Meeting the Challenge of IS Curriculum Modernization: A Guide to Overhaul, Integration, and Continuous Improvement

Sean T. McGann
Raymond D. Frost
Vic Matta
Wayne Huang
Management Information Systems
Ohio University
Athens, OH 45701, USA
mcgann@ohio.edu

ABSTRACT

Information Systems (IS) departments are facing challenging times as enrollments decline and the field evolves, thus necessitating large-scale curriculum changes. Our experience shows that many IS departments are in such a predicament as they have not evolved content quickly enough to keep it relevant, they do a poor job coordinating curriculum development, and they do not market the major well. For these very reasons, our IS department was on the verge of extinction, as enrollment was down over 56% (down from 475 to 208 students) over a two-year period (2001-2003), while college enrollment remained constant at approximately 1900 students. We submit that these issues can and must be addressed proactively in order for IS programs to survive. This paper conveys the approach we used to revamp our IS curriculum. We present the curriculum overhaul process and lessons learned in our successful revamp project that enabled us to transform our program into one of the most successful in our business college. Through our efforts, we have increased enrollment 75% from 2003. We hope that our lessons learned will help others facing similar challenges.

Keywords: Information Systems Curriculum Model, IS Pedagogy, Integrated Curriculum, System Development Lifecycle

1. INTRODUCTION

Our experience shows that two of the largest problems faced by IS departments are declining enrollment in the major and a lack of strategy to inform curricular decisions. We argue that the problems are not unrelated. We acknowledge that some of the declining enrollment is market driven and may be attributed to fallout from the dot.com bust, declining economy, and from offshore outsourcing. However, we found that it is far too easy to look outside, and it is futile to do so since there is little that can be done to control these external factors. Therefore, we decided to proactively address these issues by looking internally at factors we could change. In the process of assessing our curriculum, we discovered generally inconsistent courses, lack of integration across the various courses, outdated and irrelevant concepts, and almost no collaboration between faculty in ongoing curriculum development. In response, we listened to student feedback/complaints, we consulted with our IS curriculum advisory board, we conducted qualitative research, and then we took action to effect positive change.

Our qualitative research involving key stakeholders showed that, although many individual elements of the old curriculum were well designed, the IS department did not have a central strategic focus or model for the IS curriculum. Through this research, we concluded that due to our lack of focus provided by a common curriculum model and a shared vision, curricular decisions were being made in an unstructured fashion. Accordingly, we found that our courses were being changed unilaterally, which caused the curriculum to diverge and erode over time. In response, we developed a central curriculum model as a guide, which allowed us to more effectively evaluate which curriculum alternatives should be pursued and why. This strategic model, presented below, guided the creation and continuous improvement of the content of individual courses and perhaps most importantly, it promoted integration among courses. We adopted concepts from previous research on integrated curricula (Hudson and Tonkin, 2004; Maudsley, 2003) to define integration as the process of assuring that: 1) all courses leverage a common framework/process (in our case, the SDLC), 2) later courses are consistent in their emphasis on skills, tools and deliverables taught in pre-requisite courses, 3) pre-requisite courses build foundational skills, which will be used in later courses.

The IS discipline recommends a curriculum model for IS departments (IS 2002) (Gorgone et al., 2002). Although many of our courses align with that model, we found it lacking in the area of course integration. For example, the fact that systems analysis and design is isolated in one or two courses in models like IS 2002 is problematic, in our opinion. We contend that there should be a systems analysis and design component in most every IS course. We also found that the model and IS curricula in general do not feature the introductory course as a strong recruitment vehicle for undecided majors. We have discovered that using the introductory course to generate interest in the IS major is a key to increasing enrollment.

Pedagogical research in fields such as medicine consistently presents the benefits of an integrated curriculum model (Hudson and Tonkin, 2004; Maudsley, 2003). Findings are that students learn more, retain more, and perform better in the marketplace as a result of an integrated model. Further, research shows that curricula should reflect real-world needs and be driven by an overall model or framework (Scott, 2004). Based on our qualitative research and consultation with our board of advisors and recruiters, we chose the Systems Development Life Cycle (SDLC) framework (Steenkamp and Van, 2004) to serve as our curriculum model. The faculty chose this framework due to its pervasiveness in the IS field and the fact that it provides a common thread for integration across the curriculum. In implementing this model, we found that the real challenge is integrating that framework across all courses; a challenge which is ongoing.

We introduce a new model, courses, tools, and an overall strategy for the curriculum overhaul and its continuous improvement. In the account that follows, we will describe the process we utilized, the pitfalls, and the payoff. Our contribution is twofold. First, we created an overhaul process which is replicable, and can therefore be used by other IS departments to make similar changes. Second, our model augments curriculum models such as IS 2002 by demonstrating the use of the SDLC to provide the element of integration which current models lack. Although we recognize that many IS departments who have experienced declining enrollments in the past are beginning to recover without revamping their curricula, it is our position that all could benefit from a process such as the one we describe in this paper.

The remainder of the paper is structured as follows: Section II is a literature review of IS curriculum models, and integrated curriculum research. Section III describes the overhaul process. We divide the process into phases and describe what occurred in each phase, the deliverables created, and the lessons learned. Section IV offers discussion on the process, overall lessons learned, and concluding remarks.

2. LITERATURE REVIEW

2.1 IS Curricula Research

The IS 2002 report is the latest output of the model curriculum work for information systems that began in the early 1970s (Davis et al., 1997; Couger et al., 1997). It is largely the basis for accreditation of undergraduate programs of information systems. The IS 2002 model curriculum is grounded in the expected requirements of industry, represents the views of organizations employing the graduates, and is supported by other interested organizations (Gorgone et al., 2002). It identified the following four main characteristics of the IS profession: a broad business and real world perspective, strong analytical and critical thinking skills, interpersonal communication and strong ethical principles, and the ability to design and implement information technology solutions that enhance organizational performance. These characteristics are necessary to prepare students for the real world where they will need to demonstrate effective communication skills when working with clients, solving problems creatively, and working within teams (Russell, et al., 2004).

However, the IS 2002 model, as well as the previous model curricula, do not specifically address issues such as: what essential links and relationships exist between the suggested courses, and how to integrate those courses (e.g., the ten courses in the 2002 model curriculum) so that IS students can be trained to solve practical and real IS problems in organizations. Further, a review of IS 2006 (Gorgone et al, 2005), the latest IS curriculum model due to be released in 2006, still does not address this issue. Although the model does have a course referred to as the "Integrated Capstone", its concentration is on concept assimilation within that single course, and not across the entire curriculum. We argue that no real business problem can be resolved by using the skills from any IS course in isolation. In our case we found that due to our flawed curriculum design, our IS students were not integrating concepts learned from separate courses. Our research and teaching experience indicates that this lack of ongoing integration between courses is a learning barrier. We found that, when one course's concepts do not begin where the preceding ones end, or courses use inconsistent approaches, concepts and tools, students lose sight of the overall goal of the curriculum.

As a result, we are convinced that it is vital to design an IS curriculum that tightly integrates separated IS courses. Our IS department designed such an integrated curriculum and has used it over the last two years, which has resulted in positive feedback from students as well as from industry. This model leverages the fundamental SDLC concepts as a common integration framework across all courses. The successful use of this model in our IS curriculum to achieve integration demonstrates its potential, fills a gap in IS curriculum research, and would potentially fit well in the coming release of IS 2006.

