIS 119 (Visual Basic Programming) are given above. Similar outcomes were developed for all the other courses in our curriculum.

4. EARLY LEARNING OUTCOMES ASSESSMENT

Data collection and analysis of the course outcomes for each course is described in a previous paper (Abraham, 2006). This paper will be concentrating on the data collection and analysis for a period of ten years. At the end of each academic year we consolidate the data collected during the Fall and Winter semesters and report back to the academic dean who collects the data for accreditation and program assessment purposes. Please see appendix B for a sample form for this reporting. This form is developed by the office of the academic dean and used by all programs/departments in the University for uniform reporting. Column one states the outcomes while column two lists the course that satisfy the outcome and these course come from the previous mapping (table 2). Column three lists the activities that a student will perform to satisfy the fulfillment of the outcome. Column four lists the percentages of students who met or exceeded the outcome expectations. The last column lists the actions taken by the program to address the issues raised during the data analysis. Columns one, two and 3 are self-explanatory while columns four and five need some explanation. We have made two assumptions to generate the data in column four. It is assumed that if a student has received more than 70% for an outcome then he/she has met the requirements for that outcome. For example if a student has received an aggregate of more than 70 % for the program development (outcome 1) then that student has met the requirements for outcome 1. It is assumed that if more than 80% of the students in a class met the requirements for an outcome, then that class has met the requirements for that outcome. For example if more than 80% of the students in Visual Basic Class (CIS 119) have met the outcome 1 requirements, then the whole class has met the requirements for outcome 1. The 70% and 80% guidelines are quite arbitrary and we thought that those numbers are suitable for our purpose. Only the students who have completed the course are included in this analysis. As mentioned elsewhere, we are a small institution and thus all courses are not offered all semesters. So the outcome assessment data collected every year may consist of data from uneven offerings. For example we offer CIS 119 (Visual Basic Programming) every semester while CIS 340 (Java Programming) will be offered only once in a year. To make things more difficult we offer some courses only once in two years.

The outcome analysis revealed a lot of strengths and weakness of our course offerings and course delivery. Colum five lists the changes we have already implemented or the changes we are planning to implement as a result of the yearly outcome analysis. Some of these changes are easy to implement while some others need budget support from administration. For example we were able to emphasis the importance of more team work in all upper level courses as a result of the evaluation in early years. Some of the changes require hardware and software implementation and these kinds of changes need support from the administration.

5. TEN YEAR ASSESSMENT

We have been collecting and reporting the yearly outcome assessment data to the academic dean for almost ten years. Last year we had an accreditation visit from the North Central and thus we were asked to produce a five/ten year report of our program outcome assessment to be included in the final self-study report. The faculty members from the CIS program generated a ten year outcome assessment report. I am very happy to report that the visiting team was very impressed with the progress we have made in the outcome collection and analysis and we have received our 10 year unconditional accreditation.

We are a small institution with a low but steady enrollment and thus 100 and 200 level classes are offered every semester while 300 and 400 level classes are offered in a two year cycle. Data from all classes are collected every semester and combined into an annual report to the dean. For the ten year report we combined all these annual data into one document. A copy of this report is presented in appendix A. Column one of this report restates the six program outcomes and column two lists the course that will satisfy these outcomes as described for table 4 above. Column three lists the average for each course for the ten year period. This average is generated from the yearly data that is reported to the administration at the end of each academic year. The ten year average is calculated for each course and then for each outcome and listed in column 4 of table 6. As stated before the 100 and 200 level classes are offered more times than the 300 and 400 level classes and thus they have a greater effect on the ten year outcome average. We also calculated a final average by taking the average for all the six outcomes. In its current form the report shows that we as a program are doing well and our students are satisfying the stated program outcomes well.

6. LESSONS LEARNED

To draw reasonable conclusions from learning outcome assessments, we should make our assessments as fair as possible. Lam (1995) pointed out that a fair assessment is one in which students are given equitable opportunities to demonstrate what they know. Suskie (2000) suggested the following steps to make our assessments methods as fair as possible: (1) Have clearly stated learning outcomes and share them with your students, so they know what you expect from them, (2) Match your assessment to what you teach and vice versa, (3) Use different measures and many different kinds of measures, (4) Help students learn how to do the assessment tasks, (5) Engage and encourage your students, (6) Interpret assessment results appropriately, (7) Evaluate the outcomes of your assessments.

Learning outcome assessment must be an ongoing process. According to Rodrigues (2002), assessment must become a part of an institution’s culture. We have been doing these assessments for almost ten years now. The faculty in our program felt that the experience of going through the process was very worthwhile, even though it was very time consuming and frustrating. As a result of the data collection and analysis we were able to correct a number of problems in our course offerings as they occurred. The data collection and the subsequent data analysis show our strengths and weaknesses and we were able to address these issues in a timely manner. We strongly believe that the process of data collection and analysis is more important than the final number to understand what is happening in our program at any given point.

We used a number of other assessment techniques other than those described in this paper. All our graduating senior students are required to attend an exit interview. During the interview, a faculty member and the student address the program and course outcomes and solicit recommendation from the students. In addition to oral, written, and poster presentations, faculty members usually visit internship sites to evaluate the performance of the student interns.

7. CONCLUSIONS

Outcome based education promises a better way of understanding student learning, and in turn provide ways to improve the quality of education. To measure or assess the learning outcomes effectively, we need to start with measurable, concise, and specific learning outcomes for our program and individual course that must be shared and explained to the students. Clear and concise measuring tools, techniques, instruments, and methods must also be developed and must be conveyed to the students to avoid confusion and frustration. Assessment data must be collected in an ongoing basis using multiple methods and instruments. Collected data must be analyzed to understand the strengths and weaknesses of the program, courses, teaching, and learning. This information must be used to improve teaching and learning, incorporate innovations in pedagogy, redesign programs and courses, redevelopment of the outcomes, and the development of new tools for assessment. For outcome assessment to be successful it must be ongoing and must be part of the institution’s culture. Administrators must recognize the importance of this process by providing financial and collateral support. Outcome based education is here to stay and it is important for educators to be prepared to accept the challenge of developing measurable outcomes for their programs/institutions, assess these outcomes, and then use the assessment data to improve what they are doing.

8. REFERENCES

Abraham, S. (2006). Assessing the Learning Outcomes of a Computer Information Sys-tems (CIS) Program, Information Systems Education Journal, 4(14). http://isedj.org/4/14/. ISSN: 1545-679X.

Axellson, K and Melin, U.(2010) How to Use the Potential of Learning Outcomes in IS Courses – Listening to the Voices of Students, 2010 ISECON Proceedings Nashville Tennessee, USA v27 n 1327 http://proc.isecon.org/2010/pdf/1327.pdf

American Association for Higher Education. (1996). Nine principles of good practice for assessing student learning [Online] .Available: http://www.aahe.org/assessment/principl.htm

Acharya, Chandrama 2003, Outcome-based Education (OBE): A new Paradigm for learning. CDTLink, November. Retrieved August 15, 2004, from http://www.cdtl.nus.edu.sg/link/nov2003/obe.htm

Lam, T.C.M. (1995) Fairness in performance assessment: ERIC digest [Online]. Available http://ericae.net/db/edo/ED391982.htm.

Maki, Peggy L 2002, Developing an Assessment Plan to Learn About Student Learning, AAHE, January. Retrieved August 15, 2004, from http://www.aahe.org/assessment/assessmentplan.htm

Osters, Sandi 2003, "Writing measurable Leaning Outcomes" Texas A&M 3rd Annual 2003 Assessment Conference, February. Retrieved September 30, 2012, from http://www.gavilan.edu/research/spd/Writing-Measurable-Learning-Outcomes.pdf

Rodrigues, Raymond J, 2002, What campus Buy-In Your Assessment Efforts? AAHEBulletin.com, October. Retrieved August 15, 2004, from http://www.aahebulletin.com/member/articles/2002-10-feature02_pf.asp

Spady, W. and Marshall, K. (1994). Light, not Heat, on OBE. The American School Board Journal. Vol. 181.pp 29-33.

Suskier, Linda 2000, Fair Assessment Practices. AAHEBullitin., July 16. Retrieved August 15, 2004, from http:// www.aahebulletin.com/public/archive/may2.asp

William, K. C and William, S. D. 2004, Planning Assessment of Student Leaning Outcomes: A process Within Your Grasp. International Journal of Nursing Education Scholarship, January.. Retrieved August 15, 2004, from http://www.bepress.com/ijnes/vol1/iss1/art3

Appendix A
Ten Year Outcome Summary

Outcomes	Courses	10 Year Course Averages	10 year Average
Students will demonstrate the skill to write complete, complex programs that are fully tested.	CIS 119 CIS 252 CIS 340 CIS 460 CIS 495	83.25% 82.50% 85.00% 95.00% 96.00%	88.35%
Students will demonstrate the skill to develop a complete information system including feasibility study, analysis, design, systems development, testing, implementation and maintenance.	CIS 353 CIS 363 CIS 460 CIS 495	97.50% 94.00% 94.00% 95.00%	95.13%
Students will demonstrate the ability to solve problems using the computer as a tool, using either application packages or custom programs.	CIS 119 CIS 218 CIS252 CIS 340 CIS 363 CIS 370 CIS 460	83.75% 80.50% 84.00% 84.00% 90.00% 90.00% 95.00%	90.75%
Students will demonstrate the ability to work as a team in a problem-solving situation.	CIS 218 CIS 353 CIS 363 CIS 370 CIS 460 CIS 465	85.75% 95.00% 95.00% 88.00% 89.00% 94.25%	91.17%
Students will demonstrate the ability to investigate existing literature in Information systems	CIS 218 CIS 353 CIS 363 CIS 370 CIS 460 CIS 465 CIS 495	82.00% 86.00% 88.00% 85.00% 90.00% 94.00% 96.00%	88.71%
Students will demonstrate the ability to communicate effectively.	CIS 119 CIS 218 CIS 252 CIS 340 CIS 353 CIS 363 CIS 443 CIS 460 CIS 495	82.75% 82.00% 84.50% 84.00% 90.00% 90.00% 95.00% 90.50% 96.00%	89.52%
		10 Year Program Average	**90.60%**

Appendix B
Academic Program CIS
Learning Outcomes Assessment and Subsequent Actions 2010 – 2011

Outcome	Course or Graduation Requirement	Assignment/Measurement	% of Students Who Met or Exceeded Expectations(% Attainment Desired) **80%**	Actions Taken or to be taken* ***Actions in bold print have been taken**
Students will demonstrate the skill to develop complete, complex programs that are fully tested.	CIS 119, CIS 340, CIS 460, CIS 495	Programming Assignments and projects	CIS 119: 85% CIS 340: 90% CIS 460: 100% CIS 495: 100 %	**We modified the way specification is developed. We now use a standard format to develop the spec.** Continue to emphasis the importance of design and spec.
Students will demonstrate the skill to develop a complete information system that incorporates feasibility study, analysis, design, systems development, testing, implementation and maintenance.	CIS 353, CIS 460, CIS 495	CIS 353: A project where students developed an information system as a team. CIS 460: Students developed a web site as a team CIS 495: Each student developed a complete system individually.	CIS 353: 94% CIS 460: 94% CIS 495: 84%	**We incorporated project management and object oriented aspects in CIS 353**. We are planning to include the above concepts in CIS 460 and 495.
Students will demonstrate the ability to solve problems using the computer as a tool, using either application packages or custom programs.	CIS 119, CIS 218, CIS 340, CIS 353, CIS 455, CIS 460, CIS465, CIS 495	Assigned Lab projects Assigned Homework problems Case Studies Programming assignments	CIS 119: 84% CIS 218: 82% CIS 340: 85% CIS 353: 90% CIS 455: 100% CIS 460: 100% CIS 465: 90% CIS 495: 84%	We are constantly reassessing and modifying our assignments, projects, and case studies in these classes to incorporate more software tools. Also we are watching the changes in technology
Students will demonstrate the ability to work as a team member in a problem-solving situation.	CIS 218, CIS 353, CIS 460, CIS 465	Team projects to do Web search, Complete lab projects, Develop Systems, Develop web sites and Complete Case Studies	CIS 218: 89% CIS 353: 94% CIS 460: 92% CIS 465: 100%	**We incorporated some web2.0 tools in CIS 218** We need to pay more attention to individual performance in teams
Students will demonstrate the ability to investigate existing literature in Information systems.	CIS 218, CIS 353, CIS 455, CIS 460, CIS 465, CIS 495	Research papers with references using APA format	CIS 218: 82% CIS 353: 86% CIS 455: 96% CIS 460: 90%	**We are emphasizing the importance of proper citations and APA formatting in every class.**

			CIS 465: 95% CIS 495: 100%	We need to do a better job in educating our students to reduce plagiarism incidents
Students will demonstrate the ability to communicate effectively	CIS 119, CIS 218, CIS 340, CIS 353, CIS 455, CIS 460, CIS 465, CIS 495	Presentation of lab Assignments Presentation of Research Papers PowerPoint Presentations Poster Presentations	CIS 119: 80% CIS 218: 82% CIS 340: 84% CIS 353: 90% CIS 455: 95% CIS 460: 91% CIS 465: 90% CIS 495: 84%	**We provided specific guidance for proper report preparation, proper format and presentation in all classes**

Action taken or to be taken: (column 5)

1. Earlier assessment data showed that some components of the program development (outcome 1) need more attention. This year we refined the rubric to include more details of the assignments. We developed a standard format for developing the specification. We need to pay more attention to the idea of software engineering rather than developing just programs.
2. We spent a lot of time guiding the senior project students to develop real world projects (outcome 2) that will help them to see the complexities that are associated with developing a real technology project. This year's projects showed a substantial improvement over previous years. We are planning to keep the pressure on them to improve the quality of the senior projects. We are also in the process of incorporating project management tools in CIS 465.
3. We are constantly assessing the use of software packages in our classes. We always use the most recent releases of the software packages. We are also constantly assessing our assignments, Cases Studies, and projects to increase the problem solving skills of the students. We are getting ready to use Microsoft Office 2010 in our classes.
4. Majority of our upper level classes are now using team based projects and they are required to present their team projects orally in addition to their written paper.. We are using a rubric to assess the team involvement.
5. Majority of our upper level classes are now required to write and present research papers. We are emphasizing the importance of proper formatting, citations, and reference. We are also emphasizing the dangers of plagiarism in all our classes.

