

Multidisciplinary Computer Science Design Projects

Kenneth L. Alford
United States Military Academy

Introduction

By its very nature, engineering represents an integrative and multidisciplinary experience. In their senior year, students majoring in computer science at the United States Military Academy at West Point, New York are required to take two multidisciplinary senior project design courses. Each course is 3.0 credit hours with a 0.5 credit lab. These courses are part of an ABET-accredited computer science program.

Throughout the year faculty members from several engineering departments gather information on suitable multidisciplinary senior design projects. Project customers range from Army research programs to inter-collegiate design competitions to local projects that benefit one or more campus organizations.

This paper outlines how computer science multidisciplinary senior design projects are organized and conducted in the Department of Electrical Engineering and Computer Science. It discusses, in turn, the structure, course content, sample projects, objectives, process, assessment, successes, challenges, and lessons learned that are associated with these multidisciplinary student projects.

Structure

With few exceptions, all computer science senior design projects are multidisciplinary. The project teams include students from one or more of the following engineering disciplines: computer science, electrical engineering, mechanical engineering, civil engineering, systems engineering, environmental engineering, and information systems engineering. This enables student teams to analyze problems from several viewpoints. These differing perspectives usually result in a more thorough, and in some cases, more innovative design and implementation.

The senior design project for students enrolled in the computer science major is a two-semester design, build and test experience obtained through two courses – CS407A, Computer Systems Design I, and CS408A, Computer Systems Design II. CS407A is taken by seniors during the fall

The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Military Academy, the Department of the Army, the Department of Defense, or the United States Government.

semester; CS408A is taken the following spring semester. CS407A emphasizes software engineering and design principles. Students are assigned to project teams and determine specific project requirements with their project advisor. The primary student references are Dennis¹ and Brooks.² In CS408A, students implement, test, and complete their project.

Completed project solutions must address technological, social, political, and economic considerations in the design and fielding of their completed product. Ethical engineering decision-making consistent with the safety, health, and welfare of the public is also an important consideration.

Multidisciplinary senior design project teams range in size from four to twenty students. The organization of individual teams varies from teams with students from four or more engineering disciplines to teams with students from two engineering disciplines. Each team configuration provides its own set of opportunities and challenges for both the students and their faculty advisors.

Course Content

During CS407A, each project team must: (a) evaluate their project to determine what needs to be done in order to reach a successful completion prior to graduation the following spring, (b) solicit and record specific project requirements from their client, (c) organize their project team to effectively and efficiently complete the project prior to graduation, (d) schedule all project requirements, (e) identify technological skills that are required to complete their project, (f) identify which technological skills they need to learn, (g) complete a technology prototype demonstration, (h) complete a requirements design document, (i) complete a detailed software design document, (j) conduct research to see what prior work has been completed on the topic of their project, (k) complete the first half of a research paper outlining the nature of their project, prior work completed in this area, and the design methodology they are taking to solve their design problem.

During CS408A, each project team must: (a) complete a detailed test plan, (b) adjust their project schedule to ensure completion, (c) write software to complete the computer science portion of their project, (d) test, integrate, and fix software errors, (e) complete a detailed user manual, (f) field an initial version of their completed project, (g) correct problems and errors in the initial version, (h) field a final version of their completed project, (i) complete a research paper outlining their entire integrative experience, (j) demonstrate their completed project during the USMA Projects Day which is the Academy's equivalent of a college science fair with booths, displays, demonstrations, discussion, and evaluation.

Sample Projects

Student teams work on a wide variety of research design projects. Many of them produce graduate-level results. Here are a few sample projects from recent academic years:

- **Autonomous Vehicle.** This team—which consisted of students majoring in computer science, mechanical engineering, civil engineering, electrical engineering, and

systems engineering—was responsible for constructing an autonomous unmanned vehicle that was capable of self-navigation. Students designed and built the autonomous vehicle. The final vehicle incorporated dual digital cameras, a laser range finder, and an onboard laptop computer that responsible for movement decisions. The completed vehicle participated in a national collegiate competition.

