AC 2007-1889: ENGINEERING DESIGN FOR HUMAN NEEDS: EXPANDING THE SCOPE OF ENGINEERING SENIOR DESIGN

Susan Conry, Clarkson University

Dr. Conry is a faculty member in the Department of Electrical and Computer Engineering at Clarkson University. She obtained her B.A., M.S., and Ph.D. degrees at Rice University. Her interests include engineering education, multiagent systems, and parallel and distributed systems. Dr. Conry teaches in a variety of areas in computer engineering and software engineering.

Engineering Design for Human Needs: Expanding the Scope of Engineering Senior Design

Abstract

The culminating design experience in engineering curricula is usually intended to provide a framework within which the emerging engineer can draw upon an acquired base of knowledge in his or her discipline to solve an open ended problem in that discipline or in a multidisciplinary context requiring contributions from that discipline. In this paper, we show how the culminating design experience can be framed so as to expand the scope of its contribution in the education of engineering students. We describe a pedagogical framework within which educational outcomes associated with multidisciplinary activity, legal, ethical, and professional responsibilities, and the impact of engineering solutions on society can be emphasized. Drawing upon student experiences in design of systems for use by persons with developmental or cognitive disabilities, for persons studying human skeletal structures, and for persons with certain neurological dysfunction, this paper also identifies and illustrates ways of leading students to understand the impact of engineering solutions in a broader social context.

Introduction

ABET accredited programs in engineering must engage students in a "major design experience based on the knowledge and skills acquired in earlier course work and incorporating appropriate engineering standards and multiple realistic constraints"[1]. The educational program must also be one in which students attain a number of demonstrable outcomes, specifically those outlined in the *Criteria for Accrediting Engineering Programs*, Criterion 3 (a) – (k)[1]. Most programs have little difficulty demonstrating some of these outcomes. On the other hand, several of these required outcomes are more difficult to demonstrate.

In this paper, we outline a pedagogical framework for the culminating design experience that also facilitates student attainment of the following Criterion 3 outcomes:

(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
(d) an ability to function on multidisciplinary teams
(f) an understanding of professional and ethical responsibility
(h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
(j) a knowledge of contemporary issues

Thus our desired learning outcomes for the engineering senior design experience are extended beyond those that are required of the culminating design experience and are intended to encompass a broader range of expectations and engage the students in activities that are truly directed towards development of an understanding that technology must serve humanity.

Building on work that has been reported in the European community[2], we first describe our pedagogical framework in terms of its components then indicate how it is instantiated in three contexts, each of which is one exemplar of a software engineering senior design experience.

Pedagogical Framework

Goodyear[2] has suggested that a pedagogical framework has four components:

1. *Philosophy*: a set of beliefs about what knowledge and competence are and about how learning occurs,
2. *High level pedagogy*: the concrete manifestation of the philosophical principles guiding the framework,
3. *Strategy*: a broad brush depiction of plans – a description of what could or should be done to achieve the desired objectives,
4. *Tactics*: a set of specific detailed activities by means of which the strategy is implemented.

The *philosophy* of the framework we have used is based on a fundamental belief that students learn best by doing. They need to make their own mistakes to cement their understanding of the underlying principles that come into play. Knowledge and competence are thus best demonstrated by considering the artifacts that are constructed and the modes of interaction and behavior that are evident as the students engage in learning activity. With regard to the broader issues that our design experience addresses, namely those associated with multidisciplinary activity, legal, ethical, and professional responsibilities, and the impact of engineering solutions on society, our belief is that the best way to foster understanding of these issues is to place students in situations in which they must confront them at a personal level in designing systems that address realistic problems.

Our *high level pedagogy* is probably best termed guided discovery learning. There has been a great deal of interest in problem based learning, constructivist learning and other active learning modes in recent years. The idea is that students build knowledge for themselves, with minimal guidance from their instructors. There is evidence[3] that pure discovery methods are not the most effective methods, but rather that guided discovery may provide the best way of engaging the student in cognitive activity.

The *strategy* that is used in our software engineering senior design experience is very simple.

1. The problem to be addressed is chosen so that it has several relevant dimensions:
   - It must reflect a problem that a real client needs to have solved, and the client must be willing to interact with the students.
   - The students must not have had extensive experience working in the application domain involved, so it will be necessary to interact with the client in an interdisciplinary setting to determine necessary system features.
   - There must be several viable candidate system structures so that students have to evaluate alternatives in order to define the architecture in a manner that meets the client's objectives.