Assumptions for Columns 4 in the above table

1. It is assumed that if a student received more than 70% for an outcome then he/she has met the requirements for that outcome. For example if a student has received an aggregate of more than 70 % for the program development (outcome 1) then that student has met the requirements for outcome 1.
2. It is assumed that if more than 80% of the students in a class met the requirements for an outcome, then that class has met the requirements for that outcome. For example if more than 80% of the students in Visual Basic Class (CIS 119) have met the outcome 1 requirements, then the whole class has met the requirements for outcome 1. Only the students who completed the course is included in this analysis

Wiki Mass Authoring for Experiential Learning: A Case Study

Harold Pardue
hpardue@southalabama.edu

Jeffrey Landry
jlandry@southalabama.edu

Bob Sweeney
bsweeney@usouthal.edu

School of Computing
University of South Alabama
Mobile, AL 36688, USA

Abstract

Web 2.0 services include sharing and collaborative technologies such as blogs, social networking sites, online office productivity tools, and wikis. Wikis are increasingly used for the design and implementation of pedagogy, for example to facilitate experiential learning. A U.S. government-funded project for system security risk assessment was conducted using a wiki powered by MediaWiki. Participants were geographically disbursed students, faculty, and industry partners with highly diverse backgrounds and expertise. The focus of this research was the experiential learning practiced by students carrying out the work of the project. Through the use of a wiki as a mass authoring tool, students constructed knowledge in the form of an annotated bibliography of extant systems security literature. Results from a student survey offered convincing support for the use of the wiki's influence on students' experiential learning, particularly through the benefit of observation and reflection, as well as the motivational influence of social norms. Lessons learned and possible extensions of the approach described in this study to other educational settings are discussed.

Keywords: wiki, experiential learning, mass authoring, social norms

1. INTRODUCTION

The proverb, "experience is the best teacher" has been reincarnated in multiple forms by various authors since Julius Caesar (52 B.C.) recorded the earliest known version; "experience is the teacher of all things." The idea that one's experiences can be incorporated into a formal educational approach is the foundation for experiential learning theory (Kolb, 1984). The professional baseball player Vernon Law (n.d.), with his humorous rewording of the proverb as, "experience is a hard teacher because she gives the test first, the lesson afterwards" reinforces the idea that a feedback loop is an integral part of the experiential learning process. A problem faced by educators seeking to incorporate experiential learning into classroom activities is a lack of tools to support their inclusion; however, the introduction and use of Web 2.0 technologies has opened new avenues of instruction which were previously unavailable. New pedagogical models founded on experimental learning

theory and supported by Web 2.0 mass-authoring and social-networking tools have emerged (Huang and Behara, 2007).
Web 2.0, a term coined as a result of a brainstorming session between O'Reilly and MediaLive International, encompasses the use of World Wide Web technologies which seek to improve web users' creativity, collaboration and sharing, and communications. The underlying core competencies listed on O'Reilly's Web 2.0 Meme Map include "services, not packaged software; architectures of participation; cost-effective scalability; remixable data source and data transformations; software above the level of a single device; and harnessing collective intelligence" (O'Reilly, 2005). These competencies represent the common features and characteristics of Web 2.0 web services and technologies.

Web 2.0 services and technologies include web logs (blogs), video-sharing and social networking sites, online productivity applications, and wikis. One of the most widely used of these Web 2.0 technologies is the wiki--a web site designed to create a collaborative working environment or knowledge management community.

The remainder of this paper is organized in the following manner. First, we provide a brief literature review on the use of wikis in education and experiential learning theory. Next, a case study of a United States (U.S.) government-funded project for system security risk assessment is described where mass authoring was conducted through the use of a wiki. The results of a survey of student participants are described and discussed. Finally, we discuss the lessons learned and possible extensions to the approach used on this project.

2. BACKGROUND

The use of wiki technology as a teaching tool is well documented in the literature (Bergin, 2002; Bower et al, 2006; Konieczny, 2007; Parker and Chao, 2007).The first published use of wiki technology in education was the CoWeb wiki built in 1997 by researchers at the Georgia Institute of Technology (Leuf and Cunningham, 2007). This wiki became a standard part of course delivery at Georgia Tech and has been adopted by many other universities. The adoption of wikis in education has grown dramatically since 1997 if the number of publications related to wiki technology in education is a valid indicator. A Google Scholar search of the terms "wiki education" returned 83 results for the year 1997 and 15,400 results for 2009 (see Figure 1). For a sample of recent publications, see (Banks et al, 2010; Chidanandan, 2010; Every et al, 2010; Hastie et al, 2010; Meishar-Tal and Gorsky, 2010; O'Connor, 2010; Walsh, 2010).

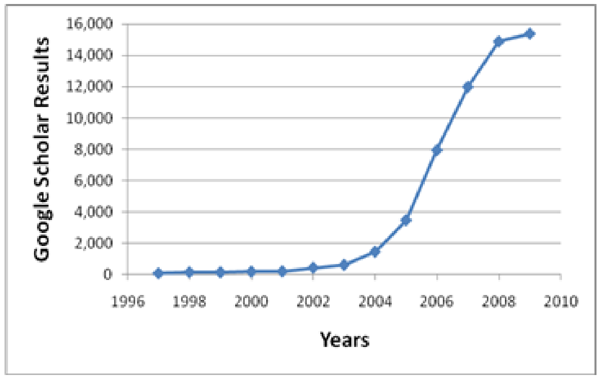


Figure 1. Google Scholar Results for "wiki education"

Because of the wiki's wide, transparent, and easy access (Leuf and Cunningham 2001), experiential learning theory is an excellent theory to explore the benefits of the wiki as a facilitator of learning. Kolb (1984, pp. 41) defines learning as, "the process whereby knowledge is created through the transformation of experience." This transformative experience view of learning is likewise supported by Kaagan (1999). Kolb (1984) derived his experiential learning concepts from the learning models developed by Lewin, Dewey, and Piaget.

Experiential learning is learner-centered, relying on learning from experience, rather than teacher-centered, emphasizing content delivery. Furthermore, experiential learning involves tangible learning activites as opposed to merely abstracted knowledge exposure (Pimentel, 1999). Learner-centered teaching emphasizes a coach-facilitator role for the professor, rather than information giver, and emphasizes learning from mistakes rather than assessment of right and wrong answers (Saulnier et al, 2008). The learner-centered paradigm is important for IS education for a variety of reasons (Landry et al, 2008). Most notably, the current project's use of learner-centered, that is, experiential learning, should develop the students' learning-related skills important to the dynamic IT profession. These skills include technology evaluation, innovation adoption, and lifelong learning.

3. CASE STUDY: MASS AUTHORING AN ANNOTATED BIBLIOGRAPHY

The U.S. government-funded system security risk assessment project used a wiki, loosely referred to as "the project wiki" as a basic Intranet-based project portal. The wiki served as an information repository for the project team and sponsor, featuring the project solicitation, proposal, plan, team directory, and other content. This case study describes the use of the project wiki for creating a mass authored annotated bibliography. The case illustrates how the wiki supported an approach that used experiential learning concepts. Each of four subsections of the case contrasts the collaborative, wiki-based experiential approach to an issue with a more traditional educational approach considered. A fifth subsection summarizes the major contrasts in the educational approach used. Perceptions and interpretations of the faculty participants (co-authors) are interspersed with feedback from a student survey given at the end of the project.

Working with students—group collaboration vs. independent research

Once the university was awarded the government systems security risk assessment project, the project's principle investigator (PI), who also serves as dean of one of the university's colleges, assembled a team of three faculty members, all professors of various ranks, for a meeting. The PI wanted to get a fast start on the first of several project phases. The goal of this phase was to develop an annotated bibliography of literature relevant to the project. The bibliography the group was to assemble dealt with government systems security, and consisted of publication sources that included government reports, scholarly articles and books, newspaper articles, and political activist web sites. The government agency sponsoring the contract agreed to fund a team of ten students to work on the project. Although the PI was pleased with the resources obtained and opportunities for students, the three professors had reservations about managing so many students on a project with such a large scope. Their main concerns were with the amount of time required to train and oversee the mixture of undergraduate and graduate students with varied backgrounds, and whether these students were capable of meeting the quality demands of the sponsor.

The mental model used by the professors was one of mentored, independent research. Such a formal approach, used in dissertations, theses, directed study projects, and term papers, features heavy guidance and control by a professor as authority with a view towards creating an independent researcher producing high quality work. The student is responsible for the whole project, but progresses under close scrutiny by the professor. No work is released for public consumption unless the very high quality standards set by the professor are met, and the student has to rigorously defend every decision made. Mentored research had required a large commitment on the parts of both the professor and student.

The PI had a more collaborative approach in mind, and suggested a different kind of mentoring approach, using a wiki for support. There were more students than professors, so a one-on-one mentoring model would be too time-consuming for professors, and there was insufficient time to conduct multiple interdependent projects and then integrate them into a coherent whole. The PI suggested a less formal, more collaborative, relationship among professors and students. He viewed the students as co-collaborators that could be quickly trained and would help out in various ways as needed, but not necessarily equally in effort or result.

The professors were unsure if enough qualified students could be found to fit into such a collaborative model, and if a minimal training approach would adequately prepare students for the task. The professors left the meeting apprehensive, but proceeded with the task of recruiting students. The professors recruited students from late afternoon classes for ten-minute interviews, and in just over two hours, ten students were successfully selected.

Training student annotators—experiential learning vs. structured guidance

The initial project goal was to complete a detailed literature search that would provide the foundation for all further project activities. The PI thought that an annotated bibliography consisting of about 200 entries would be more than adequate. The schedule goal was an ambitious three weeks. The PI was already familiar with what articles needed to be annotated, and he called on a team of colleagues to generate a list of articles. One of the professors took charge of this phase of the project, and was encouraged to use one of the students to post to the list of articles and perform additional keyword searches to find more. A professor selected an experienced graduate student for this task and began directing him.

As for writing the annotations on the articles, the PI suggested that any of the remaining students could accomplish the work with little guidance. The PI suggested a training approach in which the professors and students would separately write an annotation for the same article, and compare results. The students would compare their annotation to the professor's and vice-versa. The professors would give helpful feedback, and if necessary, repeat the process on a second article. The combination of experience and observation and reflection—the first two stages of the experiential learning model, would guide the students in the learning process.

Still unsure of whether this could be accomplished expeditiously with the whole group, the professors decided to put this approach into practice, using a group training session. Although the wiki could have been used at this point, the large, face-to-face meeting provided a richer context for group *forming* (Tuckman and Jensen, 1977), in which groups are initiated and introduced. A distinct forming stage is even more important for virtual groups who potentially will complete much of the work using lean technologies. At the training session, held the following day, the professors followed the PI's advice and kept the project overview brief. One professor gave an explanation of the main goals of the project and how to write an annotation. Another professor distributed a 12-page seminal article and assigned each person in the group, including the professors, the task of reading the article, then writing a 100-200 word annotation that covered key points of the reading while emphasizing its importance to the project. The guidance was to make evaluative comments in addition to writing an objective, abstract-like summary. Following the collaborative approach, the students were instructed to learn by doing. Some of the annotators worked in the training room, but most left and then sporadically returned until all had reassembled. Copies of each annotation were distributed to all participants. It was important to observe the performance of the others in the group, as an additional source of experiential learning.

Having received all of the annotations, the professors sought to provide the students with feedback. One of the professors read and assessed the annotations submitted by the students. He provided summary feedback, specific guidance on how to write the remaining bibliographic entries, and remarks intended to provide encouragement. The professor pointed out examples of phrases thought to be good evaluative comments by quoting from the articles, and encouraged everyone to write similar comments. The professor then fielded questions. In less than 90 minutes, the professors had trained—in an experiential, collaborative manner—all ten students.

The professors wrote a half-page of guidance for the group, based on what was learned from the meeting, and posted this guidance on the wiki. The guidance included the one-hour approximate time to complete one annotation, the encouragement to write evaluative comments, the instruction to write 5-10 keywords to describe the reading, and the suggestion of using their own judgment to add any cited readings to the bibliography. The students were guided by their own experience, by observing others, and finally, by informal and formal feedback given to the group. The students did not receive individual feedback, nor were they "graded" individually, as students would be in a traditional classroom environment. The written guidance posted on the wiki would end up being detailed and structured, but the structure and detail came as a result of experiential learning and was written after the shared experience, not before.

The professors were more encouraged, but not sure what to expect when they assigned students the articles, books, technical papers, and political activist content to be annotated. The students would be self-managing their time and work, and using the project wiki.