- **Autonomous Radio-Controlled Plane.** This student project team contained electrical engineering, mechanical engineering, and computer science students. The goal of this project was to modify a radio-controlled plane to fly to a pre-designated location without direction from ground controllers. The aircraft was flown by remote control during takeoff and landing, but used an onboard computer and Global Positioning Satellite (GPS) information to guide it through the flight plan.
- **Handheld Terminal Unit.** This team contained computer science and electrical engineering majors. The Handheld Terminal Unit project team worked to replace the Army's Handheld Terminal Unit (HTU) with a platform independent, smaller, lighter, and less expensive system that can run on a personal digital assistant (PDA). HTU's are used for a variety of purposes—calling for fire from Apache attack helicopters, receiving or sending operation orders, submitting information on battlefield surveys, etc. The Army's current HTU costs approximately \$18,000 and weighs 8.6 lbs. PDA's, on the other hand, weigh less than 1 pound and cost less than \$200 each. This project was conducted in coordination with the Program Manager, Advanced Field Artillery Tactical Data System.
- **Controlled Parafoil Descent.** This team contained electrical engineering, computer science, and mechanical engineering students. The Controlled Parafoil Descent project created an unmanned aerial vehicle that was capable of autonomously navigating itself to a pre-designated location. The computer science team was responsible for designing and implementing the software portion of the flight control package. The software design team used a personal digital assistant as the hardware platform. GPS hardware supplied a stream of latitude, longitude, and altitude information to the PDA. Software running on the PDA calculated the current flight path and any required adjustments.
- **Terrain Trafficability.** The Terrain Trafficability team consisted of computer science and geography majors. This project team developed a rule-based expert system to predict the trafficability of a topographical region. Trafficability decisions are based on data read from processed map and satellite imagery files. The data input file included information regarding terrain, soil, slope, diameter and spacing of vegetation, and ground moisture levels. After applying a series of rules, the expert system determined which specific areas within a given map grid would support both tracked and wheeled vehicle traffic, tracked vehicle traffic only, wheeled vehicle traffic only, or no vehicle traffic.
- **Laser Displays.** The Laser Display project was a joint effort between computer science, electrical engineering, and mechanical engineering students. The laser

display is an array of lasers that projects pictures or text strings with simple animation and sound on a horizontal surface.

- **Terrain Visualization.** This team contained computer science and information systems engineering majors. This project translated large digital elevation terrain data files into a 3D topographic model displayed on a computer screen. The system communicated with the Advanced Field Artillery Data System in order to obtain and plot information regarding the location of friendly and enemy units on the terrain visualized.
- **Intelligent Power Plant.** This team consisted of computer science and mechanical engineering majors. West Point has two options for satisfying its energy needs. It can either buy all of the electricity or it can buy some and generate the rest using its own power plant. It can be cheaper at times to generate some of our own electricity while at other times it is cheaper to buy all of it. This project figured out how much electricity West Point should produce each day, if any, to save the most money.

Objectives

Students who complete CS407A, Computer Systems Design I, during the fall semester should meet these course objectives:

- Effectively lead and work with peers.
- Identify risks, schedule and budget resources, and follow through on project planning.
- Evaluate and select new technologies needed for a project based on economic, political, and social uncertainties.
- Learn and apply new technologies.
- Clearly articulate client requirements for a project.
- Complete a detailed project design.
- Consider human-computer interaction during all stages of design.

At the conclusion of CS408A, Computer Systems Design II, students should meet these additional course objectives:

- Implement new technologies as a part of project completion.
- Respond to changing project circumstances.
- Deliver a completed project to the client.

- Formulate project results for a scholarly audience.

Process

Computer Science faculty members are continually on the lookout for suitable senior design projects. Project possibilities are gathered by the course director throughout the year with the possibility of using them during the following academic year.

A great deal of effort is put into developing a catalog of projects that addresses the needs of the graduating seniors. If a workload error is made early on, an instructor can add or remove requirements to ensure the team is challenged, but not overwhelmed. A more difficult faculty requirement is to match problem domains with the backgrounds and interests of students.

