Capstone Projects:
Putting the Pieces Together

Christopher Malinowski
cmalinow@liu.edu

Richard E. Noble
richard.noble@liu.edu
Computer Science Department, CW Post, Long Island University
Brookville, N.Y. 11548, United States

Abstract

Three exemplar senior, or capstone, projects are suggested as a method of incorporating the body of knowledge and skills acquired by undergraduate students as these skills may be implemented by entry level professionals. Additionally these projects introduce students to research and help prepare them for entry into today's workforce.

Keywords: capstone project, information systems project, student employment, student research, training

1. INTRODUCTION

Employers hiring new graduates into entry level positions often complain about the degree of unpreparedness they find among newly hired students. These areas include both our students' technical as well as their social or "soft skills". As instructors we therefore strive to construct a curriculum incorporating the skills, knowledge and abilities required for students to cope in the workplace when they leave our institutions.

Sadly, students have difficulties meeting many of their employers' expectations. These unsatisfied expectations however, often do not fall directly within the subject matter areas of our class material. Employers often understand that students may not be well versed in a particular language or technology as utilized within their organizations, and expect to train them as a routine matter of course. This training generally extends both to develop the technological skills as well as to socialize employees within their organization. What employers do expect is that their employees can function within a team environment, and that they are capable of communicating and extending their body of knowledge. According to interviews with IS managers, students are also lacking in such areas as business operations, knowledge of a specific industry and "soft skills" (Wilkins and Noll, 2000.)

Lee, Trauth & Farwell (1995) stated that industry expects professionals to have technological skills as well as knowledge of business operations, [project] management and interpersonal skills. They also suggest that the "generic" IS curriculum is obsolete, and that curricula should be developed in order to accommodate different IS career choices. In other words we should focus more narrowly and develop knowledge and skills within specific areas of a discipline, such as database or networking, similar to specializations in other professions. This suggestion is reiterated by Trauth and Hafner (2000.) While some may suggest that these citations themselves are obsolete, many of us who maintain professional contacts would probably still agree to their validity.

Professors' complaints about students typically include students' poor time management skills and their dedication to assignments. It is felt by many that students fail to identify and concentrate on project details, do not demonstrate critical thinking, and often fail to pursue their assignments past the bare minimum specifications despite an impending threat of poor grades. In fact, many students apparently do not internalize the need for any work regimen or discipline in a given curriculum.

Methods of motivating students seem to elude us at every turn; certainly negative motivators such as poor grades apparently do not work for many students. One positive reinforcement and motivator is the ability to produce. It is widely held that a sense of accomplishment that is associated with the successful completion of any assignment or project is a strong positive motivator. The key is to ensure such a successful project completion and to provide a transition for the student who is about to enter a professional work environment in which the expectations of the student may be radically different than the reality he or she is about to face.

2. OBJECTIVES

C.W. Post offers three undergraduate degrees: Computer Science, Information Systems and Information Management Technology. We currently offer a 'concentration' in networking and in the fall of 2007, we intend to offer an undergraduate concentration in Computer and Network Forensics (a concentration consists of five elective courses in a subject area). Our computer language de jour is VBA for the instruction of our beginning students. Later we differentiate by instructing with Java for IS and C++ for the Computer Science students. Perl is being introduced to those students whose concerns will deal with information administration (Information Management Technology majors).

In order to address the suggested specializations of curricula, we have built two concentrations into our programs. Our undergraduate network offering consisted of a single course, until we recently implemented the network concentration. These additional courses build on a student's knowledge in such areas as Network Administration, Network Security, Network Coding, Intro to Forensics /Incident Response and a Special Topics course. The planned Computer Forensics concentration, also consisting of five courses, will focus on subject matter pertaining to Information Assurance.

With this in mind, we suggest a set of senior or capstone projects which would be appropriate for each major discipline. The goals are to have students demonstrate and integrate the concepts and skills acquired during their undergraduate coursework. The objective of the project is to meld a student's knowledge into a cohesive "real world" project. Due to time constraints however, only selected components of this project might be applicable, allowing students to complete their assignments. This condensed project will help ensure that students experience as wide a range of project and coding skills as possible in the available time.