Posting to the wiki—autonomy vs. authority

After the training session, the students and professors were assigned articles whose citations had been posted on the wiki. Each person was assigned four or five articles at a time, and given more after completing their current batch. The professors questioned whether the annotators should post articles directly to the wiki, or if the annotators' intermediate work should be reviewed first. Taking advantage of the ease with which the wiki enabled students to post content, and the collaborative approach encouraged both by the PI the students directly posted annotations. One of the professors reviewed and edited annotations as needed, which primarily involved fixing typos. The professor rarely edited content, resisting the tendency to overly control as a formal authority mechanism, and gave the students autonomy. Unfortunately, the students were unable to review each other's entries as well due to time constraints. When the quality of annotations seemed adequate, the reviewing professor found it acceptable to cease reviewing the entries altogether.

The very nature of the wiki as a tool that is easy to use for uploading and tagging web page content, emphasized collective effort without authoritative control. The wiki provided an easy way to upload annotations, and also to support the editorial process with submission and review tasks. See Figure 2 in the appendix.

The wiki allowed the professor to instruct students to upload their work directly rather than e-mailing documents for review. Direct uploading meant more trust and less control, and more efficiency. The wiki also maintains a page edit log with user name, date of change, and summary of change information. If mistakes or corrections were made, the history could be traced or even undone. By keeping track of edits, collective ownership rather than individual ownership of work was encouraged.

The professors considered the quality of the work to be very good. The professor who was the primary contact person for student questions on writing annotations reported that the most common questions asked by students, via e-mail and face-to-face, were about exceptions. For example, several students reported that they had been assigned articles much longer than the practice article. Other students asked about annotating articles that were of a different style, such as an activist web site with links to many articles rather than a single academic article. The professor reported that these questions were easy to answer, because he had himself experienced the activity, and had seen many examples of what the students could do. The professor, in following an experiential learning paradigm, did not want to give a lot of guidance, but wanted the students to reflect on experience. Invariably, the professor's responses fit the form of "look at so-and-so's example on the wiki for a model of how to do it." The professor also forwarded individual responses to the group for guidance, and thought that the responses might even be something that could be posted to the wiki itself in the guidance section.

Staying on task—virtual vs. proximate

Although the quality of work using the virtual, collaborative model was very good, the quantity was not. The use of the wiki alone was not effective in maintaining a rapid pace of work. Obstacles to learning experienced by students included time pressure and difficulties reading the lengthy articles while keeping the abstract short. As the project was taking place during the academic year, the students and professors were distracted by coursework demands and teaching assignments. Students, consequently, tended to work in sporadic intervals. Mass e-mails were sent out with status updates on the number of articles reviewed ("82 articles have been posted and 66 of them reviewed and tagged") and intermediate goals ("let's reach the 100 mark by Friday—yes we can do it!"). The use of the wiki was also supplemented by face-to-face meetings, which kept the students and professors involved in the ongoing progress of the project, and made them more personally connected. The goal of annotating 200 articles in three weeks was not met, however progress was made. Although status updates and meetings were needed to maintain pace, the use of these tactics was believed to be less than if there were no wiki. In fact, students reported that they were motivated by observing the posted entries of their fellow students.

Mass authoring process—a summary

At each step of the mass authoring process, the professors challenged traditional mindsets, opting for a newer, more collaborative and experiential approach. In working with students, the professors opted for more group collaboration over independent research. In training student annotators, the professors allowed experiential learning to flourish, while refraining from excessively structured guidance and only then to document the collective learning experience. The process used to post to the wiki granted autonomy that superseded unnecessary authority. Despite the virtual environment of the wiki, the use of proximity was still needed. However, proximity, in the form of meetings, was for connecting with students to keep them on task, and not for task guidance, task delivery, or quality control.

4. STUDENT FEEDBACK

In order to test students' perceptions of the wiki for experiential learning, the authors constructed a 15-item questionnaire, approved for use with human subjects by the university's institutional review board, and administered the questionnaire to the students at the conclusion of the project in March 2010. Eight of the 11 student participants completed the survey, consisting of 12 close-ended and three open-ended items. The close-ended items used a 7-point, disagree-agree scale. See Table 1 in the appendix for a complete summary of the results for closed ended items. The students who completed the survey were all employed on the government-funded project. One was an undergraduate computer science major, another was a recent alumnus of the master's program in information systems, and the others were all currently enrolled in the information systems master's program. There were two females and six males, and seven of eight were international students.

We interpreted the results in terms of the experiential model of learning. See Table 2 in the appendix for a summary. Although these are not intended to be constructs, we wrote and organized the survey items in terms of: *observation, structured guidance, learning appropriations of the wiki, social influence, and obstacles*. *Observation* is the degree to which the wiki facilitated experiential learning by allowing students to observe the work of other students. *Structured guidance* is the degree to which learning came from guidance and feedback from the professors as authorities as opposed to unstructured and experientially from the collaborative use of the wiki. *Learning appropriations of the wiki* is the degree to which wiki features were used by students to support experiential learning. *Social influence* measures whether observations of other student's work motivated and inspired higher quality outcomes. As the project wiki is not assumed to be a technological "silver bullet," we asked students about o*bstacles*, which measure the influence of difficulties encountered using the wiki on learning outcomes.

From our analysis of the results, only one item appeared to confuse the study participants. Apparently the term "features" in item 6 was confusing because even those that strongly disagreed with the close-ended item 6 supplied an answer to 15. One of the strongly disagreeing (value=1 on item 6) students actually admitted that the "chance to look at the entries done by the other members and could get an idea of writing entries" helped him learn, but said the project "wiki had no specific features of helping us in writing a bibliographic entry." The other strongly disagreeing student stated that posted guidance helped him learn, but that that was not a feature. Both, in fact, are features as intended by the item.

5. DISCUSSION

A major factor in the success of the experiential learning process was that the acquisition and creation of knowledge was social and situational. That is, the wiki enabled the effort to be truly collaborative. By widely dispersing the task assignments, doing very little editing and reviewing, and posting as you go, a collective "intelligence" emerged from appropriating the wiki for experiential learning. The relevance of the literature to the project was a function of the collective efforts, and did not simply feed downward from the PI as project visionary. Through a form of Web 2.0 mass authoring (wiki), the meaning and relevance of each article was implicitly negotiated by members of the social group through cycles of reviews and revisions. Further, the wiki became a social resource or social accumulation of knowledge. An implication is that teachers should embrace and foster a sense of student ownership of the process and results.

The shared experience was useful and effective for reasons pointed out by the experiential learning model. The students were trained as much by the concrete experience of writing an annotation as they were by guided instruction, which was minimal. The students, in fact, did not receive very much feedback. The results of this case study supports a limited, efficient, one-hour training, and then learning by doing, on the wiki, providing more opportunities for concrete experience, focused awareness, observation, and testing. An implication for teachers is to reduce the instructive component, and feedback, while enabling collaborative, shared learning experiences instead.

The use of the wiki enabled the establishment of social norms. In particular, students and faculty had to develop a working level of trust within the social group. Because each person's work was essentially public and under scrutiny, there had to evolve a tacit social norm that no one's work would be "attacked" or ridiculed. In the initial meetings, there was a general sense of apprehension about having one's work undergo public review. But soon it became apparent that everyone was learning from the experience and that no one was writing "perfect" annotations. In fact, everyone was learning from each other's struggles with the annotation process. Mass authorship is not simply a technical or technological process; successful mass authoring requires inter-personal trust. An implication of this is that the teacher must recognize, plan for, and foster the development of trust as part of the learning process.

The experiential learning process was fluid and dynamic. What we learned from this experience is that the process is very difficult to conduct 100 percent online using only a wiki. Periodic face-to-face meetings were essential. We needed pep talks to keep students motivated, face-to-face meetings when productivity flagged, and meetings where students could be reminded of the larger goal and the alignment of their efforts with this goal.

We learned best practices for use of meta-tags in the wiki. Firstly, we use meta-tags to construct relationships among content on the wiki, meta-data about the data. We used meta-tags for keywords, author names, and other bibliographic content. This use of the meta-tags enabled students to form abstractions and generalizations beyond the individual bibliographic annotations they were writing. Secondly, we used meta-tags to manage work. Tags were effectively used to identify the responsibilities for bibliographic entries still requiring an annotation.

Finally, we learned that we should have incorporated use of the wiki in our training sessions. Had the team installed and had the wiki operational at the training meeting, it would have facilitated the student's posting of training entries. This approach is recommended for research-based experiential learning.

A future direction for this study is the generalization of the process to other educational situations and courses. We believe this approach can be extended to, for example, cultural immersions, professional practice, and service learning/civic engagement. See Ithaca (2007) for a discussion of experiential learning in other educational contexts.

In thinking about other courses in the IS curriculum, the success of our project suggests that the use of wikis could be effective at augmenting traditional classroom delivery of content and extend the functionality of current online offerings. Wikis could be used to create experiential learning processes within a systems analysis and design or database course. In addition to formal education in Entity Relationship modeling, students could be given a concrete example to work with and learn from and then work collectively to review and revise homework problems.

A wiki-based experiential learning process could enhance the core IS survey course by having students construct and maintain a course Wikipedia (Kane and Fichman, 2009). Students would directly experience the use of information technology to improve the performance of people in organizations (McNurlin, Sprague, and Bui, 2009). In this course, performance would be related to the acquisition of knowledge as measured on exams and assignments.

6. CONCLUSION

The success of our system security risk assessment project depended initially on the professors' ability to rapidly educate a loosely coupled, diverse group of students on how to create an annotated bibliography of a very technical and rich body of literature. The scope and scale of the task was tantamount to the initial literature review for a PhD dissertation. However, there was no time for months of formal education on research methods. The students needed to begin work immediately. The students were presented with a brief overview of the concepts and process prior to giving them a concrete example from which to work and learn. Wikis were then used to create an environment where each student could directly experience the process of creating an annotated bibliography by performing the work collectively as a member of a social group or network.

The results of this case study suggest that wikis can be successfully used to facilitate experiential learning of a mass authoring task in a time-pressured environment requiring high levels of quality by enabling collaboration and the establishment of social norms. An additional finding is that teachers, although not needed as much, were not totally removed from the learning process once the students were engaged in experiential learning. Experience is not, as it turns out, the teacher of all things—teachers still have a role in experiential the learning process.

7. REFERENCES

Banks, D., Stewart, C., Gendron, M. (2010), “An Ad-Hoc Collaborative Exercise between US and Australian Students Using ThinkTank:E -Graffiti or Meaningful Exchange?” The Journal of Issues in Informing Science & Information Technology, 2010, Vol. 7, pp. 133.

Bergin, J. (2002) “Teaching on the wiki web”, 7th annual conference on Innovation and Technology in Computer Science Education ITiCSE (Aarhus, Denmark), (2002) pp. 195-195.

Bower, M., Woo, K., Roberts, M. and Watters, P. (2006), “Wiki pedagogy: a tale of two wikis”, International Conference on Information Technology Based Higher Education and Training, 2006, pp. 191-202.

Chidanandan, A., Russell-Dag, L., Laxer, C., and Ayfer, R. (2010) “In their words: student feedback on an International project collaboration”, Proceedings of the 41st ACM Technical Symposium on Computer Science Education (Milwaukee, Wisconsin, USA, 2010, pp. 534-538.

Every, V., Garcia, G. & Young, M. (2010) “A Qualitative Study of Public Wiki Use in a Teacher Education Program. In C. Crawford et al. (Eds.)”, Proceedings of Society for Information Technology & Teacher Education International Conference,2010 pp. 55-62.

Forte, A. and Bruckman, A. (2007) “Constructing text:: Wiki as a toolkit for (collaborative?) learning”, Proceedings of the 2007 international Symposium on Wikis, pp. 31-42.

Hastie, Peter A., Casey, Ashley, Tarter, Anne-Marie (2010), "A case study of wikis and student-designed games in physical education" Technology, Pedagogy and Education, 2010, Vol. 19, No. 1, pp. 79-91.

Huang, C. D. and Behara R. (2007) "Outcome-Driven Experiential Learning with Web 2.0," Journal of Information Systems Education, Vol. 18, No. 3, pp. 329-336.

Ithaca College of Humanities and Sciences (2007) “Appendix B: Models of Experiential Learning”, in A Framework for Experiential Learning in the School of Humanities and Sciences, March 2007, Retrieved June 27, 2009 from http://www.ithaca.edu/hs/resources/docs/explrngdocs/elframeworkdoc/

Kaagan S.S. (1999) Leadership Games: Experiential Learning for Organizational Development. Sage, Thousand Oaks, CA.

Kane, G. C., and Fichman, R. G. (2009) “The Shoemaker’s Children: Using Wikis for Information Systems Teaching, Research, and Publication,” MIS Quarterly, Vol. 33, No. 1, pp. 1-17.

Kolb, D. (1984) Experiential Learning. Prentice Hall, Englewood Cliffs, New Jersey.

Konieczny, P. (2007) “Wikis and Wikipedia as a teaching tool”, International Journal of Instructional Technology and Distance Learning, 2007 Vol. 4 No. 1, pp.15-34.

Landry, J. P., Saulnier, B. M., Wagner, T. A, and Longenecker, Jr., H. E. (2008) "Why Is the Learner-Centered Paradigm So Profoundly Important For Information Systems Education?” Special Issue on IS Education Assessment, Journal of Information Systems Education (JISE), 2008, Vol. 19, No. 2, pp. 175-179.

Leuf, B. and Cunningham, W. (2007) The Wiki Way. Addison-Wesley, Boston.

McNurlin, B. C., Sprague, R. H., and Bui, T. (2007) Information Systems Management in Practice. Prentice Hall, Upper Saddle River, New Jersey.