Each summer Academy faculty members meet and determine which projects will be supported during the upcoming academic year. Computer science students are assigned to a design team within the first three weeks of the fall semester. All student teams are integrated and interdisciplinary.

Computer science students are assigned to a unique multidisciplinary project in computer science teams of two or more. Project teams work together for the entire academic year. The specific project is different for each student team.

Each project is assigned to a CS407A instructor for oversight, coaching, and grading purposes. Each project is also assigned to a faculty member who serves as the project advisor. If a project is for an outside agency or organization, there will be an additional outside advisor who will help define system requirements with the student project team.

Shortly after the beginning of the fall semester, the computer science team members are exposed to the basics of project management and review the software development process in order to apply a discipline-specific framework to their project. In addition to answering to the instructor regarding how their system will be designed and built, each project team must work with the client and advisor(s) for whom their system is being built. Project team members and their client determine exactly what kind of finished project will be delivered at the end of the second semester.

Each student team is responsible for organizing themselves, selecting a team leader, and figuring out how to best understand and complete their assigned project. During both semesters, lessons and assignments are organized to teach students essential project management skills that coincide with the design, building, and testing phases of their team projects.

Student teams are encouraged to seek opportunities, with faculty assistance, to present and publish their research and project results at academic conferences. Each year several project teams are successful in their search for publishing and presentation opportunities.

Although set primarily in an academic environment, the senior design projects share many common elements with “real world” software projects. Students face numerous technical

challenges. Projects are designed so that every team is required to learn and apply one or more new technologies in order to complete their project. Students have strict schedule requirements. The academic semester is fixed at 16 weeks; all project work must be confined to that time. Students experience management problems. All student projects are peer-managed, and like their real world counterparts some projects are better managed than others. And like industry projects, resources are also constrained.

The focus of the first lessons in CS407A is to understand system requirements. By the end of the third week of the semester, instructors have covered the theory of information gathering and systems analysis and have paired student teams with their clients so that they can immediately apply their newly acquired knowledge to their recently assigned research project. Instructors check their progress in an informal review, and then let students finish writing their requirements specification.

Soon after being assigned to a project team, each student team is asked to identify several technologies that are essential to the successful completion of their project. Students next identify which of those technologies they currently do not understand. Each student selects one of the new technologies and sets about to become the team expert on that topic. At the midpoint in the semester, each student is given an opportunity in front of faculty and peers to demonstrate the successful application of their technology to their project.

The requirement for an individually-graded technology prototype originated from two observations. First, we noticed that students would often make poor decisions early in their design process because they didn't know much about the tools they would be using. In some cases teams were unable to recover from these early mistakes and could not satisfactorily complete their project. A technology prototype allows teams to validate their decisions early and hopefully avoid costly mistakes.

All team members are trained in the use of formal scheduling, tracking methods, and tools. Then, as their project passes through its various life cycle phases, the diverse perspectives of the team members aid in the discovery of tasks and dependencies that might well be overlooked by a homogeneous group of computer science software engineers.

During the first semester students have two major graded events – the Systems Requirements Specification (SRS) and the System Design Document (SDD). There is not enough time during these two courses to attempt to duplicate the software acquisition process prescribed by the Department of Defense; therefore, the SRS and the SDD represent hybrid versions of documents that are used to procure software for the federal government.

To help students learn how to write good requirements, we use a three-step approach. First, we discuss textbook and other readings and examples of good and bad requirements. Whenever possible, we use excerpts from the work of previous student teams to show current students items that are similar to what they will find in their own projects. Second, we pair teams with their clients immediately, so there is proximity between learning and doing. This just-in-time learning is key to reinforcing the concepts discussed in the classroom. And third, we hold frequent informal in-progress reviews to address any questions the students may have. We have found

that this interim review allows us to find problems and address questions that would not have become apparent until the submission of the final document. It also forces students to get started early.

The Software Design Document that each team completes describes in detail the system that the computer science students will build during the following semester. Students also outline how they test their system to ensure that it has been properly built to the requirements they elicited from their project advisor.