2.2 Integrated Curriculum Model Research

Curriculum integration at its core requires taking concepts from earlier courses and using them in a cascading fashion in later courses in a curriculum (Hartzel, et al., 2003). Medical schools in the US pioneered the integration of scattered courses to foster integrated learning (Hudson and Tonkin, 2004; Maudsley, 2003; Sefton, 1998) through interdisciplinary integration of the humanities, clinical medicine, and basic sciences. Medical education creates integrated learning by coalescing different curriculum themes, such as basic/clinical science theme (structure and function), behavioral science theme, population science theme, and ethical and legal aspects of professional practice,

etc. (Maudsley, 2003). Some engineering programs integrate learning by combining engineering knowledge with societal and environmental factors (McCowan, 2002; McCowan and Knapper, 2002), and to a lesser extent in computer science which focuses largely on developmental solutions (Blumberg, et al., 2002; Webb, Wells and Zheng, 2001). In each of these cases, the literature demonstrates the benefits and importance of integrated curriculum models in promoting long-term retention of skills and ability to apply them in later courses, internships, and ultimately to pursue successful careers. It is obvious that revising program curricula by integrating separated courses is not new to educators in the fields mentioned above. However, we found no such research in the IS field.

Prior research in IS identified the problem of scattered courses in current IS curriculum, suggesting some changes and revisions to IS curriculum (Rosenthal, 2003). However, those limited number of prior studies focused on integrating course content for individual courses (Zack 1998), or discussing the integration issue of IS curriculum without providing a shared theoretical/pedagogical framework across different IS courses in the curriculum (Guthrie, 2004). To the best of our knowledge, there are no published papers in the main IS journal outlets on the creation and implementation of an integrated IS curriculum model.

Our central argument is that even a simple IS project requires knowledge and skills that cannot be learned from a single IS course. Specifically, solving a real system development problem requires IS courses to be highly integrated so that students can directly apply the knowledge acquired in the earlier courses to complex projects such as those found in IS capstones and internships. We acknowledge that the spectrum of IS skill sets is so large that they must be broken down into discrete areas and taught in individual courses. However, what IS pedagogy research fails to provide is a framework to promote building a cumulative repertoire which allows students to solve business problems as a whole, applying in tandem all skills learned throughout the curriculum. We introduce a framework, which takes a holistic approach to modernizing and integrating our curriculum. Leveraging the principles from previous research cited above and our own experience, we integrated our IS curriculum at many levels through the use of the SDLC to promote consistent skills, tools, and deliverables. The following is an overview of that curriculum and the process used to create it.

3. THE OU IS CURRICULUM OVERHAUL PROCESS

3.1 Overview

In this section, we detail the process used to overhaul our IS curriculum. We break the process down by phases, highlighting what occurred in each phase. We also discuss project execution details such as timelines and interim milestones, deliverables created, and initial/ongoing challenges faced and lessons learned.

3.2 Pre-Overhaul Phase (1999-2003)

We begin the story with the dot.com boom. During that time a great many students in the College of Business chose to be either IS majors, or double majors with IS. As with many IS programs, our enrollment soared to an all time high of 475 students (out of 1900 total majors on the College of Business) during this period. We had not only the most majors in the College of Business, but actually the most majors of any academic department in the University. We scrambled to hire faculty and had to cover many classes on overload contracts or with adjunct faculty.

Due to the high volume of majors, the faculty made two critical decisions. The first was to stretch resources by tripling the teaching load of the introductory course instructor. The only way to extend resources was to let the introductory course (essentially MS-Office skills) become a self-paced course with optional class attendance and minimal instructor involvement. The second decision was to begin the major with two challenging programming courses to "weed out" the uninterested and strengthen the skills of those who remained. At that point, we expected enrollment to decrease as a result of these "right sizing" measures.

However, then came the dot.com bust. The faculty expressed relief as declining enrollment eased the overload burden. However, we knew we had a significant problem in late 2002 when enrollment dipped so low (down to 295 students, a decline of 38% from 2001) that we lost a faculty line. This trend only worsened over the next year, as our numbers dropped to 208 students (a decline of over 56%) by the spring of 2003. Meanwhile, the total number of students in the College of Business remained constant at about 1850. At that point the faculty decided it was time to devise a plan to reverse the enrollment trend. But first, we had to identify root causes which were within our realm of control.

Shortly after we had begun to devise a strategy to increase enrollment, the department received a $25,000 grant from Microsoft to integrate .NET technologies throughout the curriculum. As a result, the faculty formed a committee to investigate ways to change courses to incorporate the new .NET toolset. As we examined the courses individually in this process, we began to recognize not just individual course problems, but also fundamental integration problems across the entire curriculum. We then started to realize that there was probably a correlation between these issues and our enrollment decline. At this point, the faculty decided to take a step back from the .Net integration process to initiate a larger curriculum-reengineering project. This "top-down" initiative was intended to overhaul the entire curriculum to assure that all courses were consistent, modernized, and that proper integration existed between them. The need for an overhaul gave rise to the research process which follows, whose goal was to delve further into the issues that existed and to reverse the enrollment decline.

3.2.1 Lessons Learned

3.2.1.1 Respond proactively to enrollment trends: Had we identified the severity of our enrollment problem earlier, we might have responded before enrollment declines threatened faculty lines.

3.2.1.2 Consensus on the cause of a problem is difficult in the absence of a shared vision for the curriculum model: There was no consensus as to the cause of the enrollment decline. As a result there were many theories. Surprisingly, almost no one questioned the fundamental design of the curriculum in this initial phase.

3.2.1.3 Grants can serve to enable the curriculum development process: A core team came together in the same room on a regular basis to implement change in conjunction with the Microsoft grant. These meetings led us to look at the curriculum as a whole and spawned the larger initiative. Therefore, the grant served as an enabler to change and also provided financial incentive for those core members involved.

3.3 Phase I – Preliminary Research (9/2003-12/2003)

As a result of the initial change drivers discussed above, we initiated a preliminary research project, which did point to a host of structural problems in the curriculum. Evidence accumulated from five sources: students, alumni, the IS advisory board, course evaluations, and faculty with consulting experience.

The research process consisted of roughly 30 formal and informal interviews and focus group with IS students and alumni, bi-annual meetings with our IS curriculum advisory board, and collection of IS career competency and system development models from top corporate and consulting organizations.

