ISSN: 2167-1435
v31 n3073

Improving an Information Systems Capstone/Consulting Course for Non-Traditional Undergraduate Students

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

Abstract

This work reports on lessons learned from the first offering of an information systems capstone project for non-traditional bachelor's degree students at the University of Richmond. Capstone projects have long been offered in lieu of comprehensive thesis papers for information systems master's degree programs, and to integrate core competencies in both masters and undergraduate programs. It is accepted that such a course is ideal for pulling together and integrating core competencies of the information systems bachelor's degree program and certificate program for those already having an undergraduate degree in another field. These students are non-traditional students at the University's School of Professional and Continuing Studies. Little change to the syllabus from master's degree syllabi was made for this initial course offering. It was immediately obvious that significant changes to this course would be required for future offerings to tailor the course to extend its value to the target population. The goal is to maintain the realism of a real-world capstone/consulting project, while ensuring academic goals are met, and students realize value from the process. The author uses observation, course outcomes and evaluations, and reviews the literature to suggest ways to improve future offerings of the course.

Keywords: Capstone, Consulting, Information Systems, Capstone Projects, MIS.

1. INTRODUCTION

Background

This paper report on lessons learned from development and presentation of a new capstone/consulting course for non-traditional undergraduate students at the University of Richmond's School of Professional and Continuing Studies (SPCS). Lessons learned from the first presentation of the course are discussed, and efforts to improve certain aspects of the course based on these lessons are detailed.

Although there is no widely accepted definition of a non-traditional student, demographic information helps place the SPCS student in perspective. The mean age of the SPCS student is 37. Although the gender breakdown of SPCS students is 71% women and 29% men, the ratio is reversed for information systems majors, with the vast majority of Management Information Systems (MIS) majors being men. Forty-two percent of students are undergraduate, 40% graduate (there are no MIS graduate programs), and 18% are non-degree seeking students. Eighty-one percent of students are enrolled part-time, with 19% fulltime. There is wide variance in all demographic variables among SPCS

students. Some are attempting to complete degrees started at a community college, others are older and seeking a bachelor's degree for promotion purposes, while many are trying to break into the MIS field mid-career. Anecdotally, there appears to be a bi-modal distribution of capability among students as well, with some outstanding students balanced by those struggling through their programs. Virtually all of the students are residents of the local area, and the majority tend to stay in the area after graduation. SPCS allows up to sixty transfer credits, and most students bring 45 – 60 credits into the program.

SPCS offers a Bachelor's of Science in Professional Studies (BSPS), with a major in Information Technology Management. A 21 credit hour undergraduate certificate in information systems is also offered for students who may already have bachelor's degrees and would like to change careers, or simply improve their MIS core competencies. The program is small, with 70 – 80 students typically enrolled at any given time. Classes are small, as sections are usually capped at 15 students.

It was felt that although the required courses address all the technical competencies expected of recent bachelor's students in MIS, a real-world course to strengthen and integrate core competencies would be beneficial to the student experience. The course would give students the tools to conduct a requirements analysis in an industry scenario immediately upon completion. A consulting section would also be added to the course to prepare students who desired to break into the field of MIS consulting. This course was developed and presented for the first time during the Spring of 2014.

Course Description

The overall purpose of this course is to help students develop necessary consulting skills, and to a comprehensive understanding of the design and use of online business information systems. The value of the course is to facilitate student understanding of consulting and development processes in a real-world environment. Although students are not anticipated to have developed all of the core competencies of the management information systems professional through previous courses (networking, systems analysis, database development and administration, application development, project management, etc.), students collaborate in teams, leading to a synergistic thoroughness and excellence in their final design document.

The course description states that the capstone course ties together all of the core competencies developed during the information technology management curriculum. The course introduces students to working as consultants in analysis, design and implementation of online and distributed management information systems (MIS). Topics include: consulting issues, organizational environments, requirements analysis, requirements tracking, preparation of functional specifications, modeling, database selection and design, man-machine dialog, reliability and response time requirements, security, user psychology, implementation planning, testing, and costing estimates. Students will complete an online system design document, and may develop a prototype online system. The analysis is conducted with teams of students working with a real world client. Although students may not have developed all core competencies of the MIS professional through previous elective courses (networking, systems analysis, database and application development and administration, project management), students learn from others in their team to understand the consulting and requirements collection process.

