

DESIGN, IMPLEMENTATION, AND ASSESSMENT OF AN INFORMATION TECHNOLOGY SENIOR CAPSTONE COURSE

Alex Koohang, Macon State College, alex.koohang@maconstate.edu
Kevin Floyd, Macon State College, kevin.floyd@maconstate.edu
Richard Spiers, Macon State College, richard.spiers@maconstate.edu
Liz Riley, Macon State College, liz.riley@maconstate.edu

ABSTRACT

This paper describes the design and implementation of an IT senior capstone course. The design takes into account the inclusion of various unique characteristics of a capstone course reported in the literature. The structure of the course is explained and the unique characteristics are emphasized within the design. In addition, the design gives attention to the use of the capstone course as a means for overall program assessment and evaluation.

Keywords: Capstone course, Information Technology, accreditation, ABET

INTRODUCTION

Capstone courses are commonly a part of the core requirements in an academic major. A capstone course is typically one academic semester or term and includes the completion and presentation of a major project. A capstone course requires students to summarize, evaluate, and integrate their skills, knowledge and experience they have accumulated throughout their program of study (Henscheid & Barnicoat, n.d.).

In a comprehensive research study, the following characteristics of senior capstone courses in general have been summarized:

1. “promotion of the coherence and relevance of general education;
2. promotion of integration and connections between general education and the academic major;
3. fostering of integration and synthesis within the academic major;
4. promotion of meaningful connections between the academic major and work and career experiences;
5. explicit and intentional development of important student skills, competencies, and

- perspectives that are tacitly or incidentally developed in the college curriculum;
6. enhanced awareness of and support for the key personal adjustments encountered by seniors during their transition from college to post-college life;
7. improvement of seniors' career preparation and pre-professional development;
8. enhancement of seniors' preparation and prospects for postgraduate education;
9. promotion of effective life planning and decision making with respect to practical issues likely to be encountered in adult life after college” [3].

Fanter [5] asserts that a capstone course provides students the opportunity to integrate their knowledge and skills. The author lists four different types of capstone courses. They are the following: field or internship programs, the portfolio-building capstone course, the multiple-project course, or a major project course. Fanter [5] contends that capstone courses can be challenging, however, they can be a rewarding learning experience for students and contribute to their preparation for life beyond college.

DESIGN OF CAPSTONE COURSES WITHIN IT

Jin [8] asserts that a senior capstone in computer science is an exit course. It should include a hands-on project that incorporates students' previous knowledge within their academic program. In addition, the capstone includes a term paper that assesses students' oral and written communication abilities. The aim of the capstone course is to improve students' learning outcomes and prepare them for graduate schools and/or entry into workplace [8].

Bruhn & Camp [2] believe that their engineering capstone course creates useful business products and corporate-ready students. Students work in teams to complete a real-world project. The goals of this capstone course include the following: 1) encouraging students to acquire professional skills, 2) providing the industry with a product completed by

students, and 3) assessing and evaluating the curriculum.

Jones [9] postulates that the design of the capstone in a software development program should focus on integration and application of concepts and skills students have learned throughout their program of study. The capstone should be a means to careers as systems analysts and software engineers.

The literature notes that a successful senior capstone in computing should allow students to experience real-world projects from analysis phase to implementation and delivery of the information system [1].

In redesigning the curriculum's capstone course, Gould [5] believes that the course must include real-world or experiential learning projects. In addition, the project should involve real clients or sponsors with larger teams that includes students with diverse knowledge and skills. The author further believes that the course must focus on project management in order to provide a more structured process of product design, development, and implementation [5].

Others also assert that a capstone course should include experiential, i.e., real-world experience and conceptual elements [9]. In addition, a capstone course in IS should emphasize interpersonal communication, teamwork, and project management elements [9].

PURPOSE OF THE STUDY

The purpose of this paper is to describe the design and implementation of an IT senior capstone course. The design includes the unique characteristics of a capstone course reported in the literature. In addition, the course is designed to be a part of the overall IT program/curriculum assessment.

This paper is organized in a manner consistent with its purpose. Introductory remarks include a review of literature about senior capstone courses. Unique characteristics of senior capstone courses are defined in general and within the field of IT. Secondly, the discussion turns to the design and implementation of an IT senior capstone course focusing on program accreditation and how the capstone course is used as a means of assessment. Thirdly, the structure of the senior capstone course is described. Next, the self-assessment is discussed. To show how the self-assessment is used to assess the program outcome, an informal example of an analysis from a set of data collected from students taking the senior capstone is

presented. Conclusions and recommendations for future research round out the paper.