Qualitative analysis on this data suggested a number of themes deserving attention such as:

1) The need for tight coupling of IS concepts with experiential activities. Theory and practice were not tightly coupled even when taught in co-requisite courses. The lack of coupling created problems when dealing with a live client in the capstone series.
2) The importance of integration and consistency across the curriculum. Concepts, technologies, and life-cycle frameworks were not consistent across the curriculum. Students had trouble making connections between pre-requisite and follow-on courses.
3) The necessity for a holistic student understanding of IS business solutions as opposed to just technology. Many courses in the curriculum had a strong technology component, but did not sufficiently emphasize the context of business solutions.
4) The need for mastery of a strong systems development life cycle (SDLC) process, which aligned with those being used in industry. Only one course covered the SDLC late in the curriculum. That course presented a survey of design methodologies including some methodologies not commonly used in industry. It lacked a standardized and detailed methodology that aligned with industry. Confused and having little value for the design process, students immediately moved to development in almost all their courses—with predictable results. One of the largest shocks students faced upon graduation was the emphasis that industry placed on analysis, requirements definition, and design.
5) The critical need to revamp the introductory IS course. Our course evaluations and interviews showed that the vast majority of students who took this course considered it to be among the worst courses in the College of Business. We found that it created an extremely unfavorable first impression of the IS major and turned away many potential majors.
6) The need to deemphasize programming, as students considering the IS major perceived that programming was the main part of the curriculum. We found that the programming emphasis was a clear turn off for most business majors.
7) The importance of placing emphasis on the business side of systems development, as opposed to the technical side. Students could not differentiate what we do from what computer science students do.
8) The obligation to provide clarity on potential IS careers and how the skills we are teaching can be leveraged in industry. Surprisingly, even graduating students knew little about specific IS career paths.
9) The necessity of teaching fundamental concepts early and often throughout the curriculum. Some core IS business concepts first appeared in the capstone series. Throughout most of the curriculum, students did not understand how everything they learned fit into the larger business context. We did not present "the big picture" until they were at the end of the major, and would often ask the following question: "Why am I just learning about this (e.g. ERP, CRM, SCM, eBusiness) now?"
10) The need to de-emphasize technical details of software tools. In the interest of comprehensive coverage, some faculty presented technical details at the expense of the perspective of developing business systems.

3.3.1 Lessons Learned

Qualitative analysis helps build consensus. The challenge during this stage was to analyze qualitative data. We were fortunate to have a faculty member who specialized in such analysis. We believe that this type of expertise is essential to help identify the problem in a curriculum overhaul process, as much of the data available in these early phases is not quantifiable. Without such detailed analysis, we would repeat the error of everyone having their own favorite, but unsubstantiated theory of what the problem is. In the past, divergent theories never led to any type of real resolution. However, with this type of analytical evidence, the department quickly came to consensus on what the key problems were. We then immediately shifted to problem resolution mode, which was the formulation of a shared vision and overall curriculum model, and the institution of a process to create a sequence of courses that were true to this model.

3.4 Phase II – Organization (9/2003)

Concurrent with the research phase, the department embarked on a series of weekly meetings to quickly address the research issues raised in Phase I. The department chair moderated the meetings while the core faculty team helped set the agenda. We first established a process to reform the curriculum (see Figure 1).

Using this process, we held multiple facilitation sessions to establish a shared vision and model for the curriculum. We then addressed individual courses in light of that model. The shared vision for our curriculum specified the following:

1) The entire curriculum would be based on the SDLC model (see appendix 1) and the key focus would be on integration of courses.

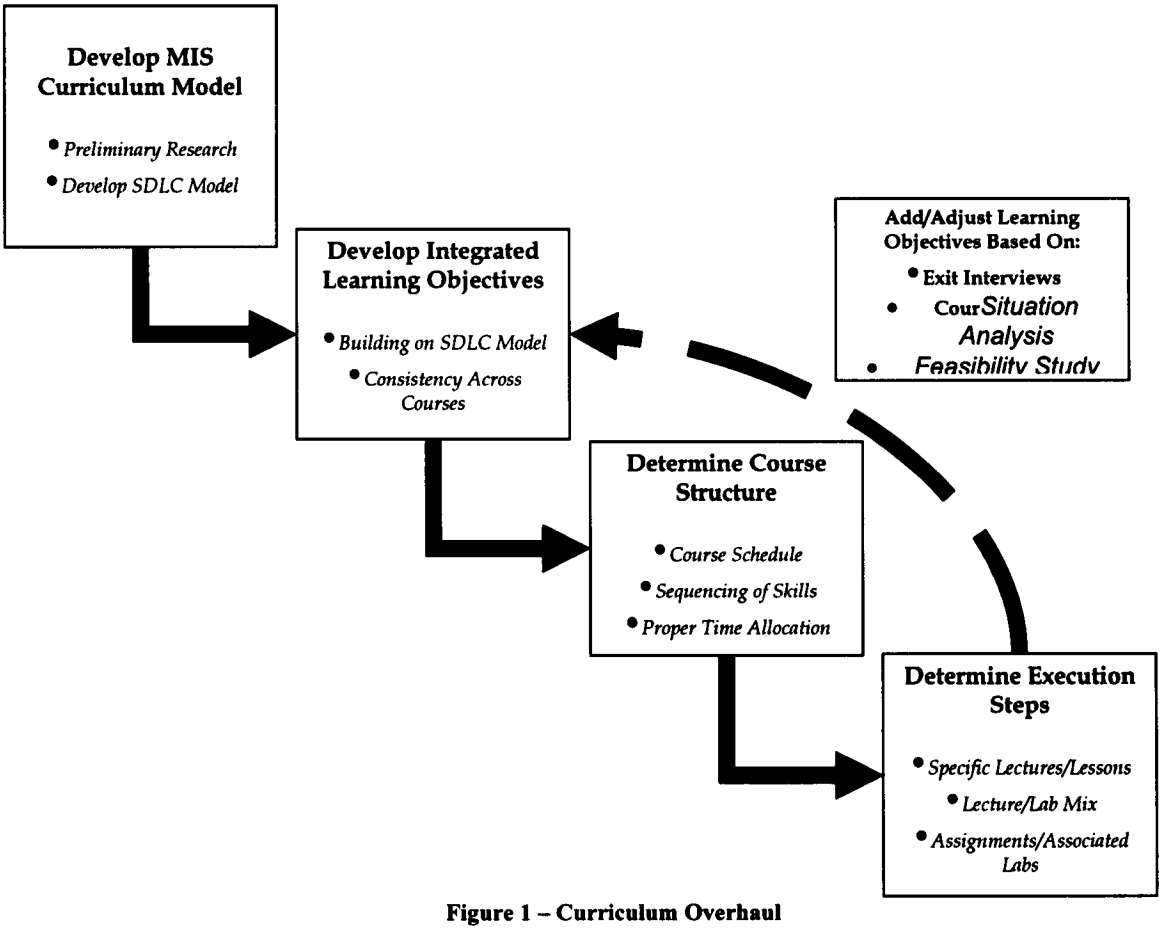


Figure 1 – Curriculum Overhaul

2) All courses would be integrated to build towards the capstone course, which leveraged all skills, tools and deliverables inherent in the SDLC model in a challenging senior client project.
3) Leveraging the power of .NET and the SDLC model, all courses would deemphasize the technical aspects of programming languages and focus on the process of building systems through the use of the SDLC. Therefore, pure programming courses (i.e. courses teaching just technical programming skills) would no longer exist.
4) Core business system concepts such as eBusiness, security, ethics, ERP, CRM and SCM would be taught throughout the curriculum beginning in the introductory courses, rather than revealing them for the first time in the capstone series.
5) Presentations and activities geared towards IS career readiness would take place at all levels to ensure clarity in this area.

3.4.1 Lessons Learned

3.4.1.1 There must be a process to create a shared vision: Without our overhaul process, each faculty member arrived at meetings with their own vision and agenda. We often moved towards "fixing" individual courses with no alignment with overall guiding principles or structure. As a result, these meetings yielded little or no progress. We also found that having a structured process to reform the curriculum was actually a novel idea to most faculty members. Fortunately, two of our members had extensive business consulting backgrounds and were therefore able to assimilate existing curriculum reform research with their past organizational reengineering experiences to design and facilitate our process. Once faculty saw results starting to emerge, they subscribed to the process and the speed of change quickened.