This course is a four credit-hour course, which meets for a 3 hour session once a week. Individual classes consist of a lecture/discussion, followed by a group exercise and any remaining time devoted to group collaboration. Students are required to collaborate on their projects outside of class, either in-person or electronically, although the exact means is left up to the students. Students are also expected to meet with their clients outside of class hours. The instructor is available to the students for assistance, either during office hours or electronically. There is also a reference librarian assigned to the course.

As with all group projects, management roles, collaboration and group dynamics play a large role in success or failure of teams. However, consideration of those variables is beyond the scope of this paper, with any problem issues being resolved by the individual groups with instructor assistance.

Course deliverables in addition to the final report include weekly updates to the customer, three interim deliverables comprising major parts of the final project, and several individual

deliverables such as project plan, scope statement, project charter, etc.

2. LITERATURE REVIEW

In a 1998 survey of 1250 IT users Kennedy finds that 78 percent of participants believe that fulltime study is the best way for undergraduate students to prepare for IS careers. However, only 20 percent of managers believe that graduates were prepared for the work. It remains well accepted that while the percentages may vary, the feeling that graduates from programs lacking integrative, real world experience persists. The course in the current work was designed to help students build confidence in their ability to complete a requirements document in the real world, integrating the competencies acquired throughout their programs.

It is equally well accepted that capstone projects universally add value to MIS programs. Dunlap (2005) suggests that capstone projects add value by introducing students to problem-based learning, and increases their self-efficacy at problem solving in a real-world environment. This increased self-efficacy motivates participants to attempt to solve problems they would not attempt otherwise. Others (Gupta & Wachter, 1998; Lesko, 2009; Brandon, Pruitt & Wade, 2002) suggest that a capstone course would help students assimilate and integrate the core competencies inherent in an information systems curriculum, particularly in the areas of technology and business, and to acquire practical knowledge. Cameron (2008) argues that the enterprise focus and integrative nature of capstone projects adds value and creates the kind of IT professional that is in demand today.

Gupta & Wachter (1998) further argue that such a course could be used to quickly and easily integrate emerging technologies. The authors additionally suggest several interesting delivery methods that may be incorporated into a capstone course. These include targeted assignments, case studies and Situation Analysis Reports (SAR). The targeted assignments are designed to lead students to think about specific scenarios, with a specific outcome in mind. Case studies help students understand the impact of technology on the enterprise. The SARs help students identify specific business problems and define how technology may be leveraged to solve the problems. In a 1998 paper, Novitzki discusses development of an integrating capstone course, and the problems encountered when initially presenting it. Solutions are suggested for the problems encountered, and the author finds that the capstone experience adds value to student programs, and gives them real world experience.

There is a body of literature discussing how best to present capstone courses. Lynch, Goold & Blain (2004) discuss student pedagogical preferences regarding how IT capstone courses are delivered. They suggest the delivery method affects the amount of control instructors have over the conduct of the course, based on the teaching model pedagogy. They discuss four models, industry sponsored, studio, traditional and directed. The industry sponsored model involves students playing the role of junior company employees, with the takes dependent on company needs. In the studio model, students receive value by collaborating with experts and mentors. In the traditional model, students collaborate in teams on internal or external projects, with a low level of interaction with faculty. Finally, in the directed model, students work with a technical and a managerial faculty member, on a clearly defined set of deliverables. In their study of 196 students from three institutions, studio, traditional and directed models were examined. The most significant finding is that students much prefer well-defined deliverables.

Sutcliffe, Chan and Nakayama (2005) describe how flexibility in emerging technologies and paradigms can be achieved through a modular, competency based approach. In a 2009 paper on using Wiki's to support the Net Generation in capstone projects, Ras and Rech find that the Net Generation, being digitally literate, differs from previous generations due to a paradigm shift in their use of technology. Since they have been exposed to technology all of their lives, they differ in the ways that they communicate and use knowledge and technology. The researchers use a Wiki-type system to help German software engineering students assimilate data in a controlled experiment during a capstone project. In a 2001 paper by Clear, Goldweber, Young, Leidig & Scott, the authors discuss issues that should be considered when developing and presenting capstone courses, with the intent of providing resources helpful to potential course instructors and designers.