- The problem must have inherent in it a solution that has the potential of improving some aspect of life for people.
- The problem must have dimensions that require the students to grapple with issues such as information security, privacy, ethical use of information, legal constraints on the use of information, and other contemporary issues in the information age.

2. The students must engage in a disciplined sequence of activities over the course of the semester that culminates in a demonstration of the end product to the client. This sequence of activities is guided at a high level but focused on discovery at lower levels. Students are allowed a great deal of latitude in discovery but "reined in" and nudged in a more productive direction should they begin to be unproductive.

The *tactics* employed are structured around design processes that are well known in software engineering. We have ordinarily used a structured set of activities similar to those suggested in the Team Software Process(TSPi)[4]. As originally conceived, TSPi involves three development cycles, with each cycle having several phases:

i. Launch – discuss the process, assign teams and roles within teams, discuss customer needs statement, establish goals for the development cycle
ii. Strategy – determine conceptual design for the product, decide what will be produced in each cycle, establish configuration and management plans, determine risk management strategies
iii. Plan – estimate size of artifacts to be developed, identify tasks to be done, estimate how long they will take to perform, assign to team members, make a schedule for task completion, make a quality plan
iv. Requirements – analyze customer needs statement, engage in dialog with customer, specify and inspect the requirements, generate a system test plan
v. Design – specify and inspect the high level design, develop integration test plan
vi. Implementation – perform detailed design of modules and units, review the design, translate to code, engage in code reviews, compile and test the modules and units
vii. Test – build and integrate the entire system, conduct a system test, produce user documentation
viii. Postmortem – conduct a postmortem analysis of the development cycle and produce cycle report.

Rather than employing a three-cycle process, we have most often made use of two development cycles in a semester due to the length of time required for client interaction and the fact that students have many other demands on their time.

Use of this Pedagogical Framework

We have employed this pedagogical framework for the last several years in software engineering senior design. The senior design course is required of all software engineering majors and is ordinarily taken in the fall of the senior year. The class size is usually on the order of 10 to 18 students. In order to engage the students in a large team environment, the entire class is required to function as a software development team. The class is responsible for identifying the need for specialized subteams and determining the composition of these subteams. They do this as they determine candidate system architectures early in the semester. Students learn to identify where individuals have strengths that can be brought to bear on particular kinds of tasks and they form subteams to take advantage of these strengths. Subteams usually involve on the order of five students but some are larger and some are smaller. The size of the subteams is dependent on the complexity of the tasks for which they are responsible. The class frequently "rebalances" itself after the first development cycle, allocating more manpower to some tasks and reducing the number of students working on others. The role of the instructor is important early in this process, as the students do need some guidance as they get themselves organized. The most difficult issues faced by the students are often centered on communication within and between groups of students and on resolving situations in which expectations have not entirely been met.

In this section, we outline three different design scenarios, emphasizing the character of each application domain and the ways in which each fosters accomplishment of the learning objectives that are articulated in concert with the process-oriented aspects of the tactics associated with the pedagogical framework. To ensure that our curriculum provides a comprehensive treatment of the topics and skills an emerging software engineer should master, the department has adopted a set of core learning objectives for each required course in the program of study. The core learning objectives for the software engineering culminating design experience are:

1. Students will participate in the process of carrying a significant software development effort from a conceptual idea through integration and testing of the complete product.
2. Students will be thoroughly familiar with issues encountered in the maintenance phase of the software life cycle.
3. Students will be aware of the social and ethical issues of concern to software engineers.
4. Students will learn how to function as effective members of a team of professionals.

Design Scenario 1: Cognitive Rehabilitation Software

The incidence of head injuries and stroke is highest in age groups ranging from the late teens into the early sixties. It is very important in rehabilitation of these individuals that cognitive function be redeveloped. In most cases there is speech or dexterity limitation as a result of the injury but it is critical that people regain their ability to function in the world to the greatest extent possible. Among other things, this means that they may need to re-learn associations between printed words and images in order to re-learn how to read; they may need to re-learn about basic mathematics (addition and subtraction) and currency, so that they can make simple change or do effective grocery shopping. On a more sophisticated level, they may need to learn how to make appropriate choices concerning what to wear under various weather conditions or how to shop so that their grocery choices are appropriate for their nutritional needs.