3.4.1.2 Faculty must commit to a weekly meeting: The challenge is to maintain the momentum for change. Initially, there were lapses of weeks between meetings and we felt as though we were continually starting over. Only a weekly meeting, fully attended, could guarantee progress and continuity.

3.4.1.3 The discussion must remain at a strategic level: Past curriculum changes focused on individual courses, or the sequencing of courses, rather than working from an overall model of the entire curriculum. Therefore, the tendency was to slip into "quick fix" mode. Our sense is that this focus on individual issues, working independent of any

type of overall process or integration guidelines is a typical cultural issue in academia, and one that anyone going through this process will face.

3.4.1.4 A moderator is key. To remedy the problem in point number 3, one faculty member served as a moderator to keep the discussion focused on the big picture. In our case, this moderator was one of the instructors with a consulting background who designed the process. The moderator proved invaluable at helping assure that the overhaul process and high-level strategy were in place before we moved on to course-level issues.

3.5 Phase III – Building the Curriculum Model (10/2003-12/2003)

Considerable adjustment of the initial SDLC model occurred during this phase. Upon assessing each course, we added different elements to the overall model to incorporate best practices for teaching and deliverable creation. The result of this process was the SDLC model shown in appendix 1. This model shows all phases of the SDLC and the skills required/tools used in each phase.

In conjunction with this model, we also constructed a substantial guide for all SDLC projects called the MIS Handbook. (The handbook as well as other materials described here are available upon request from the authors). This handbook defines each phase and provides guidelines for the consistent creation of deliverables during each step in the systems development process. We benchmarked the handbook against similar life cycle handbooks from industry. In the interest of achieving consistency across the curriculum, the faculty immediately adopted the handbook in all courses with SDLC projects. We decided the use of this guidebook would mimic the process in industry where employees are handed a life cycle handbook when they start a new job and consult it frequently throughout their tenure as IS professionals.

As the SDLC model neared completion, a discussion ensued as to whether we should cover just part of the SDLC in a given course or teach the entire SDLC in all courses. Ultimately, we decided to cover the entire SDLC in every course in order to promote consistent, holistic understanding on the part of the students. A parallel discussion also concerned the depth to which each phase would be covered in each course. The faculty decided that the earlier courses would address the SDLC at a less detailed level, and depth would increase as students progressed through the curriculum, culminating in applying all SDLC skills, tools, and deliverables in the IS capstone series.

After settling on the SDLC model, the faculty then designed the new course sequence. Based on our research, we decided to change the courses as shown in table 1 below.

The new course sequence reflects the following major changes:

1) A complete redesign of the introductory course (MIS 201) from a self-paced MS Office course to a marquee course representative of the new IS SDLC model. The course features SDLC projects focused on the design and development of business systems and on the development of deliverables to support business decision making. We designed the course to serve as a recruiting tool for new IS majors. This course creates the first impression of the department and the IS major. Therefore, the design team expended great effort to make it an engaging, positive experience. We also deployed our best professors to teach the course (in the past, only junior faculty and adjunct instructors taught the introductory course).
2) A change to the MIS 202 class to include emphasis on IS business core concepts and career readiness as well as SDLC projects.
3) A movement to de-emphasize programming and technology. The new IS core begins with only one class that involves programming (MIS 220). However, we redesigned this course to focus on leveraging the SDLC to develop business solutions, with programming playing a lesser role. Further, we repositioned the other programming class (formerly MIS 225 (Java), now MIS 400) as a systems integration course that appears later in the curriculum. The faculty made this change to avoid the problem discovered in the old course sequence of overwhelming students with back-to-back programming classes immediately as they begin the IS core. We designed the new MIS 400 course to deemphasize programming, consistent with our new guiding principles. This move also involved switching toolsets from Java to C# in MIS 400, as part of the move to the .NET platform. We moved Java to an elective position.
4) Combining the new systems analysis, design and development class (MIS 320) with the database design and development class (MIS 380). We combined the courses to promote tighter integration between the concepts of these classes. We decided it did not make sense to teach an in depth system design and development and database course separately, as they are inextricably linked in the SDLC process. Combining the courses allowed for more detailed projects, which comprehensively leverage all skills, tools, and deliverables in the SDLC model.

3.5.1 Lessons Learned

3.5.1.1 Consensus on SDLC phases and associated deliverables is difficult to reach: No accepted standard for the SDLC exists in industry. Our research showed different companies using anywhere from four to eleven phases for the SDLC. Similarly different faculty had their own set of SDLC phases and associated deliverables. Students are sensitive to these types of changes in the curriculum and the differences make it difficult to integrate courses. Considerable discussion took place over a period of weeks to agree upon the phases and associated deliverables. We believe that these discussions helped to align the faculty team with the same SDLC model and achieve our goal of end-to-end integration within the curriculum.

3.5.1.2 The SDLC adopted by the department must be documented: We spent an entire summer developing the MIS Handbook outlining in detail the SDLC and its associated deliverables. All faculty adopted the handbook in order to maintain continuity and integration among courses. The results have exceeded our expectations. All instructors have perceived

	Old Course Sequence		New Course Sequence
201	MS Office Tutorials	201	Information Analysis and Design
202	IS Fundamentals	202	Business Information Systems (SDLC Project, IS Business Fundamentals and Core Issues)
220	Programming fundamentals with VB	220	Application Development using the SDLC and VB.Net
225	Object Oriented Programming with Java	320/380	Systems Analysis, Design and Development (ASP.NET) & Database Design and Development (Oracle and Access)
320	Systems Analysis and Development (Classic ASP)	325	Networking and Hardware
380	Business Database Design (Oracle)	400	Systems Integration and Web Services
325	PC LAN Applications	420/485	Capstone Series (Advanced Systems Development with Live Client and Key Concepts, Systems, and Issues)
420	Systems Development: Design and Implementation	Electives	XML, ERP/Supply Chain, Java, UML
485	IS Capstone		
Electives	Distributed Systems, Groupware Apps, XML, Designing for Web & beyond		

Table 1 – Course Sequences

an increase in deliverable quality and consistency. Students also comment frequently on the usefulness of the handbook and the fact that they are finally being taught SDLC concepts in a consistent fashion, which minimizes their confusion regarding the correct way to create SDLC deliverables.

3.5.1.3 Once the SDLC is adopted and documented, the process of constructing individual courses can begin: In the absence of consensus on the SDLC it would be pointless to proceed to individual course-level design. Initially, we tried working on individual courses before completing the model and quickly saw that we were repeating old mistakes. Courses immediately diverged. We then realized that the SDLC provided the needed roadmap to the consistent individual course creation.

3.6 Phase IV – Designing and Developing Individual Courses (10/2003-Present)

After completing the curriculum model, our process called for designing individual courses in accordance with the SDLC model. In this process, the department formed subcommittees to construct each new course. They then presented each new course to the department overhaul team for critique and approval. The committee prioritized the courses to be implemented and decided to form a subcommittee to address the introductory IS course, MIS 201, due to the paradigm shift regarding its importance as a recruiting tool. As this course integrated closely with the second course in the business core, Business Information Systems, MIS 202, a parallel initiative began to redesign this course. This section outlines the process undertaken to design and implement these two courses. Although we redesigned every course in the curriculum, we present these two examples below in the interest of brevity.