Meishar-Tal, H. & Gorsky, P. (2010) "Wikis: what students do and do not do when writing collaboratively. Open Learning", The Journal of Open and Distance Learning, 2010, Vol. 25, No. 1, pp. 25-35.

O'Connor, E. (2010) "The Use of a Wiki in Teacher Education: How Does Learning and Instruction Change When Work Can Go Public?" In C. Crawford et al. (Eds.), Proceedings of Society for Information Technology & Teacher Education International Conference, 2010, pp. 2822-2829.

O'Reilly, T. (2005) What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software, Retrieved January 15, 2009 from http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/20/what-is-web-20.html

Parker, K. R., & Chao, J. T. (2007) "Wiki as a teaching tool", Interdisciplinary Journal of Knowledge and Learning Objects, 2007, Vol. 3, pp. 57-72.

Pimentel J. R. (1999) Design of Net-Learning Systems Based on Experiential Learning. Journal of Asynchronous Learning Network Vol. 3, No. 2, pp. 64–90.

Saulnier, B. M., Landry, J. P. Wagner, T. A., Longenecker, H. E. (2008) "From Teaching to Learning: Learner-Centered Teaching & Assessment In Information Systems Education", Special Issue on IS Education Assessment, Journal of Information Systems Education (JISE) Vol. 19, No. 2, pp. 169-174.

Tuckman, B. W. and Jensen, M. A. C. (1977) "Stages of Small-Group Development Revisited," Group Organization Management, Dec. 1977, Vol. 2, pp. pp. 419 - 427.

Walsh, Lynda (2010) "Constructive Interference: Wikis And Service Learning In The Technical Communication Classroom", Technical Communication Quarterly, 2010, Vol. 19 No.2, pp. 184-211.

Editor's Note:

This paper was selected for inclusion in the journal as a ISECON 2012 Distinguished Paper. The acceptance rate is typically 7% 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 2012.

Appendix

Figure 2 - Editorial Process

USING WIKIS TO MIMIC AN EDITORIAL PROCESS

Wikis have a feature called Categories. You can add a tag on a page just by entering something like this: [[Category : *Category_name*]]. The phrase *Category_name* then shows up in a list on a page with other category names that were tagged elsewhere. The category names are listed in alphabetical order. Clicking on any of the category terms brings you to a page that lists all wiki pages with that category tag. This is a way of indexing pages in various ways.

In the systems security risk assessment project, category tags were used on the annotated bibliography for assigning and reviewing articles. Using category tags served as a very easy means of mimicking an editorial process for the mass authoring project, supporting the interrelated roles of editors and authors.

To assign an article, a professor, acting as editor, would add the tag.

[[Category: Assigned *Last First*]], for example [[Category: Assigned Smith John]]

, to a page containing a citation and blank annotation block.

The students and professors authoring an annotation would go to the Categories page to find their assignments. For example,

- Assigned Doe Jane (2 members)
- Assigned Smith John (4 members)
- Assigned Thomas Jim (1 member)

The "member" refers to the number of pages with the tag, and thus, the number of articles assigned. Annotators were instructed to edit the tag when the uploaded their annotations, changing the category tag to [[Category: Summary written]].

This action would take away one member from the "Assigned" tag that was removed and add one to the "Summary written" tag. The professor as editor, making a review, would then access the "Summary written" page, by clicking on the hyperlink on the Categories page, to access all of the annotations available for review, changing the "Summary written" tag to "Summary reviewed." Once all 140 articles were uploaded and reviewed, the summary review category had 140 members and appeared as follows on the Category page:

- Summary reviewed (140 members)

Table 2 - Student Survey Results for Closed-Ended Items

Item	Disagree 1	2	3	4	5	6	Agree 7	Mean
1. I learned to write entries through my own experience of writing my entries.	2		1	2		1	2	4.13
2. I learned to write entries by observing the wiki entries written by others.	1					3	4	5.88
3. I needed better feedback from professors on my entries.	2	1	1	1	2		1	3.50
4. I needed to see more examples of entries written by others	1	1	1		2		3	4.63
5. I needed more guidance by professors up front on how to write entries.	3	1	1		2		1	3.13
6. The features of the wiki tool helped me learn to write entries.	2				2	1	3	4.88
7. Seeing my entries posted on the wiki got me more engaged in the project.	1			1	2	1	3	5.25
8. Knowing that other students and professors would read my entries motivated me to do a better job.	1					2	5	6.00
9. Seeing the entries of others gave me concrete examples from which to learn.						3	5	6.63
10. Seeing the entries of others inspired me to work harder.		1			1	2	4	5.88
11. Seeing the entries of others gave me confidence that I could do it too.	1			1		2	4	5.63
12. I primarily learned to write entries from the initial 1-hour bibliographic training.				5	1		2	4.88

Table 3 - Interpretation of Results

Item Composition	**interpretation**
Observation Items 2 (5.88) and 9 (6.63)	The data provide convincing support for the wiki's facilitating effect on experiential learning. Students learned through observation and reflections, which lead to the formation of abstract concepts and generalizations in the Lewinian model.
Structured guidance Items 3 (3.50), 5 (3.13), and 12 (4.88)	The moderate to low numbers on the structured guidance items provides evidence that suggests students learned as much or more in the collaborative wiki environment through observation, reflection, and social norms, rather than from the structured guidance coming from professors.
learning appropriations of the wiki Items 6 (4.88), 15 (N/A)	Modest but very positive agreement to item 6 and the open-ended responses suggest the wiki helped participants learn. The most common features listed were *access to the entries of others* and the *guidance posted on the wiki by professors*. The evidence, including positive responses to item 6, and open-ended responses to item 15 that included two responses contrary to the negative responses to item 6, support learning appropriations of the wiki. Specifically, appropriations of the wiki to access the entries of others and to seek guidance posted on the wiki by the professors support learning appropriations of the wiki.
Social Influence Items 10 (5.88) and 8 (6.00)	There was strong agreement among participants that the public nature of the wiki inspired and motivated them. The data support the idea that influential peers and authority figures can influence an individual's intentions, self-efficacy, and behavior, according to attitudinal theories of motivation. Furthermore, the evidence in this case supports the idea that behavioral norms can emerge from collective use of a technology.
Obstacles Items 13 (N/A) and 14 (N/A)	The primary obstacle reported was struggles with producing quantity while maintaining a high standard of quality on articles that varied widely in length, format, and difficulty. Similar to the findings of Forte and Bruckman (2007), students sometimes had difficulty navigating links on the wiki and trouble with figuring out the right format for a given bibliographic entry, given apparent inconsistencies between entries seen, and between articles and books, for example. Responses to item 14 support the idea that instructors, although not needed as much, were not totally removed from the learning process once the students were engaged in collaborative learning because nearly half the participants listed “asking professors for help”.

Information Systems Curricula: A Fifty Year Journey

Herbert E. Longenecker, Jr.
longeneckerb@gmail.com

David Feinstein
dfeinstein@usouthal.edu

School of Computing,
University of South Alabama
Mobile, AL 36688, USA

Jon D. Clark
jon.clark@colostate.edu
College of Business
Colorado State University
Fort Collins, CO 80524-1277, USA

Abstract

This article presents the results of research to explore the nature of changes in skills over a fifty year period spanning the life of Information Systems model curricula. Work begun in 1999 was expanded both backwards in time, as well as forwards to 2012 to define skills relevant to Information Systems curricula. The work in 1999 was based on job ads from 17 major national newspapers. The ~3000 ads enabled generation of 37 skills and defined major areas of skills: software development, web development, database, operating systems and telecommunications, strategic organizational development, interpersonal and team skills, and project management. During the development of this research a ninth skill area was added: information and security assurance. The original 37 skills had been expanded to 69 skills, and within this effort, 69 additional skills were added. Analysis of the skills as of today suggested elimination of retired (24) and too new (13) skills. Of the remaining skills a set (35) of skills was common to all curricula, a large set of current skills (64) was abandoned by IS 2010 which added new skills (2). Deletion of programming as a requirement of IS 2010 accounts for a significant proportion of deletions.

Keywords: Information Systems Curriculum, IS 2002, IS 2010, AITP Model Curriculum, ACM Model Curriculum

1.0 Introduction

The ACM, the Association for Information Technology Professionals (AITP), formerly the Data Processing Management Association (DPMA), and more recently the Association for Information Systems (AIS) have taken the task of developing curricula for information systems for the past fifty years.

By the late 1960s it was recognized that computers were going to play a very important role in business and industry, and there would also be a need for a highly trained work force. Likewise, it became apparent that there were significantly different interests in the nature of the needs for academic curricula.

Initial studies funded by the NSF were carried out by an ACM committee. The committee became known as the Committee *on Computer Education for Management*. Its work, *"Curriculum Development in Management Information Systems Education in Colleges and Universities."* was published in November 1965 (ACM, 1965). It became very clear to the committee that considering the program as an extension of the computer science curriculum would not solve the problem.

The ACM developed Curriculum 1968 for Computer Science (ACM, 1968), and a different group within the ACM developed model curricula for information systems for graduate programs (Ashenhurst, 1972) and for undergraduate programs (Couger, 1973).

Teichroew (1971) originally identified the need for information systems professionals: he cited department of labor figures that there would be a need for systems analysts (165,000) and computer programmers (154,000) over the subsequent five year period extending through the mid-1970s. He also established that there was inadequate preparation by academia, and a lack of clarity in communicating the needs expressed by business managers. He further recognized there was a lack of agreement regarding an expressed body of knowledge. It was further recognized that much of the education relating to the business application of computing was being provided by vendors.

All of the IS curriculum models have as a common goal to provide advice for college and university faculty that will guide the preparation of graduates. These graduates will be better prepared to enter the work force successfully. To clarify the expectations for graduates, all of the models present a conceptual framework of the exit level characteristics. Ashenhurst (1972) describes the characteristics of people, models, systems, computers, organizations, and society.. Appendix Table 1 is a presentation of this material. You will notice the use of learned capability verbs (knowledge, ability) to explain the depth of knowledge expected. Similar writing has been used in the subsequent models expressing as learning outcomes.

In this paper, our goal was to find a way to compare the model curricula. While the goals of the curricula may seem very similar, the detailed skill requirements have evolved over 50 years. Some issues remain unchanged: for example, programmers and analysts are still outputs from most of the model curricula. Likewise, the demand for our graduates has remained high since the very beginning of the discipline. While some of us may remember a heavy focus on accounting information systems, since the beginning there has been a much broader organizational focus. Dramatic changes occurred with the introduction of machines: mini-computers in the 1970s, pcs in the 1980s, and, more recently, PDAs. This brought computing into the realm of almost everyone in the developed world. The technology also went from individual stand along machines in the 50s and 60s to the ubiquitous connectivity we have today. These changes have had dramatic impacts on the information systems community on how we develop and deploy systems today.

Yet, because of the focus of information technologies enabling people to do their work, and thereby creatively support organizational success we find many similarities over the fifty year span. In order to try to find a consistent way to compare the various models we decided to examine the skills enabled by the various models. Haigood(2001), Landry(2000) , and Colvin (2008) examined approximately 3000 job ads in 17 major national newspapers. Using a qualitative research technique of aggregating skill words associated with the ads and produced a list of about 50 skills. Surveys of faculty members in the US along with a factor analysis revealed that 37 of the skills were related to 8 factors. As it turns out, the 8 factors related to the exit characteristics expressed in IS'97 (Couger et al, 1997). These skill categories included: software development, web development, database, operating systems and telecommunications, strategic organizational development, interpersonal and team skills, and project management. The same 37 elements were re-surveyed in 2007 and were found to be almost identical in identifying the same 8 factors, and skill depths found (Landry, 2000; Colvin 2008). The skills are not product specific; but rather they are general in nature.

Since 2000 we have updated these skills with increasingly more current information by analysis of the curricula of IT (2008) and the ACM/AIS groups. We also have added material based on the DAMA body of knowledge (Henderson, et al 2004; Longenecker, et al, 2006), and the Department of Defense NICE specification (NICE, 2010, 2012). As a direct result of studying the NICE skill set, we added a skill category in Information Assurance and Security. . The result of these additions produced an expanded list with 69 skills compatible with current documents.

2.0 METHOD OF STUDY

The following model curricula were included in this study: IS'72 (Teichroew, et al, 1971; Ashenhurst, 1972; Couger, 1973), IS'81 (Nunamaker, et al, 1982), DPMA '86 (DPMA, 1986), IS'90 (Longenecker, et al, 1991a, 1991b, 1991c), IS'97 (Longenecker et al, 1995; Gorgone, et al, 1995; Davis, et al 1997; Couger 1997), IS 2002 (Gorgone, et al, 2003), and IS 2010 (Topi, et al 2010). In our work here we did not report separately on IS'97 since it was totally included within IS2002.

For all elements mapped we used the skill depth criterion established in IS'90 (Longenecker, et al, 1990) and utilized in subsequent models. The criterion was based on the Bloom (1956) task force which described a taxonomy of cognitive skills. The Bloom task force was trying to utilize a uniform method for describing learning objectives. The IS'90 modification to Bloom's taxonomy split the first level into two categories:

Level	Meaning
1	Recognize
2	Differentiate
3	Use (or translate, explain)
4	Apply (without direction or hints)

IS'97, IS 2002, and IS 2010 present a table that contains examples of the use of learned capability verbs characteristic of knowledge specifications at each level. We utilized our understanding of these tables to grade the complexity of the curriculum learning outcomes, or equivalent.