Discussions with our partner client at a major rehabilitation facility revealed that many of their clients are very interested in developing their computer skills and also in electronic games. The occupational therapists at our partner facility had determined that the software commercially available for cognitive rehabilitation was based on software developed for early childhood education. The images and activities available to adult or teen victims of head injury or stroke simply were not age appropriate, and they were not appealing.

Based on this set of observations, a problem was framed: design software suitable for cognitive rehabilitation of individuals in their late teens to sixties. After the initial launch, strategy, and planning phases (which took one to two weeks), the students needed to engage the client base. A trip to the rehabilitation center was organized for the purpose of allowing the students to interact with two groups of people: the occupational therapists and the patients themselves. It was an eye opening experience for the students. They spent a day interacting with occupational therapists and severely handicapped individuals in a residential rehabilitation facility. The nature of the interactions was very rich, and the students came away from the experience with a profound sense that their activities could have a significant impact on other individuals' quality of life.

From this experience, the students carried away a number of important observations. Some of the more important ones are:

- Humor is incredibly important to incorporate in the patient activities.
- The images used need to be clear and colorful.
- The sounds used need to be designed so that their rate of repetition can be changed.
- The human/computer interface for some activities needs to be single-switch (one-button) while others can involve scanning from a list or group of images or icons.
- It would be very helpful if the system were web-enabled so that patients could use them from home with statistics on performance gathered for therapist to review in monitoring progress.
- The set of activities available to a given patient should be controlled by his or her therapist and the therapist should be able to gradually offer increasingly complex levels of interaction and cognition.
- Security of patient data is legally required and of first order importance.
- The computers available for patient use are not the latest and fastest, so the required hardware system specification and associated proprietary software needed to be minimal and cost effective.
- Web access could be restricted to dial up access only in some instances.

Based on what they had learned from their client base, the students identified system requirements, verified them with the occupational therapists again, decided what functionality they would include in a first development cycle and which would wait until a second, and developed the baseline (cycle 1) product. At the conclusion of this stage, the students again went into the rehabilitation facility to watch how the therapists and the patients reacted to their baseline product. As one would expect, there were some additional features desired by the client and some of the images and activities did not "go over" as the students expected. Others (which the students thought were sort of silly) went over enormously well – the political humor injected into one of the activities was particularly appreciated. Cycle 2 was an exercise in maintenance mode and further enhancement with a final demonstration to the client carried out in distance-mode. At the conclusion of the project, the occupational therapists were given administrator access to the system so that they could enroll patients and use the activities as they desired.

Significant learning outcomes of this experience included learning how to interact in a large team development involving professionals from multiple areas, appreciation for the legal and ethical treatment of confidential patient data, understanding of the importance of standard protocols, personal experience communicating with people who cannot communicate in conventional ways, and first hand exposure to the benefits that can be derived from the appropriate use of technology. Another critical learning outcome was attained as the students demonstrated their ability to engage in all aspects of the software development process – from concept to deployment and maintenance phase activities.

Design Scenario 2: Physical Therapy Virtual Bonebox

Before they can study how muscles work dynamically in the human body, physical therapy students must be able to identify structures in the human skeleton and features of those structures. The course in which the student learns these things has a textbook with many pictures and also has a laboratory component in which the student spends time in the laboratory studying bones (or models of bones) at great length. We were asked by our physical therapy department to consider design and implementation of a system that would serve as a virtual laboratory for physical therapy students. The desired system would have a number of features found in instructional software such as Blackboard or WebCT, but the application domain imposed some requirements that appeared to go beyond what was available on our edition of Blackboard.

Again, the students engaged in the first three phases: launch, strategy, and plan, then met with the director of the physical therapy several times over the course of a week. The Director of Physical Therapy showed them what the physical therapy students needed to learn, demonstrated the kinds of activities physical therapy students do in their laboratory, described what he wanted in the way of a quiz/activity authoring system, and made available to them a substantial set of copyrighted photographs he had commissioned. These photographs were to form the underlying source of data for the students to use, but they needed to be cataloged and manipulated in different ways depending on the mode of system operation (study, quiz, practice quiz, quiz/exam author), etc. Some amount of dynamic labeling of the photographs (or copies of these photographs) was needed. In addition, the instructor imposed some requirements concerning records of student activity and security of all data regarding student access, progress, and performance.