3.6.1 Integrating the Two Business Core Courses: Like many business IS departments, we had two courses in the business core—one dealing with personal productivity software (MIS 201- Introduction to Microcomputer Applications) and the other with IS concepts and theory (MIS 202 - Business Information Systems). The MIS 201 personal productivity software course was just a service course to the College of Business. Through our research, we discovered that students did not take it seriously due to its loose structure and lack of substance. Business Information Systems, MIS 202, provided the first real introduction to the IS major. However, we realized that from a student standpoint, the courses are all branded "MIS"—and therefore creating a positive first impression of the major is of paramount importance. We decided that it was poor marketing practice to concede that the first time students experience MIS, the class would be a perfunctory PC skills class such as the old MIS 201. We saw that even the introductory courses must be a solid representation of the curriculum that we worked so hard to improve. Therefore, the faculty decided to design the MIS 201 and 202 classes as a microcosm of the entire IS major, providing students with a high-quality, comprehensive sample of the IS curriculum. In this two-part series, students experience the entire SDLC, use a number of tools such as Photoshop, Excel, MS Project, Access, and ASP.NET to design and develop simple systems, which solve business problems. They also

participate in activities and discussions designed to familiarize them with the IS business world, core systems, concepts, and issues. The emphasis placed on the introductory courses was a complete shift from the past paradigm that labeled these courses as unimportant. Our goal became to ensure that through MIS 201 and 202, students gained a solid grasp of what the IS major entailed, the quality of the curriculum, and familiarity with the IS faculty and department culture; all in the interest of increasing enrollment.

3.6.2 Repositioning MIS 201 as Information Analysis and Design: Consistent with our decision to use the SDLC model in all courses, we decided to begin with the first introductory course. We repositioned the course as Information Analysis and Design rather than basic PC skills. This new focus represented a strategic shift as we moved away from the previous emphasis on tools, to a focus on the SDLC as a process for developing systems. The key here was helping students understand that tools are merely a means to assist with the SDLC process. The guiding principle became that tools will come and go, but the SDLC is a fundamental process, which can be applied in any business situation to solve problems.

Therefore, we decided to create a course that would meet the following design objectives:

1) Incorporate the SDLC as a problem solving methodology.
2) Become a strong recruitment vehicle into the major.
3) Focus on the importance of design before development.
4) Incorporate examples from other areas of business (accounting, finance, management, marketing) so that students could see the connection between IS and the other disciplines.
5) Focus on using information for decision making.
6) Focus on visual design deliverables such as screen prototypes, graphs, and charts. Our research showed that students find visual deliverables more interesting to develop. Visual deliverables are also supported by rich theory in usability and analytical design. (Tufte, 2001)
7) Be interesting, engaging, and fun.

As we could not find an existing course that met these design objectives, we put a process in place to create the course. First, we assigned one of our top professors to create the course. Second, we assigned our best teaching assistant to aid that professor in the creation and delivery of materials. The professor and teaching assistant formed the core design team. Third, we had the design team seek input from practitioners from industry especially in the area of commercial web design. Fourth, we gave the design team latitude to think outside the box. The team included material from outside of the discipline—most notably the work of Edward Tufte on analytical design (Tufte, 2001). The faculty also allowed the team to introduce Photoshop to the course—a design tool that our experience showed was traditionally not found in the introductory IS course.

A series editor for a major publisher commented that the course design was the one of the more original ideas to cross his desk for the IS introductory course in years. The design team is now developing a textbook to support the course.

3.6.3 Integrating MIS 202, Business Information Systems: Revising MIS 202, the Business Information Systems course, was also a challenge. In the years prior, faculty experimented with different approaches in different sections of the course. For example, one faculty member focused on having students design process flows for a system, while another faculty member focused on application development and yet another on MS Access skills. The subcommittee decided to merge and modify all existing approaches. Further, the course modeled the capstone course, dealing with core IS business concepts, systems, and issues. Students now learn how IS fits within the value chain and is a strategic driver of business. Case studies are used extensively, and we were fortunate to find an introductory text, "Business Driven Technology" (Haag 2006), which supports our model by emphasizing the role of IS in business, and de-emphasizing technology. Students also learn about specific IS business concepts and trends such as ERP, SCM, wireless, and outsourcing. However, to maintain integration/continuity with MIS 201, the Information Analysis and Design course, system development and creation of related IS deliverables using the SDLC framework still comprise 30% of the course. The SDLC process in MIS 202 culminates in the creation of a simple web-based contact management system developed in ASP.Net. Our finding is that creating a rudimentary business system helps students better understand the more complex systems that they are studying.

We arranged the following process to develop the MIS 202 course:

1) The same design team from MIS 201 - Information Analysis and Design would develop and test the materials.
2) The textbook would be custom published to include just the materials that applied to the course. We custom published for two reasons. First, students received a better price on the book. Second an abbreviated text allowed us to focus the course on depth rather than breadth without having students question why they had to purchase a large text including material not covered in the course.
3) All faculty delivering the course would meet once a week to design future lesson plans and to analyze what needed improvement from the prior week.

3.6.4 Lessons Learned

3.6.4.1 Top teaching assistants make excellent members of a design team: Teaching assistants are more in touch with the student experience and therefore know what themes will resonate positively with students in a course designed as a recruitment vehicle. A key success factor in this area was our willingness to treat teaching assistants more as colleagues than students. Working with teaching assistants turned out to be an invaluable move for us. We not only benefited from their perspective as students, but also were able to continue our normal responsibilities in addition to this difficult course development process, as they shouldered much of the time consuming work (e.g. documentation).

3.6.4.2 The SDLC must be repositioned as a problem solving process in the introductory course: Another challenge faced in the introductory course was how to work the SDLC into a course in which students traditionally do not create systems. We found that students at this level perceive systems development as too challenging. Our solution was to position the SDLC at this level as a "business problem solving process" rather than an IS development methodology. This repositioning turned out to be a highly effective solution, which helped us achieve buy in from students more readily.

3.6.4.3 Course design and development from a brand new/innovative model is difficult: The lack of supporting textbooks or similar courses to build upon makes the process even more difficult. It is also a slow, iterative process. What is required is a vision for what the course should be and faculty dedicated to achieving that vision, despite initial resistance from students due to the fact that the course is more rigorous and challenging.

3.7 Phase V – Course Implementation Process (1/2004-present)

While in an ideal world we would develop the materials for the new course over a period of at least a year, in reality we were under tremendous time pressure. From our perspective, our enrollment problem was one which potentially threatened the viability of the major and which required a rapid implementation approach. Therefore, the process for implementation called for just-in-time delivery of course materials in accordance with the following steps:

1) Have the design team pilot the delivery of the course.
2) Have the design team create and review all materials internally.
3) For assignments that integrate other disciplines, create a peer review process for the course with at least two faculty members in the associated discipline (e.g., accounting, finance, marketing, management).
4) Pre-test the materials with student independent study volunteers prior to implementing them in the course.
5) Emphasize knowledge transfer by having faculty attend the new course prior to teaching it themselves.
6) Keep the design team involved in the continuous improvement process even after the handoff to other faculty.
7) Have the same design team continue to overhaul the next course in the sequence in order to ensure points of integration.