Several relevant articles discuss prerequisites for the capstone. In a 2003 article on processes for capstone projects, Goold discusses the evolutionary changes to the computer science

capstone course at her institution. She suggests that two specific courses, an introductory software engineering course and a project management course are critical to the success of students in the capstone course due to their focus on process. It is important to ensure that both academic and industry requirements are met during conduct of the capstone. In a 2003 work, Bridgeman discusses design of a student project lifecycle that maximizes student value in the capstone project. This is achieved by developing a lifecycle that maximizes both industry-related as well as academic requirements. The article is very specific about technical and business requirements, and provides details about assignments and deliverables.

Many also suggest the importance of incorporating new and emerging technologies into capstone courses. Kumar's 2006 paper discusses strategies to quickly add emerging technologies to capstone courses to maintain their relevance. The author suggests that failure to incorporate these technologies makes the students less competitive in the marketplace. He suggests that while it is difficult to quickly change curricula, quick changes can be made to capstone courses, and he further suggests strategies to this end.

Capstone courses help improve student self-confidence. Dunlap (2005) suggests that capstone projects add value by introducing students to problem-based learning, and increases their self-efficacy at problem solving in a real-world environment. This increased self-efficacy motivates participants to attempt to solve problems they would not attempt otherwise.

However, not all students are pleased with the rigor and effort required for capstone courses, at least while they are participating in the course. Novitzki's 2001 paper discusses a management information systems capstone course for non-traditional graduate students. The paper describes development of a course that facilitates student integration of core competencies developed throughout their program. The author finds that participation in the course significantly improves the learning experience and that while students often dislike the course while they are taking it, when complete they typically find it to be one of the high points of their programs.

3. THE INITIAL COURSE

Students enrolling in the capstone/consulting course, titled "Consulting and Design of Online Systems," were required to override into the course. In effect, department permission was required, and potential students were reviewed to ensure they had completed a sufficient amount of coursework and/or had work experience relevant to the course content. This was particularly important for certificate students, since they often had significant experience in the field, yet had relatively few information systems course credits.

Students were expected to have familiarity with PCs operations, office systems, and the Web. The syllabus stated that the course would typically be taken in the final year of study. It was recommended that they had twelve credits in information systems, for bachelor's degree students, the mid-level composition course was suggested, since the deliverable included a detailed report. The course ended up with nine students enrolled, a good number given the small population of students in the information systems program, typically 70 – 80 bachelor's degree and certificate students at varying stages of program completion at any given time.

The first class meeting was a discussion of the course syllabus, course goals and objectives, and the challenges, benefits and general details of working as a consultant. Students were broken up into groups by the instructor, and were given time to get to know their teammates and discuss possible projects. It was suggested that students find their own projects, although the instructor would assist if they were unable to find suitable enterprises for their projects.

As the course progressed, it became apparent the students lacked the focus, network and insight required to select suitable projects for the course. Consequently, the instructor found two requirements analysis projects within the school that would meet the course requirements for scale, rigor and adherence to the course schedule.

A detailed description of the final requirements document deliverable was provided, divided into four sections – business case investigation, analysis, design and implementation. These followed a traditional systems analysis methodology. Each interim deliverable was designed to be incorporated into the final report. The reasoning behind assigning interim deliverables was that the instructor could

provide feedback, and the students would be kept from waiting to the last minute to complete their requirements documents.

Even with the instructor assigned projects, groups began to universally miss deliverable targets, and had a difficult time completing the tasks required to fill the requirements document template. There was a sense that the students were overwhelmed by the complexity of the tasks, and lacked the self-efficacy to attempt them.

With extended deadlines and assistance and prompting by the instructor, all groups were able to finally complete their requirements documents, and provide outstanding presentations at the end of the course.

4. DISCUSSION

Prerequisites
It was felt by faculty and program chair that many students were unprepared technically for the course. It is agreed within the department that additional prerequisites are required to ensure that students have a modicum of knowledge to feel comfortable and have the self-efficacy to add value in a group capstone project. It has been decided that in addition to the aforementioned prerequisites, hard prerequisites of either a systems analysis course or a project management course be required. Surprisingly, this action has been reached independently of that described by Goold (2003), where she suggests that two specific courses, an introductory software engineering course and a project management course are critical to the success of students in the computer science capstone course at her institution, due to their focus on process.

Although the project management as well as the systems analysis courses also focus on process, it is felt that they impart core competencies that are essential to the success of students in a capstone course. Although it is not felt necessary for students to have completed both courses, it is important that there is a sufficiency of students in each group to ensure that the core competencies are well represented. Implementing requirement of these additional prerequisites procedurally requires the approval of the school's Academic Council. It is anticipated that this approval will be granted during the Fall of 2014, well in advance of the course's next biennial offering in the Spring of 2016.