Soft skills such as documentation, communication, research, business practices (project management issues such as time and cost), and presentation skills need to be incorporated in any project. Therefore, as part of this project, students will be required to perform those soft skills which, while not deemed as essential as their coding effort, are expected of professionals by their employers. Insofar as research indicates an expectation of a team effort in the business environment (Earl, 1995; Todd, McKeen & Gallupe, 1995), students will complete these projects in teams, building their project to specifications.

Studies have indicated that early working experiences of IS professionals can shape their work experiences for many years to come (Thompson, et al., 1974; Kaufman, 1974; Katz and Tushman, 1983.) One benefit of a carefully crafted capstone course is in providing this work experience while still in an academic setting, allowing students to experience project demands without the accompanying corporate sanctions for simply being inexperienced.

In attempting to address both the knowledge areas which students should possess, as well as the concerns of their prospective employers, our approach is to develop capstone courses which not only depend on those technical bodies of knowledge but also depend on the abilities of students to function in compartmentalized teams. These requirements force the students to interact in such a fashion as to ensure that they communicate effectively in order to complete their projects.

This approach is taken from our graduate Information Systems curriculum in which the students are formed into teams or groups. All groups are asked to analyze and develop a design for a specific scenario. Upon completion of the design, each group's design is provided to another team in the class. In turn, that group will write code matching specifications provided by the prior team's design. Testing of this code will be turned over to the next group in turn.

By dividing an undergraduate class into three groups, each group will perform discrete IS project related tasks. As these tasks are based on another group's work, they can be performed in an 'isolated' fashion; that is to say, the work approaches a real working environment in which teams are constrained to work with information and work product supplied to them. Each team will be required to produce the end result of their phase within a definitive time frame (in order to provide the next team with work product).

Selection of projects should ensure the inclusion of these characteristics, as well as allow for project completion within the timeframe allotted. Should a 'maintenance phase' not be feasible due to time constraints, a prepared maintenance coding exercise (not based on the project problem scenario) can be assigned, maintaining the philosophy of the capstone project.

The initial role of the instructor should be to portray a client, and respond to students' questions during the analysis stage of the project. At some point, the instructor's role will eventually shift to that of a project manager.

While studies have indicated that employers expect workers to have project management knowledge, a capstone course conducted in a semester's timeframe may not allow for internalization of project management principles while completing the project. Rather, vicarious lessons may be given by allowing the students to observe the "project manager" (instructor) at work. Ideally students should have been exposed to some basic project management principles during a software engineering course allowing the capstone course to reinforce these principles.

3. PROJECT DESCRIPTIONS

In our particular case, as we have three different undergraduate degree programs, we can develop three separate capstone projects focusing on technical skills required by each discipline. These are the Computer Science, Information Management and Technology, and the Information Systems programs.

Two of the three programs would have capstone projects involving a coding effort, for which the three or four team methodology described later may be appropriate. The Information Management project, while not heavily involved in any coding effort, may also employ teams in order to benefit from a real-world environment. While the focus for these students would not involve coding, it would require that they function in teams and develop those soft skills as expected by prospective employers.

A three team methodology (Figure 1) can be employed by both the Computer Science as well as the Information System students in order to develop an application, allowing a class to be divided into three functional groups or teams (A, B, and C). Each group is expected to develop a specification document based on a client's needs. In this case the instructor initially serves as the client. The 'client' will be available for assessing project needs in order to determine what information and which resources are currently available for the project.

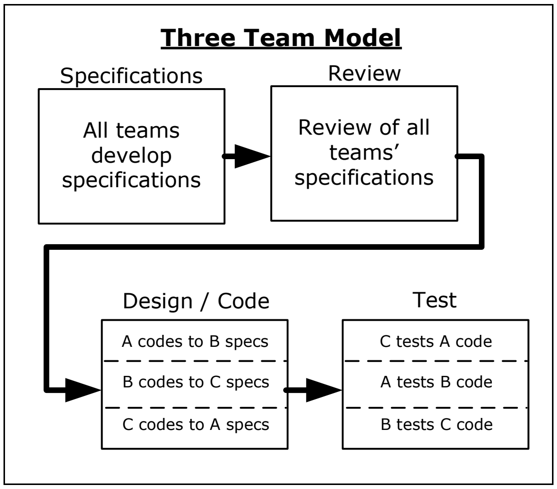

Figure 1

At some point in time, the instructor will change his/her role to that of a project manager (or provide a separate PM). As the project is being managed students will be educated as to the basic project management principles being implemented. While it might be desirable to have a student in each group function as a project manager, the overall requirement to keep the project on track is driving the project. By allowing the instructor to drive the project, the likelihood of a successful completion is significantly increased.