3.7.1 Lessons learned

3.7.1.1 Students should be given the option to opt-in to the pilot of the new course: We piloted the new introductory course simultaneously with the old introductory course but did not give students the option to choose which section they took. What we did not realize is that students anticipated an easy "A" (as the old course was notoriously simple) and constructed their schedules accordingly. The promise of improved material was little compensation in their eyes, especially for the non-IS majors. In retrospect, during the transitional quarter we should have given them the option to choose the new or old course after an explanation of each.

3.7.1.2 Just-in-time delivery of materials is challenging to coordinate: Since our process required the good will of colleagues and students for review and testing of materials, inevitable delays in the process led to large problems in the classroom. The design team also ran into internal delays as some materials turned out to be difficult to develop. Those attempting such rapid implementation should be aware that such delays and coordination problems will surface, set expectations accordingly amongst team members, and remain flexible.

3.7.1.3 Attending the pilot aids subsequent delivery: The faculty that would inherit the course attended the pilot to learn the material and delivery in the classroom. Previewing the course was a departure from the normal handoffs that often take place in academia. In our case, transitions usually consisted of simply handing off a syllabus. We found the previewing step enabled integration across the curriculum as we overhauled each course.

3.7.1.4 Having members of the design team attend the course after hand off helps with continuous improvement: Sitting in on a course that one developed is a real eye opener. It provides clear insight as to what works and what does not in the classroom. This post-view process, although time consuming, helps maintain integration and further promotes continuous improvement.

3.7.1.5 Be prepared initially for lower course evaluations: The most discouraging aspect of the course redesign process has been its initially thankless nature. We found that students are critical of any of the inevitable problems that emerge in assignments, testing, and grading in the implementation of a new course. Unfortunately, they expressed their displeasure in the course evaluations during the first quarters of implementation. However, with the MIS 201 course, they rebounded significantly from an average of 3.8/5 in the initial quarters, to nearly 4.4/5 after our third iteration.

3.7.1.6 Preserving the same design team for a pre-requisite and the follow-on course leads to better integration: Having a consistent design task force on courses that integrate with each other has obvious consistency and integration benefits. However, we found that leveraging a given design team must be done carefully, due to the risk of burnout. To mitigate this risk, we alternate members in and out of the process, with overlap to assure continuity.

4. DISCUSSION AND CONCLUSION

Since the inception of this project in the fall of 2003, we made progress in the overhaul of our curriculum and the movement towards our goals. We continue to make progress in the revamp and continuous improvement of our courses and are enjoying the positive results. Enrollment is up nearly 60% over the same time last year. Student satisfaction, as indicated by exit interviews and course evaluations, is also showing a highly positive trend. Student comments indicate that the increase in enrollment is strongly tied to the new curriculum initiative. It is typical now to hear upperclassmen

comment that they wish they had the opportunity to take the newer versions of our IS courses. Employers are highly enthused about the curriculum changes and are offering positive feedback, while aggressively hiring our students. We were near 100% placement for the 2005 graduating class, up from less then 75% last year, and 60% the year before. New relationships with a number of large businesses evolved largely on the merits of our new curriculum and its relevance to industry.

Looking back on our accomplishments, key milestones achieved thus far include:

- Integration of the SDLC and MIS Handbook across all courses.
- Successful conversion of our introductory course, MIS 201, to our marquee course, leveraging the SDLC, and tools such as Photoshop. We refined the course through five quarters and it is now well received by students. Average course evaluations stand at 4.4/5; up from a 3.8/5 average for the old course.
- The new MIS 202 course, which is a continuation of MIS 201, was successfully launched and is in its fifth quarter of operation. Although course evaluations declined in the first quarter of implementation, they are on the increase, showing a 4.4/5 average (up from 4.05/5).
- The faculty converted MIS 220 from a programming-focused course to a systems development course focusing on the SDLC process and deemphasizing programming. VB.NET is the tool used in this class in accordance with the .NET conversion project. In its fourth quarter of operation, course evaluations are near 4.5/5, which is an increase for this class.
- The department integrated MIS 320/380, Systems Analysis, Design, Development and Database, into a single systems development class with entirely new content. The course delves into more detailed SDLC problems, and emphasizes database and system development theory using ASP.NET and MS Access as the primary toolsets. In its fourth quarter of operation, this class also shows high course evaluations, with a 4.3/5 average, up from 3.9/5 previous to this project.
- MIS 400, Systems Integration, is in the piloting phase, and will be the course of emphasis for the overhaul team this year.
- The department merged the MIS 420/485 IS Capstone Series into one course that integrates the SDLC and MIS Handbook into the Senior Client Project. Course evaluations are high in this course as a result of the improved consistency of the curriculum, at a 4.9 average, up from 4.5 prior to the overhaul process.

4.1 Critical Success Factors

In an effort to provide a basis for other IS departments to initiate this process, promote curriculum integration and individual course development, we cite the following critical success factors:

4.1.1 Cultivate a culture that values teaching: There must be an organizational culture supportive of curricular innovation. The profession can help out here by rewarding pedagogical research. If top journals like this one are willing to publish pedagogical research, then more research and innovation is likely to take place.

4.1.2 Grants help enable change: Although we had already begun discussions about this change process prior to receiving the funding, the Microsoft grant was a key enabler of this change. The money helped us formalize the process and gave us a means with which to offer additional compensation to the key team members.

4.1.3 Cultivate collegiality: A collegial environment is key. After the initial buy in took place, there were no detractors on the core departmental team. In fact, colleagues supported each other's efforts to venture into uncharted waters. Another corollary benefit was that getting through this process as a team increased the level of collegiality and collaboration.

4.1.4 Engage a Moderator: A moderator is essential. The temptation to focus on course details early in the process was great. The moderator kept us on track.

4.1.5 Require all deliverables in an SDLC format: Beginning with the introductory course, Information Analysis and Design (MIS 201), students must present all of their deliverables in the SDLC format. As students progress through the curriculum more deliverables are added until complex deliverables are fully realized in the Capstone Series (MIS 420/485).

4.1.6 Use .NET technologies across all courses: Beginning with the Business Information Systems course (MIS 202), students produce development deliverables using .NET technologies. By choosing a single development toolset, consistency across the curriculum was promoted. Also, in keeping with our de-emphasis of technology/programming, .Net is designed to be much more visual through the use of pre-packaged controls (e.g. log in pages) and "drag and drop" functionality. This greatly reduces the amount or programming necessary to deliver solutions. At MS Tech Ed 2004, where we became more familiar with the tool, Microsoft representatives consistently stated that it was designed to eliminate up 80% of the coding required. The first exposure to .NET in MIS 202 is purely through the visual interface with no "code behind" programming. In fact, the design work is done in Photoshop and then sliced for development in .NET. As students progress through the curriculum, they must delve beneath the visual interface to the code view to incorporate advanced functionality. However, throughout all the courses, analysis and design come first. The precedence of analysis and design is a key point of integration supported by .NET, which allows students to focus less on the coding and technical details in building systems.