THE DESIGN OF THE IT SENIOR CAPSTONE

The IT Program Accreditation

The School of IT at Macon State College is currently seeking accreditation for its undergraduate Information Technology program. This accreditation is being sought from the Accreditation Board of Engineering and Technology (ABET). ABET accredits academic programs that prepare graduates for entry into four professional disciplines: applied science, computing, engineering, and technology. (See <http://www.abet.org>).

The process of ABET accreditation requires documentation of three elements: 1) program educational objectives, 2) program outcomes, and 3) program continuous improvement. Program educational objectives describe what the program is preparing graduates to achieve once they have completed the program and are established in their profession. Program outcomes describe skills or knowledge that students should have acquired through the completion of the program (page 8 of the 2008-2009 Accreditation Policy and Procedure Manual, retrieved from <http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/A004%2008-09%20Accreditation%20Policy%20and%20Procedure%20Manual%2011-8-07.pdf> Both program educational objectives and program outcomes must be "...based on the needs of the program's constituencies".

In addition to establishing program educational objectives and program outcomes, accreditation also requires a process for assessing the achievement of these outcomes and continually improving the program as a result of these assessments. Documentation of the assessment and improvements must be maintained.

The Program Outcomes

The Macon State's School of IT curriculum has identified 12 program outcomes. The program outcomes are as follows:

1. Demonstrate expertise in the core information technologies.
2. Demonstrate the ability to analyze, identify, and define information system requirements.

3. Design and implement effective and usable IT-based solutions in a user environment.
4. Use appropriate project management methods in the creation of an effective IT project plan.
5. Identify IT methods used in protecting the confidentiality, integrity, and availability of information and its delivery systems.
6. Identify and apply current and emerging technologies.
7. Apply relevant ethical, legal, security, and policy principles in technology environment.
8. Describe and apply best practices and standards in IT applications.
9. Demonstrate independent critical thinking and problem solving skills.
10. Work effectively in teams to develop IT based solutions.
11. Communicate effectively.
12. Recognize the need for lifelong professional development and learning.

Program outcomes represent the knowledge and skills that students should acquire as a result of successfully completing all courses in the program. These program outcomes are supported by multiple courses in the School's curriculum. Special attention to the assessment of these outcomes has been given in the core/required IT courses.

DEFINING THE IT SENIOR CAPSTONE

Taking into consideration the characteristics of the senior capstone courses as reported in the literature, it was determined that the senior capstone course at Macon State's School of IT would require students to analyze, design, develop, implement, and assess a real-world functional information system. Secondly, the senior capstone would be taken as an exit course for all IT majors. Lastly, students, working in teams, would integrate their skills and knowledge acquired throughout their four-year IT program in order to produce the functional information system. The skills and knowledge were obtained from the following core IT courses: programming, Web development, data driven Web development, digital media, networking, systems design, HCI, databases, decision support/intelligence system, project management, IT security, and IT integration. Additional upper level elective IT courses chosen by the students would further enhance their knowledge in certain areas such as networking, information security, digital media, and software development.

Once the Senior Capstone course was established as an exit course, it was also identified as a good

candidate for the evaluation of the achievement of program outcomes. This evaluation or assessment was taken into account as the course was designed. Students would be required to write a lengthy paper in which he/she would discuss the importance of the program outcomes and how well he/she achieved each of the program outcomes. This paper would provide an indirect assessment for evaluating the assessment of the program outcomes.

Given the possible variety of information systems that the students would be building, it was also decided that the course would be team-taught by 2 or 3 professors.

STRUCTURING THE IT SENIOR CAPSTONE

The structure of the course was determined to include three parts: information system analysis; information system design and development; and information system implementation and evaluation.