Based on what they had learned from the director of the physical therapy program, the students engaged in the remaining development cycle 1 activities, interacted further with their client even more during this cycle as they did some rapid prototyping of the interface, and concluded cycle 1 with a demonstration to their client. At the conclusion of development cycle 1, the students did a client demonstration and came away with some significant modification and further development targets for the second development cycle. At the conclusion of the semester, the students demonstrated their system and delivered appropriate documentation to their client. The director of physical therapy indicated that this tool set gives him a much more flexible environment in which to develop his instructional pedagogy and the students can do at least some of their lab work in a virtual environment.

Significant learning outcomes of this experience again included learning how to interact in a large team development involving professionals from multiple areas, appreciation for the legal and ethical treatment of confidential student data, understanding of the importance of standard protocols (in this case protocols involving naming conventions in another discipline), and first hand exposure to the benefits that can be derived from the use of a virtual learning environment. Here, too, another critical learning outcome was attained as the students demonstrated their ability to engage in all aspects of the software development process – from concept to deployment and maintenance phase activities.

Design Scenario 3: Diagnosis and Monitoring Treatment for Tremor Patients

Our third example design scenario was suggested by the director of our Rehabilitation Engineering Center. As the population ages, an increasingly large fraction of the population is affected by tremor. This can have a significant impact on individuals' functional independence. Parkinson's disease and essential tremor may cause trembling, gait or speech abnormalities and an inability to control fine motor actions. Treatment often involves medication with L-dopa and enzyme inhibitors or deep brain stimulation. Because of adverse side effects, the medication dosage must be closely monitored, and many individuals become less and less responsive to a given medication over time.

Assessing the effectiveness of therapies for tremor continues to be largely subjective. A trained neurologist can detect subtle shifts in motor performance by using visual observation, neurological exams and an assessment of their client's fine motor skills. Assessment of fine motor skills often includes testing patients' ability to draw simple shapes, print short words or sign their names. Distinct and repeatable characteristics appear as the disorder progresses -- the signature becomes much smaller and the drawing of objects much less fluid. Currently these tests are carried out in a more or less ad hoc manner using pencil and paper and the data recorded consists of subjective assessments of the patients' performance.

With the advent of instrumented graphical input devices, it has become possible to devise automated metrics for assessing tremor and spasticity based on graphical drawings, handwriting motions, or wrist responses to sudden movements. Interacting with our client (the Director of Rehabilitation Engineering), the students identified three major classes of users for the proposed system: patients, clinicians, and researchers. Each of these user classes has a distinct set of needs that must be met by the system. Some of the relevant requirements identified by the students in collaboration with their client are:

- When turned on, the system should provide an appropriate interface on its screen and allow the clinician to quickly enter patient demographic data.
- During the actual tests, the graphical input device will be recessed into a flat writing surface so as to minimize any effects of wrist flexion.
- Through an on-screen menu, the clinician will be able to select from a variety of signature and graphical drawing, tracing or tracking exercises.
- The input medium must be capable of capturing stylus data and produce files containing both spatial attributes (X and Y coordinates), and temporal data (time of entry) for each point.

- The data generated will then be transmitted over a secure link to a server for storage in a database – stringent security requirements are imposed.
- The clinician interface will allow the clinician to browse relevant data in the database and select various patient signatures and drawings for review.
- The clinician should be able to select any two signatures/drawings and "play" them back on the PC's monitor in real time for side-by-side comparison.
- The system should allow each patient to serve as his or her own control for the purpose of monitoring therapy.
- The software operations associated with these tasks should be no more complex than in other office applications.
- An acceptable system should require no more than one day's training for the clinician to learn and not impose excessive cognitive demand – the users are not computing professionals.
- The original graphical data files, and associated notes, should be downloadable on a per-patient basis, with access restricted to ensure compliance with confidentiality regulations.

The students specified the system structure and validated this with the customer. They became very entrepreneurial in acquiring appropriate hardware and software resources, actively sought information concerning the stringent security regulations required by the client, did significant rapid prototyping using different technologies before specifying the underlying software platforms, and arrived at a system with baseline functionality by approximately mid-term. In a second phase, they consulted with individuals with knowledge of the ways in which the raw data would be used, determined analysis algorithms and modes of display for the results that were user-appropriate, and augmented the system to produce a cycle 2 product. They also did significant work in assessing the degree of impact such a system can have on disease management. At the end of the term, the client proposed that this project be extended to the point at which it can be placed in a clinical setting.