4.1.7 Teach design theories early and repeat them often: Starting with the introductory course (MIS 201), students develop techniques for scoring web site and application usability and then carry them forward throughout the curriculum. Therefore, students are able to critique site designs all the way through to the capstone course (MIS

420/485). In MIS 420/485 students work with a live client that often has a pre-existing system in need of critique and re-design. Similarly, students learn principles of analytical design and then carry those forward to their other courses. We found that these principles together with usability theory help to better inform interface design. The principles also provide a point of integration with the rest of the business curriculum since they are applicable in almost every course.

4.1.8 Assign the same faculty member to teach co-requisites: By design, all co-requisites within the major have points of integration (e.g. MIS 320/380 and MIS 420/485). For years we tried to coordinate delivery between two faculty members. However, it turns out that students are sensitive to even minor differences in standards between courses that are supposed to be synchronized. We recently decided to have the same faculty member teach both co-requisites to increase integration. Students express more satisfaction as indicated by course evaluations and exit interviews.

4.1.9 Assign the same faculty member to teach a course and its pre-requisite: We have learned that the best way to know what happens in the pre-requisite course is to teach it. After doing so, we have observed that faculty members with pre-requisite exposure are better positioned to provide integration with the follow-on course (e.g. same instructor for MIS 201 and 202).

4.1.10 Break down course integration barriers: We see the issue of course integration barriers (i.e. lack of integration between courses) being resolved in two ways. The first is to extend material that does not fit in a pre-requisite course into the follow-on course. The second is to offer two courses concurrently as though they were one large course. For example, the material that we developed for Information Analysis and Design (MIS 201), exceeds what can be covered in a quarter. Therefore, we extended three weeks of the material forward into the Business Information Systems course (MIS 202). This extension created a natural integration because the material brought forward is necessary for the students to create the contact management system required in MIS 202. Breaking down barriers also occurred in our co-requisite courses. When the same faculty member teaches both co-requisites, for example MIS 320/380, they can strategically allocate more or less emphasis to each of the courses in any given week. This flexibility is a benefit and has the effect of making the co-requisites appear to the students as just one large course.

Although we do consider this project to be a tremendous step forward for our IS department, we acknowledge that it is an imperfect process. We still have a long and arduous process ahead. We also acknowledge limitations in our research process. Although we did go to great lengths to accumulate and analyze data to uncover our curriculum issues, we made the conscious decision to focus more on finding problems quickly than on adhering tightly to the rigors of qualitative research methodologies. Therefore, our findings in this paper are largely anecdotal. However, our purpose in writing this essay was not to generate research findings grounded in theory and methods, but instead to tell our story, with an emphasis on relevance instead of rigor. This story shows how we adopted a consulting mentality of reengineering our business, without the luxury of being able to take time for methodical research structure. However, having apparently begun to reverse our enrollment decline and successfully stabilized our position, we have initiated a more rigorous research process. These studies are designed to assess the effectiveness of our curriculum innovations and understand the dynamics that occurred. The goal of this future research is to analyze how well our courses prepare students for careers in IS, from the perspective of past and present students, and peers who have not taken our IS courses as well as their employers. We also plan to examine the overhaul process and resulting organizational dynamics through the lens of change management to examine potential contributions to that research stream. From these initiatives, we hope to offer a number of contributions to pedagogical research in IS in the future.

Despite the limitations of our methods, our paper still makes a number of contributions. First, it provides guidelines for establishing a curriculum revamp process, including a process model, lessons learned, and key success factors. Second, it puts fourth an overall IS curriculum model that demonstrates how to achieve integration across courses. Third, it establishes a set of guidelines and innovative ideas for designing and implementing individual courses. Fourth, it makes a contribution to established IS curriculum models such as IS 2002 and IS 2006 by demonstrating the importance of integration across such models and also providing an approach for attaining such integration in the use of the SDLC Model and the associated MIS Handbook. It is our position that these existing models would benefit from a focus on integration.

Through this trying process, we realized that despite our interim victories in the past two years, we have embarked on a mission that will likely never end. A change like this one is not without drawbacks. In many ways, it makes all of our lives more difficult. Our teaching lacks stability. We are chasing a moving target, which is the continuous improvement of each course. We are faced with a steep learning curve in adopting new tools, creating entirely new content, and developing new teaching approaches. However, despite all of the challenges, the consensus is that none of us would return to old ways of developing and maintaining our curriculum. Although stressed by the process, our sense of satisfaction, accomplishment, and overall collegiality more than outweighs the challenges faced.

As we have told this story at conferences and other academic forums, we discovered a high level of interest in our approach from other schools. This interest is a key reason we took the time to write this article. We have come to see that there are many others in this predicament, and we would like to assist. Although this article only represents a sample of our curriculum project story, we are happy to share more with any of our peers who would like our help. We invite any and all IS professors with an interest in curriculum improvement to contact us for copies of our materials, advice or to offer criticism and ideas of their own.

5. REFERNCES

Blumberg, B., M. Downie, I. Y., M. Berlin, M. P. Johnson and B. Tomlinson (2002) "Integrated learning for Interactive Synthetic Characters", ACM Transactions on Graphics, Vol. 21, No. 4, pp. 417-426

Davis, G. B., J. T. Gorgone, J. D. Couger, D. L. Feinstein and H. E. Longenecker Jr (1997) "IS '97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems", ACM, New York, NY and AITP (formerly DPMA), pp. 101 - 194

Gorgone, J., P. Gray, E. A. Stohr, R. T. Wigand and J. S. Valacich (2005) "MSIS 2006 Curriculum Preview", Communications of AIS, Vol. 15. No. 30, pp. 544-554.

Gorgone, J. T., G. B. Davis, J. Valacich, H. Topi, D. L. Feinstein and H. E. J. Longenecker (2002) "IS 2002 Model Curriculum And Guidelines For Undergraduate Degree Programs In Information Systems", Communications of AIS.

Guthrie, R. W. (2004) "Integrating Programming and Systems Analysis Course Content: resolving the Chicken-or-the-Egg Dilemma in Introductory IS Courses", Information Systems Education Journal, Vol. 2, No. 27, http://isedj.org/2/27/, ISSN: 1545-679X

Haag, S., P. Baltzman, A. Phillips ((2005) "Business Driven Technology", McGraw-Hill/Irwin

Hartzel, K. S., W. E. Spangler, M. Gal-Or and T. H. Jones (2003) "A Case-Based Approach to Integrating an Information Technology Curriculum", Information Systems Education Journal, Vol. 1, No. 47. http://isedj.org/1/47/. ISSN: 1545-679X.

Hudson, J. N. and A. L. Tonkin (2004) "Evaluating the impact of moving from discipline-based to integrated assessment", Medical Education, Vol. 38, No. 8, pp. 832-843

Maudsley, G. (2003) "The limits of tutors' comfort zones with four integrated knowledge themes in a problem-based undergraduate medical curriculum (Interview study)", Medical Education, Vol. 37, No. 5, pp. 417-423

McCowan, J. D. (2002) "An integrated and comprehensive approach to engineering curricula, part two: Techniques", International Journal of Engineering Education, Vol. 18, No. 6, pp. 638-643.

McCowan, J. D. and C. K. Knapper (2002) "An integrated and comprehensive approach to engineering curricula, part one: Objectives and general approach", International Journal of Engineering Education, Vol. 18, No. 6, pp. 633-637

Rosenthal, P. (2003) "Strategy Course and Integration Course in redundancy in the MSIS2000 Model Curriculum", Information Systems Education Journal, Vol. 1, No. 14. http://isedj.org/1/14/ , ISSN: 1542-7382.