Part 1 – Information System Analysis

Part 1 includes the information system analysis and project management tasks necessary for completion of the system. In this part of the project the teams are formed. Each team will have a project leader and, in some cases, a team may decide to have a co-project leader. Once a team is formed, the following tasks will be assigned and completed by the team:

Task 1 - Identify a real-world project. A list of pre-determined real-world projects is made available to students. In a normal semester, the list may include three different projects. This list is compiled from various companies by the professors facilitating the course. This saves a tremendous amount of time and aggravation on the students' part. Each project will have a designated sponsor representing the company for which the information system is being designed and implemented. The sponsor is available to answer questions and provide feedback throughout the analysis, design, implementation, and evaluation of the system. (See Table 1 for an example of a real-world project)

Task 2 – Determine System feasibility. Once the team chooses a real-world project from the list, the process of investigation and fact gathering begins. The team then analyzes system feasibility including operational, scheduling, economic, and technical details. This analysis is documented and submitted to the professors by the end of the second week in the semester.

Task 3 – Establish Project Management Documentation. Next, the team will compile a project scope statement, project charter, communication plan, team contract, Work Breakdown Structure (WBS), and a Gantt chart. The team is given one week to complete and submit this part to the professors. The WBS and Gantt chart are continuously updated throughout the semester.

Table 1: An Example of a Real-world Project

Project Description: Redesign and convert the current XYZ Company to a completely dynamic data-driven Web site. The site should be designed and developed to efficiently run in a LAMP (Linux/Apache/MySQL/PHP) environment.

Project Design Specifications: All pages should conform to XHTML 1.1 standards. Page layout should be based on valid CSS standards. The site should also include a consistent navigation system. Pages should be thoroughly tested for standard accessibility and usability issues.

Project Development Specifications: PHP 5.0 should be used to create the dynamic content associated with the information system. The back-end of the system will be a MySQL relational database management system. All database tables should be normalized to at least third normal form.

Hardware/Software Required: MySQL Database Management System, Text Editor, web server that supports PHP/MySQL, and secure FTP, IE/Firefox, MS Project, and appropriate accessibility/usability tools.

Technologies Required: PHP 5.0, XHTML 1.1, CSS, and SQL

Part 2 – Information System Design and Development

Once Part 1 is approved by the professors; the team will begin the design and development of the information system. In addition, the team is required to document the following throughout the design and development phase:

- All necessary outputs, inputs, and procedures for the information system
- All scripts and programs
- Database(s)
- Multimedia systems used for the information system

- Human computer interaction design (user interface/usability) for the information system
- Network infrastructure
- Information system integration processes
- Information systems reliability, accuracy, maintainability, and security

This part is submitted to the professors a week before the implementation of the information system. Normally, Part 2 takes about 9 to 10 weeks.

Part 3 – Information System Implementation and Evaluation

In Part 3, the team will begin the implementation of the information system. Once the system implementation is complete, the team will begin the evaluation of the system. The system will be checked for functionality, reliability, accuracy, and security. In addition the team will be required to document the evaluation method for the information system. The team will also outline and provide recommendations for maintainability of the information system.

Once part 3 is completed, the team will make a comprehensive oral presentation demonstrating the functionality of the information system. This presentation will include an overview of the tasks involved in the analysis, implementation, and evaluation. Part 3 is normally completed in 3 to 4 weeks.

THE SELF-ASSESSMENT & PROGRAM OUTCOMES ASSESSMENT

The self-assessment included in the senior capstone is used for indirectly assessing program outcomes. The self assessment has 2 parts. In part 1, each student is asked to create meaningful statements of his/her learning. These statements should focus on how he/she has developed as a learner/thinker; what skills, knowledge, and experiences he/she has learned in the IT program; and, how integrative his/her experience has been. This information is summarized each semester and is taken into account when program outcomes are reviewed. This information is used for continuous improvement of the program.

In part 2, students are asked to evaluate the program outcomes in the following manner: 1) how well they achieved each of the program outcomes and 2) how important each program outcome is in relation to their learning.

As an example, Table 2 shows the result of the analysis of a set of data from part 2 of the self-assessment that was collected from 24 students taking the capstone course.