Not all items mapped exactly, so we found it necessary to add new skills to our initial list. The revised list added 69 newly written skills giving a total of 138. Skills were added to this list because of a lack of fit to the original 69 skills. As new skills were added, we went back through all of the curriculum models used and made additional mappings for the newly abstracted skills if that seemed necessary.
A criterion for mapping a skill to a curriculum was that there had to be a learning outcome (or equivalent, e.g. learning unit) that specified the requirement for that skill as a course outcome. That is, general pronouncements about the nature of the curriculum were not used, unless they were codified in learning outcomes.

3.0 RESULTS

Appendix 2: Table 2 and Appendix 3: Graphs summarizing the same information is summary of all of the curriculum mappings to the skill sets. The left set of columns show the mapped result from the various curricula reviewed. The right set of columns is the skills. The skills are presented as a three level hierarchy. The categories of sub-skills were described in Landry (2000). A new category 1.5 was added due to the increasing awareness that information security assurance is becoming of increasing importance for information systems (Dhillon, 2007; Whitman, et al 2007, 2010, 2012). The sub-skills in this area are based on statements within the NICE objectives for professionals (NICE, 2011).

Subsequently, we broke the skills into categories base on their historical placement in the various curricula. Table 3 shows the result of this analysis. Categories A through G were created, and tabulated for each category:

A. 21 skills relatively common to all models

B. 14 skills relatively common to later models

C. 53 skills relatively common to later models yet were dropped by IS 2010

D. 11 skills relatively common to all models except dropped by IS 2010

E. 24 skills relatively common to earlier models but dropped in all later models

F. 2 Skills added uniquely in IS 2010

G. 13 Skills based on NICE (2012) specifications but not in any curriculum model

In order to begin to understand the current situation, we grouped some of the above categories:

138	Total Number of Skills
24	Retired Skills (E)
13	New Skills not in any model (G)
101	**Active Skills**
21	Skills mostly in all programs (A)
14	Skills added in later models (B)
65	**Skills Current Through 2010**
53	Later skills, dropped by IS 2010 (C)
11	All models, dropped by IS 2010 (D)
2	Skills New from IS 2010 (F)
0	All Skills Accounted

It makes sense for the Skills in Category E s to become extinct. They were based on support for very different types of computers than those in use today. They relate specifically to problems that had to be solved to make old mainframe computers work.

Category G skills are sufficiently new that there has not been time for curriculum writers to include them in newer models. Since security issues are being recognized only currently, it seems that it is a matter of time before such specifications will emerge. However, for today skills categories E and G reflect skills that are not immediately relevant. This leaves 101 skills for consideration.

It might be argued that Categories A, B, and D (46 skills) have always been relevant in IS curricula, except that Category D (11 skills) were dropped by IS 2010.

In addition, Categories C (53 skills) and D (11 skills) are arguably still considered relevant; however they were not included in IS 2010. These 64 skills represent a very considerable contraction of focus in information systems curricula. Of these skills Applications Development represents almost ¼ of the skills that were not included.

An inspection of 240 business school information systems curricula (Apigian, 2010) found that a significant majority of IS programs offered courses with titles similar to:

Fundamentals of IS
Data and Information Management (database)
Systems Analysis and Design
IT Infrastructure (network communications)
Application Development (programming)

Interestingly, 99% percent of schools offered one or more courses in programming thus the lack of its inclusion in IS'10 is a significant change. This is one of the skills may be questionable.

The work of Apigian and Gambill (2010) represents a reasonable way of formulating an update to the current model curricula. University faculties live under many constraints, one of which might be addressing a new curriculum model. With the pressure to prepare their graduates to be relevant to the working world, attention must be paid to many factors. It is important to note that Apigian and Gambill gave numerous examples of the ability of faculty to take approaches to solve their own unique environment. Therefore, they state that no program is in full alignment with any model curriculum.

4. CONCLUSION

We have found that IS curricula have evolved significantly over the past fifty years. We have used the skill set in IS 2000 and augmented it back in time. We have also worked forward in time through 2012 incorporating IS 2010, NICE, and DAMA additions to the skills. Using this expanded skill set we were able to score curricula from 1972 through 2012 for their ability to generate the specific skills.

Then by grouping the skills according to use by the various models, we were able to show that of the 138 total skills, 37 skills were either retired of too new to be included in any of the models. Of the remaining 101 skills, some were common to all programs (35), remaining skills were common for programs up to IS 2010 when a significant number were abandoned (64). IS 2010 two skills in the enterprise computing realm. Possibly the most significant loss of skills in IS 2010 can be attributed to the deletion of a programming requirement. This appears to be inconsistent with the data of Apigian and Gambill that showed the overwhelming majority of IS programs housed in a school of business.

While the world advances, many of the principles and skills of our discipline are recurring; they have existed over the fifty year period. These skills partially define the discipline. There is no doubt that new technology can and will impact the curriculum. With a current and future focus on powerful smart phones and similar devices, our definition of ubiquitous computing will have to evolve another level. Likewise, security issues have risen to the level of extreme importance. Indeed, the curriculum Over the next 50 years curriculum may change more dramatically than the past 50 years Who will be the individuals making the important decisions in 2060?

5. 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.

Apigian, C. H. and Gambill, S. E. (2010). Are We Teaching the IS 2010 Model Curriculum? *Journal of Information Systems Education*, Volume 21, Number 4, Winter 2010, pp. 411-420.

Ashenhurst, R. L. (Ed.) (1972). A Report of the ACM Curriculum Committee on Computer Education for Management. *Association for Computing Machinery, Inc., 1972*.

Bloom, Benjamin S. (Ed.) (1956). The Taxonomy of Educational Objectives: Classification of Educational Goals. Handbook 1: The Cognitive Domain. : McKay Press, New York 1956.

Colvin, R. (2008). Information Systems Skills and Career Success, Masters Thesis, University of South Alabama, School of Computer and Information Sciences.

Couger, J. (Ed.) (1973). Curriculum Recommendations for Undergraduate Programs in Information Systems, *Communications of the ACM*, Volume 16, Number 12, December 1973, pp. 727-749.

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.

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.

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.

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).

Dhillon, G. (2007). Principles of Information Systems Security, Text and Cases, John Wiley and Sons, New Jersey.

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, 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.

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).

Haigood, B. (2001). Classification of Performance Level Requirements of Current Jobs Within the Field of Information Systems, Masters Thesis, University of South Alabama, School of Computer and Information Sciences.

Henderson, D., B. Champlin, D. Coleman, P. Cupoli, J. Hoffer, L. Howarth, K. Sivier, A., M. Smith, and E. Smith (2004). Model Curriculum Framework for Post Secondary Education Programs in Data Resource Management. The Data Management

Association International Foundation, Committee on the Advancement of Data Management in Post Secondary Institutions, Sub Committee on Curriculum Framework Development.

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

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 Conference on Information Systems*, Long Beach, CA.

Longenecker, H.E., and Feinstein, D.L. (1991b.) "On Establishing Excellence in Information Systems," *Journal of Information Systems Education*, Volume 3, Number 1, Spring 1991, pp. 26-31.

Longenecker, H.E., Feinstein, D. L. (Eds.) (1991c). *IS'90: The DPMA Model Curriculum for Information Systems for 4 Year Undergraduates.* Park Ridge, Illinois: Data Processing Management Association.

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. Jr, D. Henderson, E. Smith, P. Cupoli, D. M. Yarbrough, A. M. Smith, M. L. Gillenson, and D. L. Feinstein (2006). A Recommendation for a Professional Focus Area in Data Management for the IS2002 Information Systems Model Curriculum. In *The Proceedings of the Information Systems Education Conference 2006*, v 23 (Dallas): §2115. ISSN: 1542-7382.

NICE (2011). National Initiative for Cyber Security Education Strategic Plan: Building a Digital Nation, August 11, 2011, *DRAFT National Initiative for Cybersecurity Education (NICE) Strategic Plan,* Retrieved July 14, 2012: http://www.nist.gov/itl/comment_nice_8-11-11.cfm

NICE (2010). NICE Cybersecurity Workforce Framework-Summary Booklet.pdf, *National Initiative for Cybersecurity Education (NICE),* Retrieved July 14, 2012: csrc.nist.gov/nice/framework/documents/NICE-Cybersecurity-Workforce-Framework-Summary-Booklet.pdf

NICE (2012). NICE Cyber Security Workforce Details, *National Initiative for Cybersecurity Education (NICE),* Retrieved July 14, 2012: http://csrc.nist.gov/nice/framework/,

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.

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

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

Whitman, M.E. and Mattord, H.J. (2007). "Principles of Incident Response and Disaster Recovery", Cengage Technology, Boston.

Whitman, M.E. and Mattord, H.J. (2010). "Management of Information Security", Cengage Technology, Boston.

Whitman, M.E. and Mattord, H.J. (2012). "Principles of Information Security, Fourth Edition", Cengage Technology, Boston.

Appendix 1

Table 1: Output Characteristics of Graduates

(a) people ability to hear others, as well as listen to them; ability to describe individual and group behavior and to predict likely alternative future behavior in terms of commonly used variables of psychology and economics; ability to describe and predict task-oriented, time-constrained behavior in an organizational setting. (b) models ability to formulate and solve simple models of the operations research type; ability to recognize in context the appropriate models for situations commonly encountered. (c) systems ability to view, describe, define any situation as a system—specifying components, boundaries, and so forth; ability to apply this "systems viewpoint" in depth to some class of organizations--manufacturing firms, government bureaus, universities, hospitals, service providers, etc.; ability to perform an economic analysis of proposed resource commitments (includes ability to specify needs for additional information and to make a set of conditional evaluations if information is unavailable); ability to present in writing a summary of a project for management action (suitable to serve as a basis for decision); ability to present in writing a detailed description of part of a project, for use in completing or maintaining same. (d) computers knowledge of basic hardware/software components of computer systems, and their patterns of configuration; ability to program in a higher-level language; ability to program a defined problem involving data files and communications structures; ability to develop several logical structures for a specified problem; ability to develop several different implementations of a specified logical structure; ability to develop specifications for a major programming project, in terms of functions, modules and interfaces; knowledge of sources for updating knowledge of technology; ability to develop the major alternatives (assuming current technology) in specifying an information processing system, including data files and communications structures, to the level of major system components; ability to make an economic analysis for selecting among alternatives above, including identification of necessary information for making that analysis, and also to identify noneconomic factors; ability to make "rough-cut" feasibility evaluations (in terms of economic and behavioral variables) of proposed new techniques or applications of current technology, identifying critical variables and making estimates and extrapolations; ability to develop specifications for the computer-based part of a major information system, with details of task management and data base management components. (e) organizations knowledge of the function of purposeful organizational structure, and of the major alternatives for that structure; knowledge of the functional areas of an organization--operations, finance, marketing, product specification and development; ability to identify in an ongoing organizational situation the key issues and problems of each functional area; knowledge of typical roles and role behavior in each functional area; ability to identify possible short-term and long-term effects of a specified action on organizational

goals; ability to identify information needs appropriate to issues and roles above; knowledge of how information systems are superimposed on organizational patterns, on the operational, control, and planning levels; knowledge of techniques for gathering information; ability to gather information systematically within an organization, given specified information needs and/or specified information flows; ability to specify, given information needs and sources, several alternative sets of information transfers and processing to meet needs; ability to make "rough-cut" feasibility evaluations of such alternatives; ability to develop positive and negative impacts of a specified information system on specified parts of an organization; ability to develop specifications for a major information system, addressing a given organizational need, and determine the breakdown into manual and computer-based parts. (f) society ability to articulate and defend a personal position on some important issue of the impact of information technology and systems on society (important, as defined by Congressional interest, public press, semi-technical press, etc.); ability to develop several positive and several negative impacts of a specified information system in a specified part of society; ability, given such specifications of impacts, to perform a "'rough-cut" feasibility analysis of them in terms of behavioral and economic variables.
Output Characteristics of Graduates: The text in this table is taken from Ashenhurst (1972) in its entirety. It is typical of similar tables expressed in later curriculum models.