Russell, J., B. Russell and W. J. Tastle (2004) "Teaching Soft Skills in a Systems Development Capstone Course", Information Systems Education Journal, Vol. 3, No. 19. http://isedj.org/3/19/, ISSN: 1545-679X.

Scott, E. (2004) "Systems Development Group Project: A Real World Experience", Information Systems Education Journal, Vol. 4, No. 23. http://isedj.org/4/23/, ISSN: 1545-679X

Sefton, A. J. (1998) "The future of teaching physiology: An international viewpoint", Advances in Physiology Education", Advances in Physiology Education, Vol. 20, No. 1, pp. 53-58

Steenkamp, A. L. and D. J. Van (2004) "An Approach to Teaching IT Life Cycle Processes", The Proceedings of ISECON 2004, v 21.

Tufte, E. R. (2001) The Visual Display of Quantitative Information, Graphics Press.

Webb, G. I., J. Wells and Z. J. Zheng (1999) "An experimental evaluation of integrating machine learning with knowledge acquisition", Machine Learning, Vol. 35, No. 1, pp. 5-23

Zack, M. H. (1998) "An MIS Course Integrating Information Technology and Organizational Issues", The Database for Advances in Information Systems, Vol. 29, No. 2.

6. AUTHOR BIOGRAPHIES

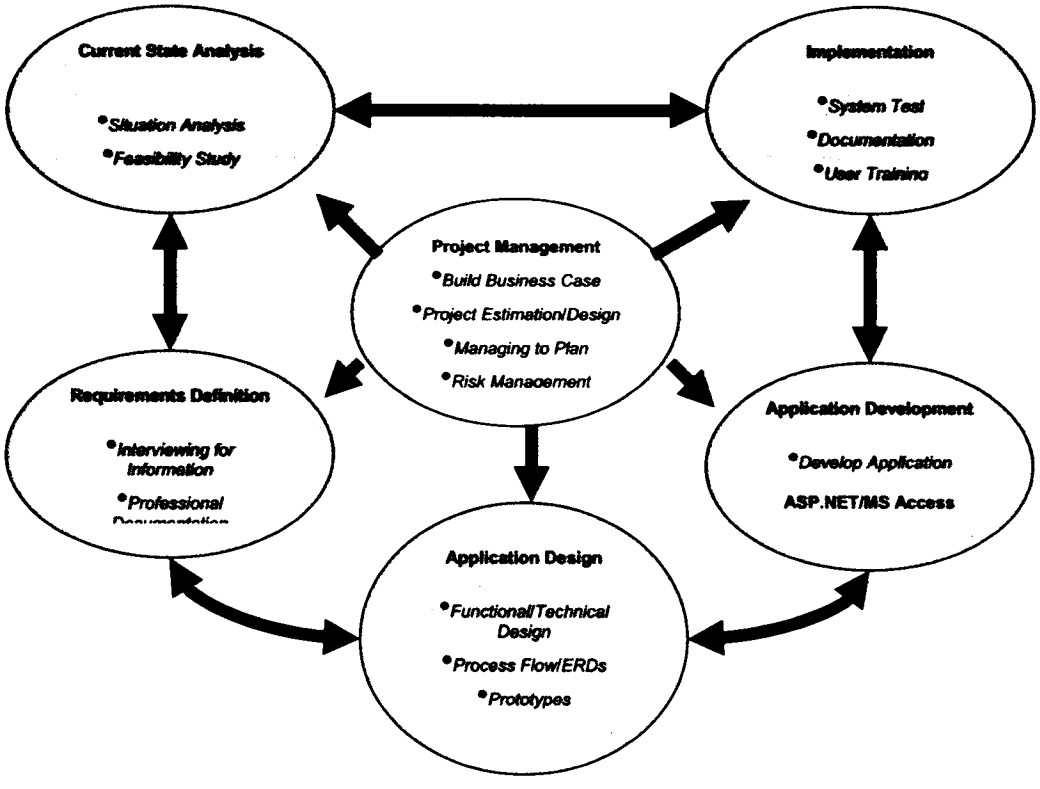
Sean T. McGann (mcgann@ohio.edu) is an Assistant Professor of Information Systems at Ohio University. His research interests include IS Pedagogy and Curriculum, IS User Improvisation, Inter-organizational Systems and Supply Chain Systems. He teaches systems development, systems analysis and design and database courses. He also serves as faculty Advisor to Ohio University's chapter of the Association for Information Technology Professionals. He has received teaching awards such as MIS Professor of the Year and The Senior Student Recognition Award. Sean earned a Ph.D. in information systems from Case Western Reserve University, an MBA from Ohio University and a B.S. in Electronics Engineering from Bowling Green State University. He spent 5 years in Andersen's Business Consulting Practice, 2 years as CEO of Pogonet Internet Solutions, Inc, and continues to operate an independent systems consulting company.

Raymond D. Frost (frostR@ohio.edu) is a Professor of Management Information Systems at Ohio University. He publishes scholarly papers in the information systems and marketing fields. Raymond is co-author of *E-Marketing* and *Database Design and Development: A Visual Approach.* He teaches database, systems analysis and design, and information analysis and design courses. Raymond received numerous teaching awards including the University Professor and Presidential Teaching awards from Ohio University. He is currently working on publications in data modeling, pedagogy, and information analysis and design. Raymond earned a Ph.D. in business administration and a M.S. in computer science at the University of Miami (Florida), and received his B.A. in philosophy at Swarthmore College.

Vic A. Matta (matta@ohio.edu) is an Instructor of MIS, also pursuing a doctorate in the field at Ohio University. He has ten years of industry experience in information technology (IT), project management, distributed computing, and database management. He has worked on several projects for scalability, capacity, and feasibility planning for high bandwidth, rich media, and client-server applications. His current areas of interest are IT in the supply chain, Radio Frequency Identification Systems, and curriculum development in information systems.

Dr. Wayne Wei Huang (huang@ohio.edu), Professor at MIS Department., College of Business, Ohio University, USA. He has worked as a faculty in universities in Australia, Singapore, China, and Hong Kong before, and received research awards in universities of Australia and USA. Wayne has had more than 15 years of full-time teaching experience in universities as well as a few years of IT industrial working experience (as a system analyst/programmer). His research combines both quantitative and qualitative research methodologies. He published more than 10 books and/or book chapters on Information Systems internationally. He has published more than 80 refereed research papers in international journals and conference proceedings, including some leading international MIS/IS journals such as IEEE Transactions on Systems, Man, and Cybernetics; Journal of Management Information Systems (JMIS); Communications of ACM (CACM); IEEE Transactions on Professional Communication; Information & Management (I&M); Decision Support Systems (DSS); Communications of AIS (CAIS); International Journal of Global Information Management (JGIM); and European Journal of Information Systems (EJIS). Wayne's research work and published papers have been cited by the top 5 international IS/MIS journals, including MIS Quarterly (MISQ), Journal of MIS (JMIS), Information Systems Research (ISR), and IEEE transactions. He is currently a Senior Editor of International Journal of Data Base: Advanced in Information Systems, an ACM publication, USA; and on the Editorial Boards of Information & Management (I & M), International Journal of Global Information Management (JGIM), Journal of Database Management (JDM), USA

Appendix 1 – SDLC Curriculum Model



Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096