When the teams have completed their specifications, these documents will be reviewed, and these specifications will be delivered to the next student team for coding. For purposes of ensuring consistency and uniformity of the project design, the instructor may very well provide the students (working in groups) with the requirements specifications as depicted in Figure 2, rather than allow each team to code to different specifications.

This may have the additional benefit of compressing the time required in order to review and correct individual group designs. All subsequent coding efforts by each group will thereby be based on a consistent design document.

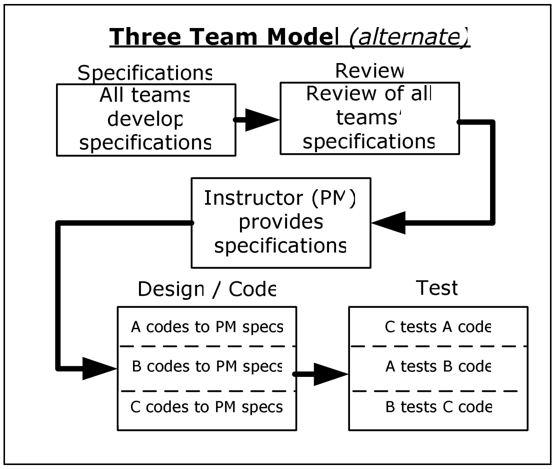

Figure 2

Coding will be performed by each team to these specifications, and this code will be subjected to a review for compliance to the specifications. Each team will then separately develop test criteria for their version of the application. At this point, it is expected that each team will develop their own unique solution to the project, notwithstanding the identical specifications issued to them. The code, along with the test criteria developed by the team, will then be delivered to the next team for testing purposes.

During the course of the project it is expected that wherever appropriate, all documentation will be prepared and presented as part of the 'normal' systems development life cycle.

An additional aspect of the 'real-world' project is to introduce maintenance coding. Studies suggest that approximately 70% of all coding effort and cost is expended on the maintenance phase of the system (McNurlin and Sprague, 2002). A modified version of the three team methodology allows for a maintenance phase of the code after the testing phase by adding a fourth group.

In reality, maintenance on the students' own project code may not be feasible in the amount of time allotted; therefore the instructor may be required to provide his own application code with modification specifications to be implemented by each team.

Empirical studies indicate that students enter the workforce with expectations of acquiring knowledge in the same manner as in the educational setting. Structured and organized classes and the proactive transfer of information from an instructor to a student do not occur in a professional setting. As professionals they will need to adjust to finding different methods of acquiring information. Our approach is to deliver the content of the capstone project to the students in a professional manner requiring them to adapt to a new knowledge acquisition paradigm.

Lee (1986, 1992, 1994) conducted longitudinal studies finding no correlation between job performance and academic achievement. Another study has shown that in the workplace, job performance is closely related to consultation with coworkers (Allen, 1977; Pelz and Andrew, 1976; Goldhar et al., 1976). With this in mind, a corresponding method of knowledge transfer should be available to the students, either in the form of the instructor as a 'technical mentor' or in providing a class assistant/tutor as a 'colleague' who is available for consultation.

Computer Science Project

Based on a growing need for Information Assurance (Information Security Specialists) (Wilkins and Noll, 2000), this project is designed to utilize the knowledge and coding skills required in order to retrieve and manipulate data from systems. While security specialists utilize software tools and suites to perform their tasks, they generally do not have the wherewithal to build such tools. The Computer Science students will focus on those areas of the OS and devices (including volatile as well as durable memory structures) in order to develop an application based on their knowledge of appropriate system structures.

Students of Computer Science will be given specifications for developing solutions in order to obtain digital data from devices. The focus of the coding effort is to have students research the appropriate topics which may not be encompassed within one 'volume of knowledge', develop technical specifications for meeting their objectives, code, test and debug their code. Lastly, the students must document and present their efforts.

Requirements for this project will be to duplicate a set of functions which are built into existing computer forensic applications, such as EnCase, FTK, or SMART. Demonstration of such a tool will provide visual cues for students, both as to the data results as well as the presentation of results to the end users. Requirements for this specific project are not so numerous as to be overwhelming, and in fact instructors may wish to conduct an abbreviated analysis stage allowing students to develop specifications. The initial demonstration of existing software tools or applications will assist in reducing the amount of time required in order to develop the specifications. Additionally, the project will be restricted to a selected subset of functions allowing students to complete the project within the given timeframe.