The significant learning outcomes of this experience again included learning how to interact in a large team development involving professionals from multiple areas. The magnitude of the legal and ethical implications of their solutions were particularly evident in this case because one of the target audiences includes patients treated at a Veterans Administration facility and the security requirements imposed by that organization are very strong. They also were exposed to the issues that arise when one deals with human subjects – preparation of a protocol for IRB review was discussed. Here, too, another critical learning outcome was attained as the students demonstrated their ability to engage in all aspects of the software development process – from concept to deployment and maintenance phase activities.

Evaluating the Learning Experience

As part of the continuous improvement process incorporated in our development of educational programs, we assess the degree to which learning outcomes are attained in each of our courses. As was mentioned previously, the core learning objectives of the culminating design experience are:

1. Students will participate in the process of carrying a significant software development effort

from a conceptual idea through integration and testing of the complete product.
2. Students will be thoroughly familiar with issues encountered in the maintenance phase of the software life cycle.
3. Students will be aware of the social and ethical issues of concern to software engineers.
4. Students will learn how to function as effective members of a team of professionals.

In addition to these core learning objectives, we have emphasized one more learning outcome that is fostered by use of this our pedagogical framework:

5. Students will understand the potential impact of engineering solutions to society.

The data collected and used in assessing the degree to which these outcomes are met is gathered from several sources.

- We ask the students to evaluate themselves *and each other* on a confidential basis regarding the quality of their team interactions, their level of professionalism, their dependability, and the quality of their deliverables (individually and collectively). The instructor evaluates each student in all of these dimensions as well based on direct observation of performance across the semester. This data provides a basis for determining the degree to which each student has learned to function well as an effective team member on a team of professionals. The evaluation scale ranged from 5 (outstanding) to 1 (poor), and Table 1 reflects max, min, and average ratings for each of the last two years. In years prior to 2005-06, a different rubric was used, making it difficult to compare results head to head.

	2006-07			2005-06		
	Max	Min	Avg	Max	Min	Avg
Team Interactions	5.0	2.8	4.2	4.8	1.4	4.0
Cooperativeness	5.0	3.2	4.3	4.9	1.3	4.1
Dependability	4.9	2.2	4.2	5.0	1.1	4.0
Quality of Work	5.0	2.5	4.2	4.8	1.1	3.9
Conflict Resolution	4.9	2.4	4.1	4.5	1.1	3.7

Table 1. Student and Instructor Evaluation:
Evidence of Ability to Function as a Member of a Team of Professionals

Based on this data, we conclude that the relevant learning outcome has been attained to a satisfactory extent. The minima in 2005-06 are at an unacceptable level, but the individual whose data reflects that minimum did not pass the course. Although the data are not directly comparable for years prior to 2005-06, interpretation of the data that was gathered for those years was consistent with the data observed here.

- We seek input from our client regarding the matter of customer satisfaction: does the client perceive that the product that was delivered reflects a high quality engineering design effort that carried a problem solution from concept through delivery of an acceptable product. This information is sought at mid-term when the cycle 1 deliverables are demonstrated and again at the end of the semester. The client input provides a basis for determining the degree to which students have engaged the issues that arise in all phases of the software life cycle – from inception through maintenance phases. This data has been solicited in a less formal way than the data outlined above. The instructor engages the client in a conversation that directly addresses these issues. In each case, the client has been highly satisfied. In the future, we plan to utilize a more refined rubric for the client input so that the results can be tracked in a systematic manner.

- We seek input from our graduating seniors and our alumni with the aid of surveys. One such survey is administered to graduating seniors as they have just completed the program. Another is administered to alumni who graduated two years previously, so have been in the workplace for two years. These surveys give us information that allows us to draw conclusions about the degree to which graduates of the program are aware of social and ethical issues of concern to software engineers, how well they have been able to work effectively as members of a team of professionals, and the extent to which they understand the impact of engineering solutions to society. The data collected reflects their relative agreement with statements that describe awareness of these issues on a scale of 1 (strongly disagree) to 5 (strongly agree). Table 2 summarizes the survey data relative to these issues, with snapshots taken in 2001 and again in 2003. It should be noted that the 2003 data reflected a cohort of seniors whose senior design experience was in the fall of 2000, when the design project was not chosen in the context of the pedagogical framework that is described in this paper. It is not surprising that the data reflect less of a sense of focus on the social implications of engineering solutions. The 2001 cohort had a design experience that, while it was conducted prior to the identification of the current framework, had characteristics that are consistent with it.