During the second semester, students must implement the requirements and detailed design they produced during the first semester. In the first half of their second semester, students write computer code, debug their code, integrate software modules into larger software programs, and test each resulting portion of their system.

A working version – Deliverable Version #1 — of their completed multidisciplinary design project is due in the week prior to spring break. Version 1 is often a no-frills version of their project. After spring break the entire team meets with the client and begins working on Deliverable Version #2.

The main product of this two-semester course is a fully functioning software system that meets the requirements delineated by the client during the first semester. By the end of the second semester, each project team has also completed a users manual, several peer evaluations, a project website, and the research paper that chronicles their senior design experience.

Assessment

The goals of the assessment tools used in CS407A and CS408A are: (1) to help students effectively and efficiently organize, define, design, and complete their design projects, and (2) to provide timely and useful feedback to student project teams. An overview of the assessment tools for CS407A, taught each fall, is found in Table 1.

What	When
Faculty Project Nominations	During the previous year
Selection of projects	July-August
Team Assignments	August
In-Progress Review #1	September
System Requirements Specification	October
In-Progress Review #2	November
Software Design Document	December
Research Paper (First Half)	December

Table 1. CS407A (Fall Semester) Assessment Tools

Table 2 contains a list of the assessment tools using during the second semester.

What	When
Status Reports	Weekly
In-Progress Review #3	January
In-Progress Review #4	February
Peer Evaluation #1	May
In-Progress Review #5	March
Project Deliverable, Version 1	March
In-Progress Review #6	April
Peer Evaluation #2	May
Research Paper (Final)	May
User Manual	May
Project Deliverable, Version 2	May

Table 2. CS408A (Spring Semester) Assessment Tools

During the first semester, student grades are divided evenly between individual and group work. In the second semester, 80% of each student's grade is based on group work. The following list describes the specific evaluation tools that are used to assess each student and multidisciplinary design team.

- **Examinations.** To help provide incentive for students to study and learn the principles and terminology of software design and software engineering, there are two in-class, one-hour examinations administered during the first semester. There are no examinations given during the second semester.
- **Faculty Advisor/Instructor.** Each design team has a faculty advisor assigned to guide and mentor them through their design experience. The faculty advisor monitors and evaluates how well the students meet the course goals and objectives.
- **Client.** Each design team has a client who works with the faculty advisor and team throughout their design experience. The client provides comments at the completion of the project regarding how well the team met the course goals and objectives.
- **In-Progress Reviews.** The faculty advisor holds numerous In-Progress Reviews with each team (two during the first semester and four during the second semester) to discuss their successes, challenges, and project assignments.
- **System Requirements Specification.** Students write a technical report for each of the major design elements and the final capstone computer systems design which requires them to demonstrate their knowledge of technological, social, political, economic, and ethical factors and an understanding of the impact on their system design. Students must also work with their client to delineate the requirements for the system they will create. All team members and their client sign a Project Contract in which the client acknowledges that the requirements are accurate and as complete as

possible, and the students commit to developing the system defined by the requirements.

- **Technology Prototype Demonstration.** Senior design projects ensure that each student will be required to learn, implement, and demonstrate a technology that is new to them. During the fall semester, each student identifies, learns, implements, and demonstrates a new technology that is essential to the completion of their team's integrative experience project. Each student prototype demonstration investigates a different technology. Here are some of the technologies learned in the most recent series of fall prototype demonstrations: programming microcontrollers, developing genetic algorithms, communicating through a serial port, rendering 3D graphics representations in Java, implementing a flocking algorithm in Ada, running Flash applets from a Visual Basic shell, and controlling a radio-controlled plane through software.
- **Software Design Document.** Students write a technical report that outlines in detail the software modules in their system. The design document includes detailed pseudocode and data definitions. It also includes a draft test plan identifying how they intend to test each software requirement listed in their System Requirements Specification.
- **Weekly Status Reports.** Every Friday during the second semester, each project team leader provides a brief status report to their CS408A instructor. The status report contains the following information for each team member: (1) one sentence stating what they accomplished on their project during the previous week, (2) how many hours they worked on their project during the previous week, (3) one sentence stating what they plan to work on during the coming week, and (4) how much time they estimate they will work on their project during the coming week. Team leaders may also add any additional information they desire, such as any problems or challenges the team is facing or any successes that they enjoyed.
- **Test Plan.** Each team writes a technical plan explaining in detail how the software in their project will be evaluated to ensure that it fulfills and conforms to the requirements previously identified.
- **User Manual.** Each project team completes a detailed User Manual for their client.
- **Peer Evaluations.** Students evaluate the performance and contribution of the members of their team at selected times during each semester.
- **Briefings.** Throughout the two-course sequence, student teams are required to periodically brief the status of their engineering design methodology and project status to their client, faculty advisor, other interested faculty, and their peers.
- **Project Deliverable, Version 1.** Midway through the second semester, each student team delivers to their client an initial working version of their completed project.