Examples of functions to be implemented might include:

- acquisition of data from various devices
- verification of data acquisition
- archiving of data (optional compression)
- examination of file structures
- examination of memory structures (processes, etc)
- rudimentary analysis of file structures (MAC times, sizes, comparison to known file list, etc)
- string searches
- file anomaly detection

For the computer science students, the "three team" or "four team" methodology will be applied. Maintenance coding can be performed on existing applications or components in order to enhance or correct current code; for example students can be asked to identify and correct situations such as TCP session hijacks in a connection request or to identify and correct buffer overflow conditions in networked applications.

Information Management and Technology Project

The Information Management project focuses on industry needs for those professionals required to interact with staffs of technicians and management, as well as provide a response to growing needs for Information Assurance staff. While these specialists have little or no requirements for application or systems coding per se, they do require special skills in order to manipulate information and manage this knowledge for organizations.

Incident Response Project

A security incident has occurred within the organization. The students' job is to respond to this incident, determine the cause(s), and develop procedures to prevent a reoccurrence.

Part of this response involves the preparation of a report to the manager which identifies the nature of the security violation, the scope of the damage (including cost), the remediation measures taken, the procedures followed in pursuing an internal investigation, and recommendations for policy changes (if any).

Responses include the identification of available information residing on a system, any processes which may have been running, or any network connections which may have existed. Once identified, these data will be acquired and archived for subsequent analysis. (It should be noted that the Computer Science project is responsible for producing the software application which ordinarily would be performing the analysis).

The analysis of the data would lead to the uncovering of vulnerabilities and provide information to identify any corrective measures.

Documentation includes the preparation of reports. Such reports would include:
A forensic report detailing the steps taken in processing any data during the internal investigation as well as any findings obtained as a result of the investigation.
An executive summary describing the incident, summarizing the outcome of the investigation, and detailing the costs to the organization.
An executive presentation regarding lessons learned and risk mitigation.
While in the business world much of this work may be done on an individual basis, teamwork may be performed in implementing the planned processes which have been identified during the analysis phase. Students can be guided to a successful completion by applying seemingly disparate skills learned over their school career. In addition, this project will allow them to developing interpersonal and other 'soft skills' desired by future employers.

Insofar as these students are not solving a coding project in their discipline, the coding phase of the three team model may not be applicable in the same fashion as for the computer science or the information systems students. Substitution of other activities, such as gathering and manipulating data, may be more appropriate for this second phase of the project described above.

Information Systems Project

The Information System project concentrates on the design and development of a 'traditional' IS based solution, ensuring that students work together in a team environment as well as incorporate those 'soft skills' cited as deficient by employers.

Utilizing a three or four team methodology, the instructor may elect to condense the analysis phase by providing the specifications, thereby allowing more time for the coding and testing phases.

While there are many interesting projects which can be proposed, one recent suggestion is that the students develop an "e-Vote" application. The selection of the project must allow for inclusion of all the important elements for the IS students. In particular, those elements that deal with the human-computer interaction should be present. The coding of the application must demonstrate the students' mastery of their ability to manipulate data, control the flow of the system, and control interaction with the end user. Additionally, those supporting elements which are integral to any IS endeavor must be demonstrated (such as documentation, testing, walkthroughs, etc).

The e-Vote project, while apparently simple, seems to incorporate all of the requisite characteristics of an IS capstone project. Students will discover during the analysis phase that user requirements may increase the complexity of the project solution. Operational concerns, such as the need for voter confidentiality, will dictate additional functions in the overall project.

Data integrity issues, telecommunications issues, and security issues will demand a more thorough analysis phase of the project. While time constraints may not allow for the implementation and deployment of all functionalities, those functions must be identified and planned for in the analysis phase. This allows the client (instructor) to determine which functions may be deferred.

The possibility exists to adapt this application to a standalone platform, a client-server platform, or a web-based platform. This range allows the instructor flexibility in keeping current with the technology, or in adapting this application to a different platform in a "maintenance phase" scenario.