Appendix 2

Table 2. Skill Depths Achieved for Indicated Model Curricula

Model Curriculum						Each skill was mapped to course outcomes in the applicable model curriculum. Only courses which were part of the requirements for the degree were considered—electives not required of all students were not included. Skill depths were originally defined in IS'90 and refined in IS'97—they were used in IS'97, IS2002 and IS2010 without change.		
73	**81**	**86**	**90**	**02**	**10**			
Skill Depths **1 – Recognize** **2 – Differentiate** **3 – Use** **4 – Apply**						**Skill**	**Skill Name**	**Skill Words**
						1.0 Information Technology Skills **1.10 Software Development**		
2	3	1	3	2	2	1.1.0	Low level data structures	bits, bytes, number representation, money representation, character representation, rounding operations, overflow
3	3	3	4	3		1.1.1	Programming-principles, objects, algorithms, modules, testing	principles, concepts, control structures (sequence, selection, iteration); modularity, objects and ADTs, data structures, algorithmic design, verification and validation, cohesion, coupling, language selection, user interface design, desk checking, debugging, testing, error correction, documentation, installation, integration, operation; writing code in a modern programming language (e.g., Java, C#); interpreted and compiled computer languages; design tools; secure coding principles and practices
	3		3	2		1.1.2	Application Development-requirements, specs, developing, HCI considerations	principles, concepts, standards; requirements, specifications, HCI planning, device optimization (e.g. touch screen, voice), development and testing, utilization of IDEs, SDKs, and tool kits; configuration management, installation, module integration; conversion, operation
2	2	2	3	3	2	1.1.3	Algorithmic Design, Data, Object and File Structures	analysis, design, development, debugging, testing, simple data structures(arrays, records, strings, linked structures, stacks, queues, hash functions). Functions, parameters, control structures, event driven concepts, OO design, encapsulation, classes, inheritance, polymorphism, sorting, searching
	3	2	2	3		1.1.4	Problem Solving-identify problems, systems concepts, creativity	devise questions to help identify problems, apply systems concepts to definition and solutions of problems, formulate creative solutions to simple and complex problems, Fishbone-root cause, SWOT, Simon Model, Triz, ASIT; embracing developing technology; methodologies (waterfall, object, spiral etc.), dataflow, structured
			2	2		1.1.5	Client Server Software Development	thin/full client; software specs, development, testing, installation, configuration, trouble-shooting, enhancement, maintenance, training and support; report/interface, development, documentation standards, application configuration management--e.g. Source-safe; Drop box, project documentation

			3	1		1.1.6	HCI Principles and Paradigms	human-computer interfaces, user interfaces, man-machine interfaces, "8 golden rules"; keyboards; touch technology, voice, video, real-time signals, GPS
						1.1.7	Digital Media	standards for sound, video (still and full motion) including wav, jpeg, tiff, raws
						1.1.8	Software Security	vulnerability, dependability, trustworthiness, survivability, resilience, threat and vulnerability analysis; software assurance; translation of security requirements in application design; secure code documentation; developing countermeasures to identified risks; assessment of vulnerabilities and risks
		2		2		1.1.9	Prototype	storyboard, build, simulate, test, re-develop
			1	1		1.1.10	Code Generators	Compilers, interpreters, specialized code segment generators
3	1					1.1.11	Storage Management	real and virtual storage, allocation and deallocation, distributed systems; stacks; garbage collection
1	1					1.1.12	Multiprogramming and Multiprocessing	jobs, job linkage, modes (batch, interactive processing), performance monitoring
3	3	2				1.1.13	File Systems	physical allocation, devices, capacity management, access modes (sequential, indexed sequential, random)
1	3		1			1.1.14	Machine Structures	words, addressing, sequential allocation, linked allocation, pointers and indirect addressing, pointer manipulation; Machine and assembler languages
	3					1.1.15	Computer Operations	input, output, jobs, job control, performance control
2	2					1.1.16	Systems of Programs	programming a system of related program components, intertask communication and linkage, run-time data storage; code sharing, reentrancy, relocatibility, dynamic linking and loading; multi-tasking
2	2	2	3	3		1.1.17	Testing	segment testing, module testing, program testing, system testing; test data, and testing strategies
3	3	2	3	3		1.1.18	Procedural Languages	FORTRAN, COBOL, PL1, BASIC, C; advanced functions (sorting, searching, mathematical and statistical routines); functions; subroutine libraries
				1		1.1.19	Object Oriented Languages	C++, C#, VB
						1.1.20	Logic Programming	Lisp, Prolog
2	2	2				1.1.21	Input Devices	cards, tape, terminals, work stations, thin clients, microphone, video, data capture, data entry mechanisms
2	2	2				1.1.22	Output Devices	cards, tape, terminals, work stations, printers, audio, video, controlled devices
	2			2		1.1.23	Information Systems	users, business process, programs, hardware, communication systems, applications, projects, services
						1.20 Web Development		
				3		1.2.1	Web page Development-HTML, page editors, tools	FrontPage, HTML, page building/edit tools, frames; http, Dreamweaver, Photoshop

				2		1.2.2	Web programming-thin client, asp, aspx, ODBC, CGI,E-commerce, web services, scripting	thin client programming: page design; HTML, *.asp/aspx coding; session variables / page security; ODBC; CGI programming; integration of multi-media; e-commerce models; tools: Perl, Visual Studio, Java, Web services, XML server / client side coding, web services, hypertext, n-tier architectures; integration of mobile technology
				2		1.2.3	Web Systems Development Tools	e.g, sharepoint, Joomla, Drupal, IDEs, SDKs, snagit, Jing
						1.2.4	Web Security and Vulnerability	vulnerability, penetration testing, vulnerability scanning; browser security; external memory issues
						1.30 Database		
2	2	1	3	4	4	1.3.1	Modeling and design, construction, schema tools, DB systems	Data modeling, SQL, construction, tools -top down, bottom up designs; schema development tools; desk-top/enterprise conversions; systems: Access, SQL Server/Oracle/Sybase, data warehousing & mining; scripts, GUI tools; retrieve, manipulate and store data; tables, relationships and views
	1	1	2	3	2	1.3.2	Triggers, Stored Procedures, Audit Controls: Design / Development	triggers, audit controls-stored procedures, trigger concepts, design, development, testing; audit control concepts/standards, audit control Implementation; SWL, concepts, procedures embedded programming (e.g. C#)
	1	2	1	1	2	1.3.3	Administration: security, safety, backup, repairs, Replicating	monitoring, safety -security, administration, replication, monitoring, repair, upgrades, backups, mirroring, security, privacy, legal standards, HIPAA; data administration, policies
						1.3.4	Metadata: architectures, systems, and administration	definition, principles, practices, role of metadata in database design, repository, dictionaries, creation, ETL, administration, usage, tools
					2	1.3.5	Data Warehouse: design, conversions, reporting	star schema, ETL, data cleansing and storage, reporting tools, business intelligence, analytic queries, SQL OLAP extensions, data mining
	1			2	2	1.3.6	Data Quality: dimensions, assessment, improvement	Data Accuracy, Believability, Relevancy, Resolution, Completeness, Consistency, Timeliness; Data definition quality characteristics, Data model / requirements quality characteristics; Data clean-up of legacy data, Mapping, transforming, cleansing legacy data; Data defect prevention, Data quality employee motivation, Information quality maturity assessment, gap analysis
						1.3.7	Database Security	SQL injection attacks and counter measures; encryption; limiting exposure in internet applications; risk management: attacks and countermeasures
		1	1	2		1.3.8	Data sources and advanced types	Accessing external data sources; use of search engines; purchasing data; image data; knowledge representations
	2	2				1.3.9	Data Models	Hierarchical, Network, Relational; DDL, DML considerations; GUI, script representations
						1.40 Systems Integration		

2	3	2	2	3		1.4.1	Computer Systems Hardware	fundamentals: cpu computer system block diagram, firmware, digital logic, serial vs parallel, bus, interface components; memory addressing, coding, data representation; assembler, multi-processors, DMA,, disk, tape, interrupts; embedded systems; fault tolerance; microprocessors
	1		1	4	2	1.4.2	Networking (Lan/Wan) and Telecommunications	fundamentals: encoding, data transmission, noise, media, devices, layered models, TCP/IP, telephony, network architecture; communication protocols such as TCP/IP, Host configuration, Domain Name Server
		1	1	2		1.4.3	Operating Systems Management-multi platforms/protocols, Win/Unix/Linux/VM	multi platforms, multi protocols; systems Win XP, Win 2003 Unix; Linux, installation, configuration; security; connectivity, performance monitoring, virtual machine emulations; Open Systems models; distributed computing
2	3	2	1	2		1.4.4	Computer Systems Software-OS fundamentals, resource mgt concepts	OS fundamentals: memory, disk, tape and resource management, remote scheduling, memory management, device management, security, file systems, real-time and embedded systems, fault tolerance, scripts; interoperability
				3	2	1.4.5	LAN/WAN Design and Management	Ethernets, hubs, routers, TCP/IP, internet, intranet; enterprise networking, Lans/Wans, network administration, design, configuration, installation , optimization, monitoring, testing, troubleshooting, router configuration, router, protocols, switches, firewalls and security, wireless considerations; network security architecture-defense in depth principles; network access and authorization (e.g. public key infrastructure); security objectives, operational objectives and tradeoffs; security controls; identification and vulnerabilities
				2		1.4.6	Systems Configuration, Operation, Administration	architecture, configuration, conversion, management, economics, installation, integration, administration, monitoring, maintenance, upgrades, documentation service pack scheduling, client services, users and user groups, replication backups, disaster planning and recovery, site management, COOP, power management, multi-site fail-over mechanisms, user education; security audit procedures; virtualization; fault tolerance
2						1.4.7	Inter-systems Communications	Customer Information Control System/Inter-systems Communications(CICS/ISC)
						1.4.8	Data mapping and exchange	types include bidirectional, unidirectional, translation key (ex. SNOMED to ICD-9)
	2		2	2		1.4.9	End-user interface	Terminals, GUI, multimedia, browsers; resource requirements, operating system drivers, communication protocol
2	1					1.4.10	Communication system hardware	channels, channel capacity, noise, error detection/correction; regulatory agencies, tariffs; transmission codes, transmission modes
	1					1.4.11	Communication system organization	single line, point to point, multi-drop; networks: centralized and decentralized; control protocol; switched, store forward; concentrators
						1.50 Information Assurance and Security		
			1	2	2	1.5.1	Information Assurance Model	Security services, information states, security countermeasures, Security implementation: gates, guards, guns; cryptography, Disaster recovery, Business continuity planning, forensics; IA architecture

						1.5.2	Security Mechanisms	cryptography: cryptosystems, keys-symmetric/asymmetric, performance authentication (who you are, what you have/know), passwords bio-authentication; Redundancy, intrusion detection
				3	2	1.5.3	Security Operations	ethical/legal issues, auditing, costs/benefits, standards-DES, ISO 177799 risk identification/mitigation, physical security implementation business impact analysis, CASPR, technology innovation and risk incident management, enforcement
		2			2	1.5.4	Security Policy	IAS policy/procedure creation, vulnerability, countermeasures security technology and system access (multi-levels classification e.g. unclassified, top secrete ...), property seizure information management/system administration and security,security services: availability, integrity, authentication user education; develop/update security policies and implement system designs which meet objectives (confidentiality, integrity, availability...)
						1.5.5	Security Attacks	denial of service, protocol attacks, active/passive attacks, buffer overflow attacks, viruses, Trojan horses, worms, adware, penetration testing, digital forensics, legal evidence, media analysis, threat analysis: risk assessment and cost benefit to business processes vulnerability--perpetrators: inside / external, hacker / cracker; hardware / software. Firewalls, demilitarized zones, and encryption; Use of tools to detect network intrusions and vulnerability
2	1	2	2	3	2	1.5.6	Privacy Impact	Requirements for confidentiality, integrity and availability; privacy impact analysis of application security design protecting personal identifiable information.
						1.5.7	Information Assurance Systems	documentation of information assurance components which allocate security functions
				3		1.5.8	Information Systems Email Management	application of confidentiality, integrity and availability principles; threat and vulnerability analysis; detection of security gaps in application and system architectures
				2		1.5.9	Information Technology Security Principles	IA certification and accreditation; process activities and related documentation, system life-cycle support plans, concepts fo operations, procedures and trainng materials; security risks and countermeasures; security controls needs; security management; concepts of policy-based and risk adaptive access controls
						2.0 Organizational and Professional Skills **2.10 Business Fundamentals**		
3	1	2	2	2	2	2.1.1	Learning Business Process and Environment	learning business process and environment, exchanges, competitive position, e-business, global concepts, business models, Creating value, Value chain, improving value creation; financial markets, determining value of securities; organizational models
			2	2		2.1.2	Accounting, Distribution, supply chain management, Finance, HR, Marketing, Production, payroll, inventory processing	accounting (language of money, representations of accounts, reports), distribution (purchasing, supply chain management, distribution systems), finance, human resources (laws, compensation, recruiting, retention, training), marketing (the market, customers and customer satisfaction, market strategies, cycle time and product life cycle; environment scanning), production, international business

		2		3	3	2.1.3	Business Problems and Appropriate Technical solutions, end-user solutions	business problems and appropriate technical solutions; quantitative analysis and statistical solutions; decision formulation and decision making; business intelligence systems; business use of spreadsheets, desk-top databases, presentation software, word processing and publishing
	1		2	2	1	2.1.4	Business Law	legal system, courts, dispute resolution processes (mediation, arbitration, conciliation, negotiation, trial); types of organizations, contracts, warranties, and product liability; policy and management of intellectual property
			1		3	2.1.5	Disaster Recovery	identify essential system functions to support business functions for restoration and recovery after a catastrophic failure; define requirements for critical system performance and continuity of business function; backup, replication, fail-over processes in support of system performance subsequent to a disaster
				2	3	2.1.6	Enterprise Information Systems and Business Intelligence	Alignment of business processes with large system structures; configuration of large systems; implementation and training; integration with business intelligence capabilities and optimization of business procedure.
	1			2		2.1.7	Modes of Business	B to B, B to C, C to C, B to G, C to G; organizational span (individual, work group, department, enterprise, inter-organization)
			1	1		2.1.8	Regulations	Federal and State Regulations; compliance, audits, standards of operation (e.g. FAR); agencies and regulatory bodies
				1	1	2.1.9	IT Standards	ITIL, CORBA
3	2	2	2	3		2.1.10	IT Support for Business Functions	Business systems (budget, personnel, capital, equipment, planning, training, control); Specific systems (production, financial, accounting, marketing, supply chain, securities, taxation, regulation compliance)
3	1					2.1.11	Operational Analysis	scheduling, allocation, queuing, constraint theory, inventory management models, financial models, forecasting, real time analysis; linear programming, simulation
2	1					2.1.12	Managing the IS Function	Development, deployment, and project control; managing emerging technology; data administration; CIO functions; security management
		2		1		2.1.13	Information Center Service	PC Software training and support; application and report generators, IS Development, Development and operations staff; corporate application management, data safety and protection, disaster recovery
						2.20 Individual and Team Interpersonal Skill		
			1	2		2.2.1	Learning to learn	attitude of personal responsibility, journals, learning maps, habits of reading, listening to tape/cd, attending professional seminars, teaching others, meta-thinking, life long learning; human learning: recognition, differentiation, use, application, analysis, synthesis and evaluation
	1	2	2	3		2.2.2	Professionalism-self-directed, leadership, time management, certification, conferences	being self-directed and proactive, personal goal setting, leadership, time management, being sensitive to organizational culture and policies; personal development (conferences, read literature, use self-development programs)
	1		2	2		2.2.3	Personal Skills-encouraging, listening, being Organized, principles of motivation	encouraging, listening, negotiating, being persuasive, being organized Personality types and relationships (DISC, MBTI, COLOR)