Preparation & Competency Acquisition
Since undergraduate students receive a broad exposure to the information systems field with less depth than master's degree students, it was found that there was great variance in the technical abilities of the students to complete various parts of the project. Master's students, when given a list of project deliverables for the final report and presentation, with interim deliverables throughout the course, were able to complete the tasks required by the list, since they had not only the breadth of education but also the depth in each subject area to give them the confidence to complete the tasks. Typical classes for the master's students consisted of an hour's worth of review on a specific topic, followed by group work where students develop that area of their project.

It was found during the initial offering that the bachelor's students often did not have the depth of knowledge or industry experience to complete the deliverables, even with the review of each topic provided by the instructor. One way to overcome this lack of competency is to use a sample project throughout the semester, where students participate in exercises developing each tenet of the report requirements. Following these exercises, they meet in their groups and perform the tasks on their own projects. In this manner, students acquire the requisite competencies through exercises, and then apply them to their own projects, reinforcing their ability to perform the required tasks. This methodology is anticipated to be followed during the next course offering.

Project Selection
The initial syllabus suggested that students independently select a real world enterprise to use as the subject of their analysis projects. In previous iterations where the author participated in capstone/consulting projects with master's degree students, students were expected to find their own subject enterprises. Although instructors typically had backup projects available for students groups who were unable to find projects of sufficient scope for their capstone, these projects were rarely, if ever required. However, several weeks into the current course, neither of the student groups was able to decide on potential projects. Since the course was on a strict timeline, the instructor provided separate real world projects within the university for the students to use.

In retrospect, there are several intuitively obvious reasons why the undergraduate

ISSN: 2167-1435
v31 n3073

students experienced difficulty in finding projects of sufficient scope and rigor. First, the master's degree students the author was experienced with had significantly more experience, and the related professional network, to find and exploit projects found through network contacts. Additionally, their experience gave them the self-efficacy to attempt the challenge of finding a suitable project.

In comparison, the undergraduate students were overwhelmed by the task of finding viable projects. While most of the undergraduate students were employed in the information systems field, they typically had lower level positions without the access and network to find suitable projects. Many of the undergraduate students had jobs ranging from help desk to call center to desktop support roles, positions not conducive to finding projects. Although many projects in the local area from non-profit organizations, businesses and government agencies were available, the students were unable to develop the necessary relationships to secure customers on their own. Consequently, it is agreed that in future offerings of the course, projects will be provided by the instructor to each group. Students may be asked if they have ideas or existing projects they may like to work on. This methodology will also allow the instructor more control on which tasks are required, and ensure that academic requirements for the course are met.

Pedagogical Delivery Method

As noted in the literature review, Lynch, Goold & Blain (2004) discuss four pedagogical delivery methods - industry sponsored, studio, traditional and directed. The initial course was presented using the traditional model, where students collaborate in teams on internal or external projects, with a low level of interaction with faculty. This was found to be difficult to implement and students were very dissatisfied and felt overwhelmed by the lack of support, scope and scale of the projects. This scenario was not found in the master's degree students. There are several reasons for this. Masters students have the experience and self-efficacy to attempt required tasks, even when they did not have specific experience with the tasks in question. Master's students also found it challenging to find solutions for problems they encountered. The undergraduate students were simply too frustrated and overwhelmed to attempt the tasks.

In the directed model, Lynch, Goold & Blain (2004) note that students work with a technical and a managerial faculty member, on a clearly defined set of deliverables. A significant finding is that students much prefer well-defined deliverables. It is suggested that a move from the traditional model to a directed model is advantageous for the undergraduate demographic. It is planned that a directed model with very specific deliverables be used during future course offerings. Interim deliverables on the subject matter covered during each lesson will be required. These interim deliverables would be chapters to eventually be submitted in the final report.

There are also other specific delivery methods that may be helpful. Students in the initial offering universally enjoyed exercises and assignments that allowed them to participate in diverse and interesting learning activities. Gupta and Wachter (1998) suggest several interesting delivery methods that may be incorporated into a capstone course. These include targeted assignments, case studies and Situation Analysis Reports (SAR). These additional methods are deemed to add value to the student experience, and it is anticipated that they will be incorporated into future course offerings.