Table 2: Results of the Program Outcome Evaluation

| Program Outcomes | I | A |
|---|----------|----------|
| 1. Demonstrate expertise in the core information technologies. | 4.86 | 4.76 |
| 2. Demonstrate the ability to analyze, identify, and define information system requirements. | 4.24 | 4.05 |
| 3. Design and implement effective and usable IT-based solutions in a user environment. | 4.81 | 4.76 |
| 4. Use appropriate project management methods in the creation of an effective IT project plan. | 4.19 | 4.05 |
| 5. Identify IT methods used in protecting the confidentiality, integrity, and availability of information and its delivery systems. | 4.86 | 4.76 |
| 6. Identify and apply current and emerging technologies. | 4.10 | 4.05 |
| 7. Apply relevant ethical, legal, security, and policy principles in technology environment. | 4.67 | 4.76 |
| 8. Describe and apply best practices and standards in IT applications. | 3.71 | 4.05 |
| 9. Demonstrate independent critical thinking and problem solving skills. | 4.86 | 4.76 |
| 10. Work effectively in teams to develop IT based solutions. | 4.10 | 4.33 |
| 11. Communicate effectively. | 4.62 | 4.86 |
| 12. Recognize the need for lifelong professional development and learning. | 3.90 | 4.43 |

I: Important, A: Achieved

The scoring strategy for the importance of each outcome include 5=very, 4=somewhat, 3=neutral, 2=not, and 1=not at all. The scoring strategy for the extent to which each student has acquired the skill or knowledge is 5=completely, 4=mostly, 3=somewhat, 2=a little, and 1=not at all.

The information above is only for the purpose of sharing with readers how information is collected and summarized each semester from senior capstone courses. This information is then taken into account when program outcomes are reviewed. This will allow for continuous improvement of the program.

CONCLUSION

The paper described the design and implementation of an IT senior capstone course. The design took into account the various unique characteristics of a capstone course reported in the literature and how the course can be used as a means to evaluate and assess the IT program outcome. Accordingly, the following characteristics were considered in the design of the senior capstone:

- The senior capstone course is an exit course.
- Students work on real-world projects. These projects are hands-on and experiential. They involve real clients or sponsors.
- The senior capstone requires teamwork.
- The senior capstone includes students with a diverse skills, knowledge, and experience.
- Students integrate their accumulated skills, knowledge, and experience throughout the program to analyze, design, develop, implement, and evaluate an information system.
- Project management is used to provide a structured process of design, development, and implementation of the information system.
- Oral and written communications are assessed throughout the course.
- Senior capstone requires a self-assessment exercise.
- The senior capstone is used to assess/evaluate the curriculum.

The structure of the course included three parts: information system analysis; information system design and development; and information system implementation and evaluation. All unique characteristics of a capstone course reported in the literature were taken into account in the design of our senior capstone course. This included a self-assessment that was used for overall program evaluation and assessment.

Although we have had remarkable success with the delivery of this course, it is our intention to empirically examine the efficacy of this design in the near future. We believe that the empirical

examination will enhance the design of our capstone course.

REFERENCES

1. Beasley, R. (2003). Conducting a successful senior capstone course in computing. *Journal of Computing Sciences in Colleges*, 19(1), pp. 22 – 131.
2. Bruhn, R. & Camp, J. (2004). Capstone course creates useful business products and corporate-ready students. *ACM SIGCSE Bulletin*, 36(2), pp. 87 – 92.
3. Cuseo, Joseph B. 1998. Objectives and benefits of senior year programs. In *The Senior Year Experience: Facilitating Reflection, Integration, Closure and Transition*, ed. John N. Gardner, Gretchen Van der Veer, and Associates. San Francisco: Jossey-Bass.
4. Fanter, A. (2006). Preparing for post-college life: capstone and keystone courses. Retrieved February 1, 2009 from <http://www.worldwidelearn.com/education-articles/capstone-keystone-courses.htm>
5. Goold, A. (2003). Providing process for projects in capstone courses. *Proceedings of the 8th annual conference on Innovation and technology in computer science education*, pp. 26 – 29.
6. Henscheid, M. & Barnicoat, L. (n.d.). Capstone courses in higher education. Retrieved February 1, 2009 from <http://www.answers.com/topic/capstone-courses-in-higher-education>
7. Jin, M. (2008). Redesign of the computer science capstone course by integrating the major field test (MFT). *Journal of Computing Sciences in Colleges*, 24(1), pp. 239-246.
8. Jones, E. (2002). Testing in the capstone course - reusable patterns for a value-added experience. *Journal of Computing Sciences in Colleges*, 17(4), pp. 142 – 144.
9. McGann, S. & Cahill, M. (2005). Pulling it all together: An IS course for the 21st century emphasizing experiential & conceptual aspects, soft skills and career readiness. *Issues in Information Systems* 6(1). Pp. 391-397.