	1	1		1		2.2.4	Professionalism-committing to and completing work	Persistence, committing to and rigorously completing assignments, can-do
2	2	1	2	2	2	2.2.5	Teams-team building, vision / mission development, synergy building and problem solving; leadership	team building, vision and mission development, planning, synergistic consensus team leadership, leadership development, negotiation, conflict resolution
2	1		3	2		2.2.6	Communication-oral, written, multimedia, empathetic listening	oral, written, and multimedia techniques; communicating in a variety of settings; empathetic listening, principle centered leadership, alignment technical memos, system documentation, technical requirements; necessity for involvement; development of resistance
			3	3	2	2.2.7	Ethics-theory/concepts, setting an ethical example	ethical theory and concepts, codes of ethics--AITP/ACM; setting an ethical example; ethical policies, intellectual property, hacking, identity theft
	1			2		2.2.8	Critical Thinking	fact recognition, argument strength, analysis (break into components), synthesis(assembling the components); abstraction; qualitative research principles
2	2	2	2	2		2.2.9	Mathematical Fundamentals	Mathematics (algebra, trigonometry, variables, operations, expressions, logic, probability, limits, statistics)
	1	1	2	2	2	2.2.10	Collaboration support by IT	It Solutions for Individuals and Groups, Problem solving mechanisms in support of meetings, consensus development
	1		3	2	2	2.2.11	Impact of IT on Society	IT impact on individuals, on groups, on enterprises, on societies; knowledge work and support by IT; computer industry and society, work force requirements
		1		2		2.2.12	IT Career Paths	Programmers, Application Developers, Information Analyst, Systems Analysis, Data Management, CIO, CTO
			1	2		2.2.13	HCI Principles: underpinnings	Cognitive Process, education learning levels, interface design, concepts of usefulness, the 8 golden rules
				2		2.2.14	Individual behavior	learning styles (visual, auditory, kinesthetic), motor skills, linguistic mechanisms, auditory mechanisms
	1			1		2.2.15	Cognition	concepts of learning; sequential levels of learning (recognition, differentiation, use / translation, apply); relationship of learning and emotion
	1	1	1	1		2.2.16	Develop Consultant Characteristics	build relationship, identify need, present alternatives, provide assistance as needed, make recommendations, be supportive
						2.30 Social Implications of Technology		
1		2				2.3.1	Historical perspectives	economic and social issues of technology development; benefits and threats of technology; availability of information; technology and the quality of life
1		2				2.3.2	The computer industry	competition, technology advances, pricing, government regulation, increasing cpu power, increasing storage capacity, higher speed communication, higher speed processors
1		1				2.3.3	Job displacement	technologic advances generate efficiencies in work flow generate a need for fewer people for current tasks; need for continuous re-training

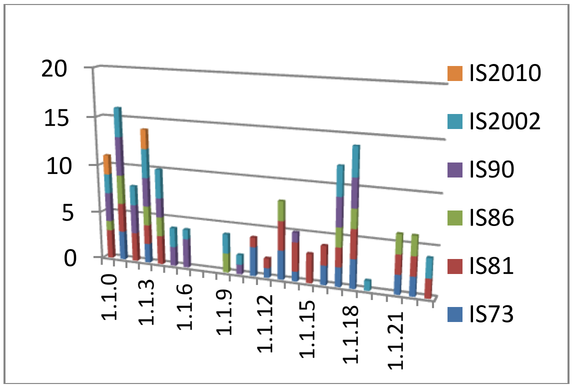
2				2		2.3.4	IT effects on Individuals	Automated record keeping (academic transcripts, checking accounts, mortgage accounts, legal system, government services, welfare management); weather forecasting, national repositories of information; healthcare systems and management
		3		2		2.3.5	PC Impact	changing responsibilities, decentralization, increase in personal productivity, direct executive support and power
						2.40 Personal Computer		
		2				2.4.1	History	Initial introduction by IBM, Apple; PC as '83 Man of the Year, Growth of Intel, growth of entire related industry
		2				2.4.2	Work station	LAN connection, PC-operating system, end-user software, browsers
		3		3		2.4.3	Personal productivity	PC tools, PC applications, personal application development
				3		2.4.4	Internet access	Browsers, http, e-commerce, email, access to the world
		1		3		2.4.5	E-mail	instantaneous access to people, access to search engines, access to storage
		1		3		2.4.6	Business software	personal applications, enabling of work-from-home, remote work group support, project management, collaborative software, group information systems, access top accounting systems and banks, entertainment
		3		3		2.4.7	Spread sheets	executive tool, financial accounting, graphics, what-if analysis
		1		3		2.4.8	Database	Relational DBMS, application support, spread sheet integration
						3.0 Strategic Organizational Systems Development **3.10 Organizational systems Development**		
1	3	2	2	4	4	3.1.1	Strategic Utilization of Information Technology	use of IT to support business process, integration of customer requirements; team development of systems, reengineering concepts and application, methodologies, interfaces, systems engineering, CRM and ERP concepts; Agile, Object, Lean UX and other methodologies; identification of security issues, incorporation of security concepts into designs ensuring security principles; development of IS policy
	3		2	2	4	3.1.2	IT Planning	value of IT, integration of IT in reengineering, IT policy, end user advocacy and optimization, IT advocacy and alignment outsourcing / off-shoring (risks, benefits, opportunities), training; capture security controls and requirements, ensure integration of security objectives, assurance of people and information protection; ensure security in interface considerations
	2		2	2	3	3.1.3	IT and Organizational Systems	types of systems relationship of business process and IT, user developed systems, use of packaged software, decision systems, social systems; information assurance and security designs; IT support of end-user computing, group process and computing, and enterprise solutions

3	3	3	3	4	3	3.1.4	Information Systems Analysis and Design	investigate, information analysis, group techniques / meetings design, systems engineering, Information architectures, enterprise IS development with strategic process; consideration of alternatives; application and security planning; conversion and testing, HIPAA, FERPA, ISACA, GAAP; requirements analysis. cost analysis, cost/benefit, satisfaction of user need / involvement, development time, adequacy of information assurance controls; consideration / adoption of emerging technology (e.g. mobile computing), consideration of optimal life-cycle methodologies and tools; physical design (database, interface design, reports design, programming, testing, system testing)
	2	2	2	2	2	3.1.5	Decision Making	personal decision making, Simon's model, structured, unstructured decisions, decision tools, expert systems, advanced problem solving (Triz, Asit); business intelligence, advanced reporting technologies.
2	1	1	2	2	3	3.1.6	Systems Concepts, Use of IT, Customer Service	develop client relationships, understand and meet need, involving the client at all phases of the life-cycle; review of customer functional requirements; consideration of improved business process; assurance of customer needs into requirements analysis
2	2		1	2		3.1.7	Systems Theory and Quality Concepts	system components, relationships, flows, concepts and application of events and measurement, customer expectations, quality concepts; boundaries, open systems, closed systems, controlled systems; effectiveness, measuring system performance, efficiency
			1	1		3.1.8	CMMI and Quality Models	quality culture, goals; developing written standards, templates; process metrics development process improvement through assessment, lessons learned
			2	2	2	3.1.9	Systems Engineering Techniques	scope development, requirements determination, system design, detailed design and specifications, Enterprise Architecture, System architecture, information architecture, make or buy, RFP/Bid Process verification and requirements tracing, validation planning and test case development, unit testing, integration, system testing, system certification, system acceptance, installation and operation of the system, post-implementation audit; ensuring security designs, secure configuration management; agency evaluation and validation of requirements; ensuring customer training and incorporation of installation teams
		2	3	4		3.1.10	End-User Systems	individual software: word processing, spreadsheets, database, presentation, outlining, email clients, statistical packages; work-group software; enterprise software: functional support systems (e.g. PI), enterprise configuration
	1	1	1	1	3	3.1.11	Enterprise Information Systems in Support of Business Functions	Systems that support multiple enterprise functions (e.g. SAP); Electronic Medical Record Systems for physician-groups, and for hospitals; Cloud solutions for individual and organizational support; TPS, DPS, MIS, EIS, Expert System
				2		3.1.12	Emerging Technology	Bleeding edge technologies; testing and adoption of new technologies; cost benefit of new technologies
2	1	2	2	2		3.1.13	Systems Roles in Organizations	operations, tactical, strategic
	1			2		3.1.14	Organizational Models	Hierarchical, Flow Models, Matrix
1		1		2		3.1.15	Metrics and Improvement	Development metrics, quality metrics, metrics in support of 6-Sigma or CMMI, customer satisfaction; Learning Cycles (Understand the problem, plan, act, measure/reflect and learn and repeat the cycle), Lessons Learned (what was supposed to happen, what happened, what was learned, what should be done, communicate the observations)

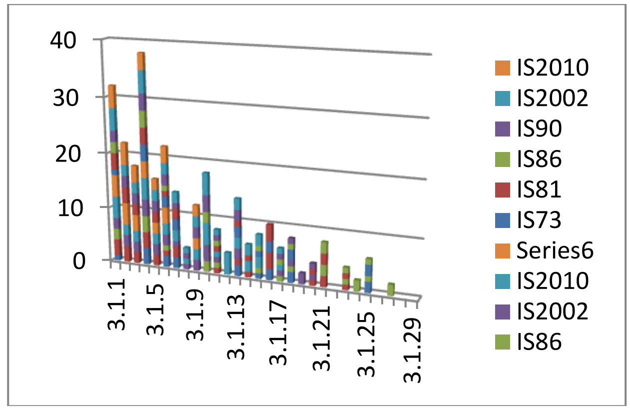
2	3					3.1.16	Hardware selection, acquisition, and installation for project	Determination of capacity for process, storage devices, and communication systems; consideration of alternative hardware; bid preparation, bid evaluation, and final system selection; hardware installation and testing; system deployment and initial operation.
		1	1	1		3.1.17	Facilities Management	Physical facility construction, access control, fire protection, prevention of flooding; power management (public utilities, generators--fuel storage, testing, battery management--lightening protection), air conditioning, fire prevention systems, physical security, protection from weather
2		1	1			3.1.18	Maintenance Programming	Fault detection and isolation, code correction, code testing, module testing, program testing; code, module, system documentation
			1			3.1.19	Decision Structure	structured, unstructured decisions, decisions under uncertainty, heuristics, expert systems
	1		1			3.1.20	Decision Tools	application results, idea generation, Delphi, nominal group, risk analysis, cost benefit analysis
	2	2				3.1.21	Structured development	process flows, data flows, data stores, process logic, database design, program specifications and design
						3.1.22	Object Oriented Development	UML; class diagrams, swim lane, use case, sequence diagram, design patterns
	1	1				3.1.23	Screen Design	menus, input forms, output forms and reports, linkage of screen modules, navigation
		1				3.1.24	Frameworks and Libraries	object libraries, source libraries, language extensions
2		1				3.1.25	Reports Development	simple lists, control break--group by--reports, error reports, exception reports, graphics reports, audit reports
						3.1.26	Develop Audit Control Reports	Document new accounts with public information: names, addresses, organizations, items, events
		1				3.1.27	Develop cash audits	deposits, batches, accounting variable controls, accounting distributions
						3.1.28	Audit analysis of separation of function	establish roles of staff, validate transactions, validate personal functioning
						3.1.29	Audit risk and disaster recovery strategies	determine risks, verify adequacy of mitigations; audit failure processes, replication, and failover mechanisms; audit backup strategy and physical results
						3.20 Project Management		
2	1	2	2	3	2	3.2.1	Project Goal Setting and Planning	establish project goals consistent with organizational goals as well as re-engineering initiatives; project plan and scope statements: cost, schedule and performance; project initiation, project charter
1	1	2	1	1	2	3.2.2	Monitor and Direct Resources and Activities, Team Leading	specify, gather, deploy, monitor and direct resources and activities, team charter, RACI charts, project team building, team assessment
2	2	2	2	2	2	3.2.3	Coordinate Life Cycle Scheduling and Planning	life cycle coordination, consultant management, schedule management, use of project planning; reporting; documentation