Integration and relevance

One of the main tenets of a capstone course is how the course integrates the student experience in the information systems program (Lesko, 2008). This integration prepares students to use all of the core competencies acquired during the program in a way that enhances their abilities to conduct requirements analyses as they enter the workforce. Ideally, the experience prepares them to immediately conduct analyses upon employment. While these requirements were met, it is important to ensure that the integration of current core competencies remains an important part of planning for the next course offering.

It is accepted that one of the key requirements for a capstone course is that it remain relevant to the needs of industry, and to the needs of the prospective employers. When the course was offered, it was not known which enterprises would be used as projects. The syllabus was based on traditional system development methodology, with analysis, diagramming, interface, database and reliability and availability sections in the requirement document. As it

turned out, the actual course projects were all cloud-based solutions. The course requirements could not be completely and effectively applied given the shift from a client-server to a cloud based solution. Also, the number of students in each group, five for the initial course offering, is anticipated to be reduced to 3 for subsequent offerings due to the change in requirements due to the cloud based solution. The next offering of the course will have an updated syllabus including current and emerging trends in the requirements document.

Satisfaction

When offering a capstone course, students are thrown into the briar patch of a systems analysis project. They get little help from the instructor, who acts as a consultant, so students can be faced with the challenge of finding solutions to the various problems endemic to systems development work. Indeed, Novitzki (2001) finds that students often dislike the course while they are taking it, but when complete they typically find it to be one of the high points of their programs. This course was no different. One student stated at the end of the course that he hated the project his group worked on, and regretted taking the course while it was in progress. However, upon completion, he found it was worth the extra effort, and was glad it was on his transcript. The lesson is that instructors must have the expectation that student stress may be high, and instructor evaluations may suffer, as they are conducted before course completion. However, for instructors as well as students, there will be a sense of satisfaction upon course completion.

5. CONCLUSIONS

This paper adds to the body of knowledge by reporting on an effort to understand and improve a capstone course by incorporation of lessons learned from the first offering of the course and from the literature. In conclusion, this work defines numerous steps believed necessary to improve the student experience and outcomes of the capstone/consulting course as follows:

1. Additional prerequisites of either systems analysis or project management courses must be added to ensure that students have a modicum of expertise in the most important aspects of the capstone project.

2. Sample projects and exercises must be used throughout the course to makeup deficiencies in the depth or breadth of student competencies.

3. Future course projects will be assigned by the instructor to facilitate completion of the course and better control conduct of the course analysis.

4. A directed pedagogical delivery method must be used to ensure that students are able to easily meet course requirements, and conform to the well-defined final report requirements.

5. Other delivery methods, such as targeted assignments, exercises and case studies must be incorporated to improve the student experience and impart knowledge on how to complete required tasks.

6. Future syllabi must be reviewed when the course is offered to ensure that the content remains relevant to industry, and that new and emerging technologies are incorporated. The number of students in each group will be reduced from five to 3.

7. Finally, such a real world endeavor as a capstone course is anticipated to be stressful to students as well as instructors, but the end result will be a beneficial experience that will be useful, relevant and immediately applicable.

6. REFERENCES

Brandon, D., Pruett, J., & Wade, J. (2002). Experiences in developing and implementing a capstone course in information technology management. Journal of Information Technology Education: Research, 1(1), 91-102.

Bridgeman, N. (2003). Project success: defining, designing, constructing & presenting a capstone project. 16th Annual NACCQ, Palmerston North, NACCQ, 211-216.

Cameron, B. H. (2008, April). Enterprise systems education: new directions & challenges for the future. In Proceedings of the 2008 ACM SIGMIS CPR conference on Computer personnel doctoral consortium and research (pp. 119-126). ACM.

Clear, T., Goldweber, M., Young, F. H., Leidig, P. M., & Scott, K. (2001, December). Resources for instructors of capstone courses in computing. In Working group

reports from ITiCSE on Innovation and technology in computer science education (pp. 93-113). ACM.

Dunlap, J. C. (2005). Problem-based learning and self-efficacy: How a capstone course prepares students for a profession. Educational Technology Research and Development, 53(1), 65-83.

Goold, A. (2003). Providing process for projects in capstone courses. ACM SIGCSE Bulletin, 35(3), 26-29.

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.

Kennedy, S., "Full-time Education to Overcome Skills Gap," ComputerWeekly.com, February 5, 1998.