	2003	2001
Social and ethical responsibility	4.0	5.0
Collaborate effectively as part of a team	5.0	5.0
social implications of engineering solutions	4.0	4.5

Table 2. Alumni and Senior Survey Data

- The instructor of the course watches the student interactions and behaviors carefully during the course of the semester, with particular attention to observing issues associated with all of the outcomes mentioned above, but especially:
    - the nature of the student interaction with the client in a multidisciplinary context
    - the degree to which students are aware of the legal and ethical issues that are pertinent to the problem being solved and its solutions
    - the degree to which students show that they are thinking about the kinds of impact their system can have on peoples' lives

These observations are tracked periodically through the semester so that at the end of the semester the instructor is in a position to draw conclusions based on a rubric such as the one found below.

	4	3	2	1
Nature of interaction with client in interdisciplinary setting	most students active in initiating interactions; active engagement of all students in understanding the multidisciplinary setting; probing, interesting questions	a few students drive the interactions, all students actively engage to a meaningful extent with the multidisciplinary setting	the client takes the initiative for interaction; students engage but at the prompting of the client; multidisciplinary activity is evident on the part of most students	little or no engagement with constraints of another discipline; student focus is on trying to get others to understand his environment
awareness of legal and ethical issues	students take the initiative to explore and identify the relevant issues in a self-directed manner	when asked what legal or ethical issues are relevant, students consider these issues in a comprehensive manner, consulting appropriate resources in doing so	students engage in exploring legal and ethical issues but only as they are pointed out to them by the instructor; the instructor must take the lead in directing their consideration	little or no awareness that there are legal or ethical issues inherent in the problem and its solution
impact on humanity	students seek to incorporate features because they will enhance the potential to impact peoples' lives	students choose a problem in part because of its potential to impact humanity, but the technology is equally important	students are able to identify ways in which solution impacts humanity when asked	students are focused on the technology with little or no regard to how it impacts humanity

Table 3. Rubric for Evaluation of Student Learning

Table 4 summarizes the history of instructor observations over the last three years. The choice of the tremor project, with its clear link to Veterans Administration activities and the coincidental widely publicized security breaches that occurred relative to VA data may well explain the high scores in 2006-07. In addition, several of the students had relatives with Parkinson's disease and the client had direct experience with essential tremor patients. There is not as much linkage with impact on humanity of the Virtual Bonebox, though the students could clearly identify the benefits to other students and also to the educational experience of a virtualized laboratory experience. In 2004-05, the cognitive rehabilitation software project resulted in multidisciplinary activities that were less strongly reflective of tight coupling in part because the occupational therapists, while interested in the project, were also required to function in their professional setting. This setting placed a high premium on time spent with the patient, so that it was harder to claim a great deal of therapist time for multidisciplinary activity. The awareness of legal and ethical issues and of the impact on humanity are present in all three scenarios, but more strongly in those that are related to systems with patient interaction components.

	2006-07	2005-06	2004-05
Nature of interaction with client in interdisciplinary setting	3.8	3	3.2
awareness of legal and ethical issues	3.8	3.2	3.5
impact on humanity	3.5	3	3.5

Table 4. Instructor Assessment of Learning Outcomes

Summary

In this paper we have described a pedagogical framework with associated strategy and tactics that focuses on the choice of design context and the tactical management of the design experience for senior design in software engineering. The available data indicates that the component of the strategy that drives the choice of design context is particularly effective when evaluated from a perspective that seeks to expand the scope of the design experience. Students have engaged in activities that serve to expand their awareness of their legal, ethical, and professional responsibilities and also to have enhanced their understanding of the impact that engineering solutions to problems can have as technology is developed to serve humanity's needs. In addition, the tactics used in managing the process serve to foster the development of strong teamwork skills and engage the students in multidisciplinary interactions as they produce a software system. It is our observation that little in this pedagogical framework need necessarily be applied only in the context of software engineering senior design. The philosophy, high level pedagogy, strategy, and tactics can readily be adapted to other engineering disciplines as well.

Bibliography

1. *2006-07 Criteria for Accrediting Engineering Programs*, ABET Engineering Accreditation Commission, Baltimore, MD.

2. P. Goodyear, "Pedagogical frameworks and action research in open and distance learning", *European Journal of Open, Distance, and E-Learning*, 1999, http://www.eurodl.org/materials/contrib/1999/goodyear/index.html

3. R. E. Mayer, "Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instructions,: *American Psychologist,* Volume 59, Number 1, 2004, pp. 14-19.

4. W. S. Humphrey, *Introduction to the Team Software Process*, Addison Wesley, 2000.