1				1	2	3.2.4	Apply concepts of continuous improvement	apply concepts of continuous quality improvement, providing reliable, cost-effective solutions that satisfy formal standards for performance, capacity, reliability, security, and safety; concept of standard practice--IEEE; ISO 9000;CMMI, 6 Sigma, Federal, state and local quality initiatives
2	2	1	2	2	3	3.2.5	Project Scheduling and Tracking	planning, scheduling and milestones; selection of process model; organizational issues; work breakdown structures; staffing; cost estimation, cost/benefit resources allocation, reviews, measurement, feedback, communication, ensuring quality, use project management software (PERT and Gantt Charts)
	1	1	1	1	2	3.2.6	Project Communications	Formal, informal, nonverbal communications; media selection: hard copy, phone, email, meetings, web conference. Individual, group communication. Communication planning and management: notifications-rules / responsibilities, When to communicate what to whom; issue log and management; communication of project goals, and progress with management
				1	2	3.2.7	Risk Management and Mitigation	Risk determination-root cause analysis, risk management: risk probability, Risk impact; probability / impact matrix and risk mitigation strategies— Avoidance, control, assumption, transfer. Risk register
2	1	1	1	2	2	3.2.8	Project Change Control	Configuration Management, Security configuration management; Project change control Board, Requirements Change process, approvals, impact on scope, schedule and cost, work project completion and acceptance, scope creep.
1	1	1	1	2		3.2.9	Change Process	introduction of change, planning for change, acceptance, resistance and its prevention; negotiation and conflict resolution strategies, use of clear standards
				1		3.2.10	Project Close-down	managing the close-down activities; development of close-down reports

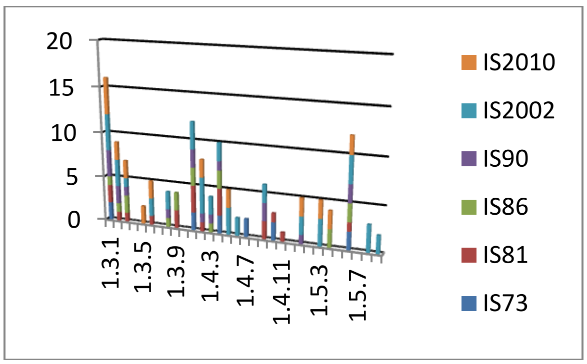
Appendix 3. Graphical representation of Skill Depths Achieved for Indicated Model Curricula



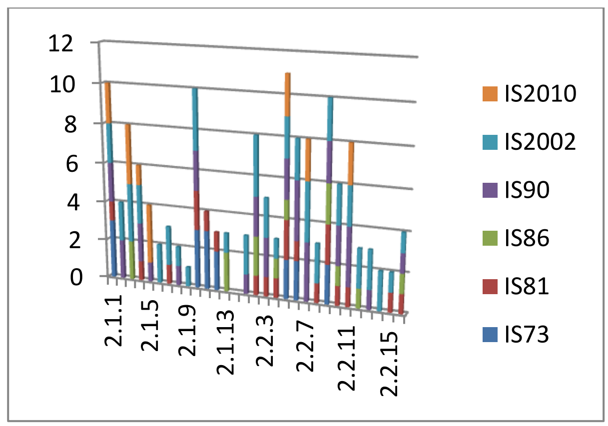
Software Development (1.1.0 – 1.1.24)



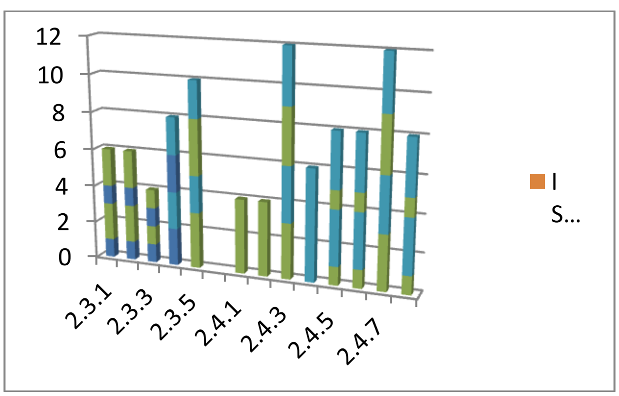
Organizational Systems Development (3.1.1 -3.1.29)



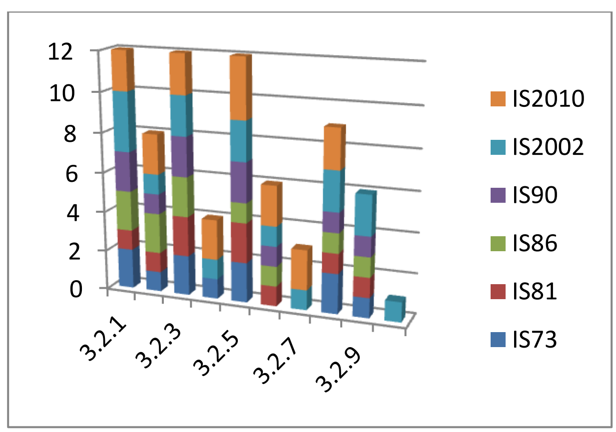
Database (1.3.1 – 1.3.9) Systems Integration (1.4.1 -1.4.11) Information Assurance &Security (1.5.1 -1.5.9)



Business Fundamentals (2.1.1 – 2.1.13)
Individual Skills (2.2.1 – 2.2.16)



Social Implications of Technology (2.3.1 – 2.3.5) Personal Computer (2.4.1 – 2.4.8)



Project Management (3.2.1 – 3.2.10)

Table 3. Skills Appropriate To Various Curriculum Models

A. 21 Skills relatively common to all models

73	81	86	90	02	10	Skills (current as of 2012)	
2	3	1	3	2	2	1.1.0	Low level data structures
2	2	2	3	3	2	1.1.3	Algorithmic Design, Data, Object and File Structures
2	2	1	3	4	4	1.3.1	Modeling and design, construction, schema tools, DB systems
	1		1	4	2	1.4.2	Networking (Lan/Wan) and Telecommunications
2	1	2	2	3	2	1.5.6	Privacy Impact
3	1	2	2	2	2	2.1.1	Learning Business Process and Environment
				2	2	2.1.6	Enterprise Information Systems and Business Intelligence
2	2	1	2	2	2	2.2.5	Teams-team building, vision / mission development, Synergy building and problem solving; leadership
1	3	2	2	4	4	3.1.1	Strategic Utilization of Information Technology
	3		2	2	4	3.1.2	IT Planning
	2		2	2	3	3.1.3	IT and Organizational Systems
3	3	3	3	4	3	3.1.4	Information Systems Analysis and Design
	2	2	2	2	2	3.1.5	Decision Making
2	1	1	2	2	3	3.1.6	Systems Concepts, Use of IT, Customer Service
	1	1	1	1	3	3.1.11	Enterprise Information Systems Supporting Business Functions
2	1	2	2	3	2	3.2.1	Project Goal Setting and Planning
1	1	2	1	1	2	3.2.2	Monitor and Direct Resources and Activities, Team Leading
2	2	2	2	2	2	3.2.3	Coordinate Life Cycle Scheduling and Planning
2	2	1	2	2	3	3.2.5	Project Scheduling and Tracking
	1	1	1	1	2	3.2.6	Project Communications
2	1	1	1	2	2	3.2.8	Project Change Control

B. 14 Skills relatively common to later curriculum

73	81	86	90	02	10	Skills (current as of 2012)	
	1			2	2	1.3.6	Data Quality: dimensions, assessment, improvement
				3	2	1.4.5	LAN/WAN Design and Management
			1	2	2	1.5.1	Information Assurance Model
				3	2	1.5.3	Security Operations
		2			2	1.5.4	Security Policy
		2		3	3	2.1.3	Business Problems and Appropriate Technical solutions, end-user solutions
	1		2	2	1	2.1.4	Business Law
			1		3	2.1.5	Disaster Recovery
				1	1	2.1.9	IT Standards
	1	1	2	2	2	2.2.10	Collaboration support by IT
	1		3	2	2	2.2.11	Impact of IT on Society
			3	3	2	2.2.7	Ethics-theory/concepts, setting an ethical example
			2	2	2	3.1.9	Systems Engineering Techniques
				1	2	3.2.7	Risk Management and Mitigation

C. 53 Skills relatively common to later curricula (dropped by IS2010)

73	81	86	90	02	10	Skills (current as of 2012)	
			1	1		1.1.10	Code Generators
2	2	2	3	3		1.1.17	Testing
3	3	2	3	3		1.1.18	Procedural Languages

				1	1.1.19	Object Oriented Languages
	3		3	2	1.1.2	Application Development-requirements, specs, developing, HCI considerations
	2			2	1.1.23	Information Systems
			2	2	1.1.5	Client Server Software Development
			3	1	1.1.6	HCI Principles and Paradigms
		2		2	1.1.9	Prototype
				3	1.2.1	Web page Development--HTML, page editors, tools
				2	1.2.2	Web programming-thin client, ASP, ASPX, ODBC, CGI, E-commerce, web services, scripting
				2	1.2.3	Web Systems Development Tools
	1	1	2	3	1.3.2	Triggers, Stored Procedures, Audit Controls: Design / Development
	1	2	1	1	1.3.3	Administration: security, safety, backup, repairs, replicating
		1	1	2	1.3.8	Data sources and advanced types
2	3	2	2	3	1.4.1	Computer Systems Hardware
		1	1	2	1.4.3	Operating Systems Management-multi platforms/protocols, Win/Unix/Linux/VM
2	3	2	1	2	1.4.4	Computer Systems Software-OS fundamentals, resource mgt concepts
				2	1.4.6	Systems Configuration, Operation, Administration
	2		2	2	1.4.9	End-user interface
				3	1.5.8	Information Systems Email Management
				2	1.5.9	Information Technology Security Principles
		2		1	2.1.13	Information Center Service
			2	2	2.1.2	Accounting, Distribution, supply chain management, Finance, HR, Marketing, Production, payroll, inventory processing
	1			2	2.1.7	Modes of Business
			1	1	2.1.8	Regulations
			1	2	2.2.1	Learning to learn
		1		2	2.2.12	IT Career Paths
			1	2	2.2.13	HCI Principles: underpinnings
				2	2.2.14	Individual behavior
	1			1	2.2.15	Cognition
	1	1	1	1	2.2.16	Develop Consultant Characteristics
	1	2	2	3	2.2.2	Professionalism-self directed, leadership, time management, certification, conferences
	1		2	2	2.2.3	Personal Skills-encouraging, listening, being Organized, principles of motivation
	1	1		1	2.2.4	Professionalism-committing to and completing work
2	1		3	2	2.2.6	Communication-oral, written, multimedia, empathetic listening
	1			2	2.2.8	Critical Thinking
2				2	2.3.4	IT effects on Individuals
		3		2	2.3.5	PC Impact
		2			2.4.1	History
		2			2.4.2	Work station
		3		3	2.4.3	Personal productivity
				3	2.4.4	Internet access
		1		3	2.4.5	E-mail
		1		3	2.4.6	Business software
		3		3	2.4.7	Spread sheets
		1		3	2.4.8	Database
		2	3	4	3.1.10	End-User Systems
				2	3.1.12	Emerging Technology
		1	1	1	3.1.17	Facilities Management
			1	1	3.1.8	CMMI and Quality Models
				1	3.2.10	Project Close-down
1				1	3.2.4	Apply concepts of continuous improvement

D. 11 Skills relatively common to earlier & later curricula (dropped by IS2010)

73	81	86	90	02	10	Skills (current as of 2012)	
3	3	3	4	3		1.1.1	Programming-principles, objects, algorithms, modules, testing
	3	2	2	3		1.1.4	Problem Solving-identify problems, systems concepts, creativity
	2	2				1.3.9	Data Models
3	2	2	2	3		2.1.10	IT Support for Business Functions
2	2	2	2	2		2.2.9	Mathematical Fundamentals
2	2		1	2		3.1.07	Systems Theory and Quality Concepts
2	1	2	2	2		3.1.13	Systems Roles in Organizations
	1			2		3.1.14	Organizational Models
1		1		2		3.1.15	Metrics and Improvement
			1			3.1.19	Decision Structure
1	1	1	1	2		3.2.9	Change Process

E. 24 Skills relatively common to earlier curricula (dropped in later models)

73	81	86	90	02	10	Skills (current as of 2012)	
3	1					1.1.11	Storage Management
1	1					1.1.12	Multiprogramming and Multiprocessing
3	3	2				1.1.13	File Systems
1	3		1			1.1.14	Machine Structures
	3					1.1.15	Computer Operations
2	2					1.1.16	Systems of Programs
2	2	2				1.1.21	Input Devices
2	2	2				1.1.22	Output Devices
2						1.4.07	Inter-systems Communications
2	1					1.4.10	Communication system hardware
	1					1.4.11	Communication system organization
3	1					2.1.11	Operational Analysis
2	1					2.1.12	Managing the IS Function
1		2				2.3.1	Historical perspectives
1		2				2.3.2	The computer industry
1		1				2.3.3	Job displacement
2	3					3.1.16	Hardware selection, acquisition, and installation for project
2		1	1			3.1.18	Maintenance Programming
	1		1			3.1.20	Decision Tools
	2	2				3.1.21	Structured development
	1	1				3.1.23	Screen Design
		1				3.1.24	Frameworks and Libraries
2		1				3.1.25	Reports Development
		1				3.1.27	Develop cash audits

F. 2 Skills added uniquely in IS2010

73	81	86	90	02	10	Skills (current as of 2012)	
					1	1.3.4	Metadata: architectures, systems, and administration

2	1.3.5	Data Warehouse: design, conversions, reporting

G. 13 Skills added based on NICE specifications (2012) but not in any curriculum model

G	1.1.20	Logic Programming
G	1.1.7	Digital Media
G	1.1.8	Software Security
G	1.2.4	Web Security and Vulnerability
G	1.3.7	Database Security
G	1.4.8	Data mapping and exchange
G	1.5.2	Security Mechanisms
G	1.5.5	Security Attacks
G	1.5.7	Information Assurance Systems
G	3.1.22	Object Oriented Development
G	3.1.26	Develop Audit Control Reports
G	3.1.28	Audit analysis of separation of function
G	3.1.29	Audit risk and disaster recovery strategies