Kumar, A. (2006). Strategies to enhance student learning in a capstone MIS course. Issues in Informing Science and Information Technology, 3, 327-332.

Lesko Jr, C. J. (2009, October). Building a framework for the senior capstone experience in an information computer technology program. In Proceedings of the 10th ACM conference on SIG-information technology education (pp. 245-251). ACM.

Lynch, K., Goold, A., & Blain, J. (2004, January). Studentspedagogical preferences in the delivery of IT capstone courses. In InSITE 2004 Informing Science and IT Education Joint Conference (pp. 431-442). Informing Science Institute.

Morgan, R. E., & Aitken, R. (2006). The Business of Information Technology: An Integrated, Multi-disciplinary Approach to a Capstone Experience for Management Information Sciences Students. Director, 07.

Novitzki, J. E. (2001). Critical Issues in the Administration of an Integrating Capstone Course. Proceedings Informing Science and Information Technology Education 2001, 372-378.

Novitzki, J. E. (1998). The MIS Capstone: Development on an Integrating Group Applied Project Course. In International Academy for Information Management Annual Conference (Vol. 1998, No. 1).

Ras, E., & Rech, J. (2009). Using Wikis to support the Net Generation in improving knowledge acquisition in capstone projects. Journal of Systems and Software, 82(4), 553-562.

Sutcliffe, N., Chan, S. S., & Nakayama, M. (2005). Competency Based MSIS Curriculum. Journal of Information Systems Education, 16(3), 301.

ISSN: 2167-1435
v31 n3073

Appendix A: Project Outline

(Minimum Document Content Requirements)

Phase 1 Investigation – Business Case January-February

1. Introduction

    1.1 System Introduction
    1.2 Project Charter

2. Business Case

    2.1 Organization Description
        2.1.1 Environment
        2.1.2 Business Goals
        2.1.3 Competitive Position & Benchmark
        2.1.4 Summary: Need for the System

    2.2 System Description
        2.2.1 Business or Administrative Problem Addressed
        2.2.2 System Context Model (DFD)
        2.2.3 Business Assumptions and Risks
        2.2.4 System Cost Targets & RIO Estimates

    Deliverables Due: Approximately 2/12/2014

Phase 2 – Analysis February-March

3. Requirements Specifications

    3.1 Functional Capabilities
    3.2 System Capacity & Performance
        3.2.1 Storage Needs
        3.2.2 Numbers & Types of Users
        3.3.3 Transactions (time to complete, and transaction numbers)

4. Models and Analyses

    4.1 Use Cases
    4.2 E-R Diagrams

4.3 Data Dictionary
4.4 Interfaces (Internal and External)
4.5 Maintenance Requirements

5. System Configuration

5.1 Architectural Design
5.2 Hardware & Security Configuration
5.3 System & Security Software Configuration
5.4 Languages/Tools (development and operations)

6. Screens and Interface Prototype

6.1 CONOPS
6.2 Screens and Screen Flow
6.3 Business Rules (for Screens/Fields)

Deliverables Due: Approximately 3/19/2014

Phase 3 Design Feb-March

7. Database Design

7.1 Database Software
7.2 Database Schema and Description
7.3 Test and Demonstration Data

8. Detailed Software Design

8.1 Modules & Programs
8.2 Internal Logic & Structure Charts for Key Programs
8.3 Security

9. Development Plan

9.1 Development Stages
9.2 Project Work Plan (MS Project)
9.3 Risks & Impact Assessment
9.4 Development Testing

Deliverables Due: Approximately 4/9/2014

Phase 4 – Implementation April

10. Support & Training

ISSN: 2167-1435
v31 n3073

10.1 User Training Plan
10.2 Help Functions (tier 3 support)
10.3 Installation & Turn-on
10.4 Ongoing Maintenance/Support

11. Code - Developed Software

12. User Test Plan

12.1 Implementation Stages & Cut-over method
12.2 User Test Plan & Scenarios

13. Near and Long Term Emerging Technologies

13.1 The Technologies (1 near; 1 long term)
13.2 Market Drivers & Benefits
13.3 Feasibility

14. Research – Marketing Paper

14.1 Target Journal Rationale
14.2 Journal Format & Contents
14.3 Article (hard and soft)

Revised Chapters & Final Deliverables (2 Hard & 1 soft)
Due: Capstone Presentation 4/30/2014

(With Capstone Presentation)

Personal Reflections and Critical Reviews

Peer Review Form