Grading Considerations

Grading of student efforts for group projects can be difficult. In keeping with the 'real-world' scenario, one possibility which supports providing students with "real" experiences is giving a group grade for the cumulative results of that group. While some may consider that such a method of issuing an all-or-nothing grade for that component may be harsh, in reality system components and code work, or they do not. Bonuses (as well as jobs) are tied to successful completion of projects. Essentially, we are associating the grade to a successful completion. While this may be deemed unduly harsh, a partial grade may be issued according to the degree of the successful effort in the project, despite its failure to achieve overall success. A carefully administered peer-review process can also determine the individual contribution within any group effort.

Likewise, detractors to this grading scheme might protest the lack of rewarding any individual student's contributions or abilities. This can be resolved by allowing part of the overall grade to be based on individual effort, such as coding in the maintenance phase, which can be evaluated on an individual basis.

The sole remaining consideration would to determine the percentage of group versus individual grading in any student's overall grade. We suggest that the individual grade received from the maintenance phase be 25%, and that the group grade can comprise the remainder, of which 50% is derived from the group effort, and 25% comes from the peer-review process.

4. CONCLUSIONS

In keeping with the philosophy that we wish to prepare students to enter a workforce with technical skills and knowledge, we need to address the additional requirement which industry places on our students in the workplace: soft skills which often are not normally developed during their studies. Additionally, an early transition to new modes of knowledge acquisition can be introduced during a capstone project.

The proposed grading scheme can reinforce the aspects of working in a IT environment by ensuring that students must cooperate in a team effort in order to secure a team result.

Furthermore, a transition from a passive undergraduate student to a more proactive student can be made by compelling students to research topics as part of the analysis and coding phases.

5. REFERENCES

Allen, T.J. (1977). *Managing the Flow of Technology.* Cambridge, MA:MIT Press.

Earl, M.J. (1995). The changing shape and skills of the I/S function. Centre for Research in Information Management, Working Paper CRIM WP95/3, London Business School.

Goldhar, J.D., Bragaw, L.K., & Schwartz, J.J. (1976). "Information flow, management styles, and technological innovations". IEEE Transactions of Engineering Management, EM-23(1), pp. 51-62.

Kaufman, H.G. (1974). "Relationship of early job challenge to job performance, professional contribution, and competence of engineers.", Journal of Applied Psychology, 59(3), pp. 377-379.

Katz, R. & Tushman, M. (1983). "A longitudinal study of the effects of boundary spanning supervision on turnover and promotion in research and development.", Academy of Management Journal, 26(3), pp. 437-456.

Lee, Denis M.S. (1986). "Academic achievement, task characteristics and first job performance of young engineers.", IEEE Transactions on Engineering Management, EM-33(3), pp. 127-133.

Lee, Denis M.S. (1992). "Job challenge, work effort, and job performance of young engineers: A casual analysis.", IEEE Transactions on Engineering Management, EM-39(3), pp. 214-226.

Lee, Denis M.S. (1994). "Social ties, task-related communication and first job performance of young engineers.", Journal of Engineering and Technology Management, 11, pp. 203-228.

Lee, Denis M.S., Trauth, E., & Farwell, D. (1995). "Critical skills and knowledge requirements of IS professionals: A joint academic/industry investigation.", MIS Quarterly, September, Vol. 19, pp. 313-340.

McNurlin, B. & Sprague, R.H. (2002). Information Systems Management In Practice,5th ed. Upper Saddle River, NJ: Prentice Hall, p. 317.

Pelz, D.C. & Andrews F.M. (1976). Scientists in Organizations. New York: John Wiley & Sons

Trauth, E. & Hafner, C. (2000). "Meeting the IT skills crisis: An interdisciplinary response.", Proceeding of the Americas Conference on Information Systems, Long Beach, CA, (August).

Thompson, P.H., Dalton, G.W. & Kopelman, R. (1974). But what have you done for me lately – The boss. IEEE Spectrum, (October).

Todd, P.A., McKeen, J.D. & Gallupe, R.B., (1995). "The evolution of I/S job skills: A content analysis of I/S job advertisements from 1970 to 1990.", MIS Quarterly, Vol. 19, No. 1, March, pp. 1-27.

Wilkins, M.L. & Noll, C.L. (2000). "Critical skills of IS professionals: Developing a curriculum for the future.", Proceedings of ISECON 2000, Vol. 17 (Philadelphia): §112.