- **Project Deliverable, Version 2.** Near the end of the second semester, each project team modifies deliverable version 1 as requested by their client and delivers a final working version of their completed project.
- **Research Paper.** Each project team writes a research paper that summarizes their senior capstone design experience. The outline of the paper is provided to students. Each paper contains the following thirteen sections: Abstract, Keywords, Introduction, Problem Statement, Related Work, Environment, Design Methodology, Implementation, Testing, Lessons Learned, Future Work, Summary, References, . The first half of the paper (through the Design Methodology section) is written during the first semester, and the complete paper is submitted at the end of the second semester. The goal for each project team is to have their paper published or presented at an academic conference.

Other tools and products are used to separately assess the structure of the two senior capstone design courses. Student surveys, a course summary document, and a course proposal document are the primary instruments used.

- **Student Surveys.** Twice each semester, course surveys assess student ability to meet the course goals and objectives. Feedback is also solicited regarding other elements of the senior design course, such as the instructional methods, textbooks selected, and specific project requirements. Students are also invited to submit ideas regarding efforts that can be taken to improve the course. Survey information is only made available to individual course instructors and not to their supervisors.
- **Course Summaries.** Course directors use student, client, and faculty feedback to assess the course. They report their findings and conclusions in a Course Summary document at the end of each semester. The Course Summary is the primary vehicle for implementing changes required to better achieve the course goals and objectives and the Academy's overarching program goals. The Computer Science Program Director reviews and approves all Course Summaries in the Computer Science program.
- **Course Proposals.** Where the Course Summary is an analysis of the previous semester, the Course Proposal is the formal course plan for the following year. It lists specific recommendations for the next time the course will be taught. Topics discussed in recent CS407A and CS408A course proposals include changing the textbook, adding a supplemental textbook, adding Uniform Modeling Language to the course, using Microsoft Project to track project deadlines and schedules in student projects, adding the weekly project report, eliminating a written implementation plan, and adding the research paper requirement. The Computer Science Program Director reviews and approves all Course Proposals in the Computer Science program.

Encouraging Interdisciplinary Cooperation

There are many actions that can be taken to improve cooperation and coordination between interdisciplinary student design teams. Here are few ideas that have worked at West Point:

- **Meet Other Capstone Instructors.** Every instructor who teaches an interdisciplinary design course should develop a good working relationship with their instructor counterparts in the other engineering departments that provide students for project design teams.
- **Synchronize Section Hours.** Each semester there are designated lesson periods when students receive compensatory time and do not meet for discussion, presentation, or lecture during the scheduled class time. If departments and instructors coordinate section hours so that the senior design courses in each engineering department are taught during the same hours, student project teams can easily meet together as an entire team during that time. For example, West Point has a “Day-1/Day-2” schedule. The third hour on the Day-1 schedule has been reserved by several departments for scheduling the senior design courses. Students appreciate being able to meet together as an entire team without having to figure out a mutually agreeable time.
- **Coordinate Briefing Schedules.** If every department’s senior design course requires two to four presentations each semester and there are two to five separate disciplines working on a single research project, interdisciplinary project teams can quickly be “briefed to death” if instructors do not take an active role to coordinate and overlap briefing requirements whenever possible. It is unlikely that all course schedules can be synchronized, but any overlap will be welcomed by students.
- **Attend Presentations.** If separate, discipline-specific presentations are given during the semester, require one student representative from each project discipline at every presentation. For example, require a computer science team member to attend the mechanical engineering student briefing and vice versa. This helps students keep the entire project in mind and also encourages students to work together.
- **Require Weekly Reports.** Each student project team submits a one-page project summary report to their instructor and project advisor by noon every Friday. One section of the report requires students to report how often during the past week they met with team members from the other disciplines working on their project. The reports serve as a weekly reminder to student teams, project advisors, and instructors.
- **Encourage Advisors.** Project advisors can have a tremendous influence encouraging students from different disciplines to work together. There are several things that project advisors can do to help students learn to work together – such as requiring all team members to attend scheduled project meetings, instead of meeting with each disciplinary sub-team separately.

- **Require a Research Paper.** The student research paper, outlined earlier, requires students to work together as a team to define and present their project in terms that readers can understand and appreciate. It also helps students to better understand the contributions, successes, and challenges that each of the participating engineering disciplines faced during the completion of their project.

Successes

Each multidisciplinary student project is unique, but highly successful multidisciplinary teams often tend to do the same things well. First, they do not hesitate to seek outside expert assistance when they have problems understanding new technology. Often, such assistance involved a short (15-30 minute) tutorial from one of the local subject matter experts (i.e. an instructor within the department) to bring them up to speed on the technology and highlight additional resources for further learning.

Second, highly successful groups do a much better job at grasping the interdependencies associated with their projects. They were more successful at executing tasks in parallel and identifying (and focusing) on critical paths.

Third, instead of focusing on the differences between disciplines and individuals, highly successful teams are able to capitalize on their strengths. Highly successful teams seem to recognize that good ideas are not discipline-specific. Good ideas about computer program structure can come by a mechanical engineering major. Electrical engineers can observe problems in a structural design. Computer scientists can identify system problems on a wiring diagram.

And finally, highly successful groups were better at visualizing the big picture and never losing sight of requirements definition. The realization that perfect solutions are impossible to achieve came much too slowly to many less successful project teams. Teams that learned to live with “good enough” solutions were able to employ time-boxing or other time-oriented approaches to meet deadlines and keep their schedules on track.

Challenges

The computer science senior design project is the largest single project that students complete during their Military Academy educational experience, and, as a result, they find it to be extremely challenging. Multidisciplinary senior design projects provide a host of challenges for both students and faculty. Many of those challenges arise because of differences between academic programs. Some of these challenges are:

- **Different course lengths.** Computer Science students are often teamed with electrical engineering majors, for example. The electrical engineering program has a first-semester informal design overview program and a second-semester formal design project instead of a two-semester senior design project like the computer science program. Electrical engineering students are assigned to projects before their senior design course work formally begins.

- **Different deliverables.** Students know that no two engineering disciplines are the same, so they should not be surprised when they have different deliverables and different formats for reporting deliverables – but they usually are.
- **Different course schedules.** Coordinating presentations, briefings, deliverables, and student meetings can be difficult when multiple disciplines are involved on a single senior design project.
- **Different graded events.** Differences between courses extends to differences between the graded events in those courses. It can be frustrating for students when uniform work is not rewarded uniformly either because of different grading standards or differences in point allocation between departments.
- **Discipline Deadlock.** Without careful planning, multidisciplinary project teams can find themselves in a deadlock situation with none of the discipline-specific subteams being able to complete their portion of the project until other discipline subteams have completed their portions. The computer science majors on an autonomous vehicle project, for example, may believe that they cannot finish writing the software controlling the turning of the vehicle until the electrical engineers have finished wiring the vehicle. The electrical engineering majors believe they cannot finish wiring the vehicle until the mechanical engineering majors finish building the vehicle, but the mechanical engineering majors believe they cannot finish building the vehicle until the computer science majors tell them how fast the software will be able to turn the finished vehicle. As a result progress comes to a screeching halt until students see through the problem or faculty help them break the deadlock they have created.
- **Completion Difficulties.** This has also been referred to as the “90% complete problem.” Student project teams often consider themselves at the ninety percent completion level before they actually are there, but then for numerous reasons they have tremendous difficulty bringing their project to a successful completion.

“How does the deadline look?” asked the boss.

“No problem,” said the engineer. “I’m close to 90 percent complete.”

— Roger S. Pressman³

Lessons Learned

What makes the difference between a successful and a less than successful multidisciplinary senior design project? After studying dozens of senior design projects from recent years, several lessons learned from successful teams have emerged.

- **There is Strength in Diversity.** Students who actively participate on multidisciplinary teams usually learn that applying multiple viewpoints and varied talents to a single problem can result in a more successful solution than if it had been designed and implemented based only on a single view of the situation.

- **Use Outside Advisors.** In addition to having a project advisor (generally equivalent to a project or program manager in the acquisition field), many successful design teams also had the benefit of additional outside advisors who could objectively critique the efforts of the student project teams and provide technical assistance when required. The outside advisors, in most cases, were course directors for other engineering courses.
- **Design Does Matter.** Without exception, teams that developed poor or incomplete designs in the early phases of their projects were forced to revisit the design phase prior to successfully completing their projects. As is to be expected, solid, well-conceived, and detailed designs were implemented quicker and with less effort. The correlation between good designs and early or on-time delivery of finished products is extremely high.
- **Technology Demonstrations.** Early in the implementation phase of each multidisciplinary project, students were forced to demonstrate a prototype that employed portions of the technology that their designs called for using. This helped to identify problems interfacing components early in the design process and allowed teams more time to consult with experts in the field. In previous years, when demonstration of early prototypes and key technologies was not required, teams often expended substantial efforts developing components that were eventually discovered to be incompatible. Late discovery of integration problems often resulted in missed deadlines and/or severely reduced functionality in the teams' final product.
- **Accelerate the Schedule.** One of the keys to a successful two-semester senior design project is to accomplish more than half of the total work during the first semester. Final spring semesters are usually not the most productive academic period for graduating college seniors. The key to a successful spring semester in a senior design project course is to accomplish as much as possible before spring break. The pull of graduation, jobs, spring break, and the desire to be finished with college usually interrupts and derails second semester seniors—even students who have the best of intentions at the beginning of their last semester.
- **“Pull the Plug.”** Software efforts are open-ended by nature; there are no perfect solutions. Software can always be improved. Successful multidisciplinary computer science projects understand the law of diminishing returns and know when to “pull the plug” after they have successfully met client requirements. Successful project teams also avoid scope creep—the gradual expansion of requirements and features.

Conclusion

With compressed schedules, resources, and requirements, academic senior design projects are more sensitive to changes than larger programs and projects. Multidiscipline computer science senior design projects can provide useful insights for students regarding management, design, cost, schedule, and personnel issues on larger projects and programs. This can give students valuable insights into the challenges and opportunities that may be waiting for them in their

future. Senior design projects can also teach valuable leadership lessons to students who fully participate.

Bibliography

1. Dennis, A., Wixom, B.H., and Tegarden, D. *Systems Analysis & Design: An Object-Oriented Approach with UML*. New York: John Wiley & Sons, Inc., 2002.
2. Brooks, F.P., Jr. *The Mythical Manmonth: Essays on Software Engineering*. Boston: Addison-Wesley, 1995.
3. Pressman, R.S., *Software Engineering: A Practitioners Approach*, 5th ed. Boston, MA: McGraw-Hill, 2001, p. 166.

KENNETH L. ALFORD is a lieutenant colonel (recently selected for promotion for colonel) with 25 years of active duty in the United States Army. He is currently an Assistant Professor and the Information Systems Engineering Program Director at the United States Military Academy, West Point, New York. He earned a Ph.D. from George Mason University and masters degrees from the University of Illinois and the University of Southern California.