AC 2012-4447: USING MINI-PROJECTS TO FOSTER STUDENT COLLABORATION IN MULTI-DISCIPLINARY CAPSTONE DESIGN COURSE

Prof. Richard S. Stansbury, Embry-Riddle Aeronautical University, Daytona Beach

Richard S. Stansbury is an Associate Professor of computer engineering and computer science in the Department of Electrical, Computer, Software, and Systems Engineering at Embry-Riddle Aeronautical University, Daytona Beach, Fla. He teaches courses in artificial intelligence, data structures and algorithms, and the capstone senior design course. His research interests include unmanned aircraft, mobile robotics, and applied artificial intelligence.

Dr. William C. Barott, Embry-Riddle Aeronautical University, Daytona Beach
Dr. Salamah Salamah, Embry-Riddle Aeronautical University, Daytona Beach

©American Society for Engineering Education, 2012
Using Mini-Projects to Foster Student Collaboration in Multi-Disciplinary Capstone Design Course

Abstract:

Multidisciplinary Capstone courses are becoming ever more attractive for departments across the nation. Such courses allow students to be introduced to concepts in different disciplines and give them the chance to apply their skills within real world sittings. However, these courses also present both students and faculty with certain challenges, such as unfamiliarity of students with concepts of other disciplines, and the lack of students’ awareness of the hands-on nature of capstone courses and the expectations associated with these courses. Also, the unfamiliarity of instructors with the abilities, work ethics, and personal issues of the current cohort of students within the class makes it harder to assign students to teams.

This paper introduces the notion of Mini-Projects within a capstone course as applied for students in the Department of Electrical, Computer, Software, and Systems Engineering at Embry-Riddle Aeronautical University. Mini-Projects are small projects that aim at introducing students to basic concepts of different engineering disciplines, while at the same time provide instructors with enough information about the students in the class in order to construct teams with higher potential for success.

Introduction:

With a strong emphasis in multi-disciplinary teaming and projects by accreditation organizations such as ABET, and the need from industry to hire students with such a background, the two semester capstone design sequence at the author’s department has combined students from four degree programs: computer engineering, electrical engineering, software engineering, and computer science. At Embry-Riddle Aeronautical University (ERAU) in Daytona Beach, FL, this course is co-instructed by three faculty members from the Department of Electrical, Computer, Software, and Systems Engineering: one software engineering, one computer engineering, and one electrical engineering. While this approach provides the students with the desired experience in working in multi-disciplinary teams, it also comes with its share of challenges for both faculty and students.

For the students, the challenges include a lack of familiarity of concepts, terminologies, tools, and methodologies across disciplines. For example, students in electrical engineering may have very little exposure to some of the engineering processes more common with software and systems engineering. Similarly, software engineers have little exposure with electrical engineering concepts or working with physical hardware. A second major challenge is that students are unaware of the expectations and structure of the capstone project environment. For the faculty, each academic year introduces a new cohort of students. It is difficult for faculty members to know the strengths, weaknesses, overall technical competency, interpersonal issues, etc. of all of the students in the class. This makes it difficult to assign the students to teams because they are unaware of each student’s level of knowledge, work ethic, and leadership.
To address these challenges, the authors have introduced the notion of mini-projects to their capstone design course sequence. A mini-project is a small project that is designed for a small group of students (4 to 5) working on a task for about two weeks. Each team is made up of students from all three of the engineering disciplines. The three mini-projects include elements of computer engineering, software engineering, and electrical engineering, which allows each student to demonstrate their unique skills to the problem at hand. The students are expected to follow the agile design methodologies\textsuperscript{1, 2, and 3}, which are utilized throughout the semester. At the end of the project, written deliverables and a demonstration are provided by the students.

In this paper, the authors highlight their experiences using mini-projects for their capstone design course. Each of the mini-projects will be presented in detail. The authors shall discuss which elements of the mini-project worked, and which did not. The paper also includes student feedback regarding the process derived from student surveys and their peer evaluations. Finally, suggestions are made regarding how these projects would be used for future academic years.

**Background**

Prior to the development of a multidisciplinary capstone for the ECSSE Department, each discipline has addressed the senior design experience differently. For the past 12 years, computer engineering and software engineering have had a common senior design. Their projects have primarily focused upon autonomous systems such as ground robots and unmanned aircraft. For this course, the students have followed an agile design process based upon the Crystal Clear and Crystal Yellow software design processes defined by Alistair Cockburn\textsuperscript{2, 3}. The course has involved requirements, preliminary design, and then a sequence of two week iterations in which detailed design, implementation, integration, and testing were all performed.

During the 2010-2011 academic year, the computer and software engineering capstone design project included a pair of mini-projects to introduce students to the agile process. The first mini-project involved creating a client-server software system in which data could be written on one computer and displayed on another. For the second mini-project, the students had to program a microcontroller to communicate remotely with a computer terminal. A graphical user interface on the terminal allowed the user to send commands to the microcontroller to enable/disable an LED, command a servo, or adjust a pulse width modulated signal. The students were responsible for all hardware and software development to achieve these goals. The results of this first attempt at a mini-project were received quite favorably. The entire process took about three weeks out of the course’s schedule.

Electrical engineering has also held a project oriented multidisciplinary model. However, its students often collaborated in the past with the mechanical engineering department. The course instructors played a greater role as mentors and advisors of a small team of students working with a larger multidisciplinary team rather than a formal instructor. Representative projects supported existing College of Engineering and/or Mechanical Engineering Department projects including, but not limited to the SAE Formula Hybrid team and the EcoCAR project. The Systems Engineering capstone is comprised of students majoring in the Systems Engineering Track of the Bachelor’s of Electrical Engineering program. These students have...
worked on capstone projects independently under the advisement of a Systems Engineering faculty member.

During the Spring 2010, the Electrical Engineering Department and the Computer and Software Engineering Department at ERAU were merged into a single new department as described above. Some of the students in computer engineering and electrical engineering had shared common courses and were familiar with the methods and methodologies of each discipline. However, students from all three four areas (computer, software, electrical, and systems) had never collaborated on a large scale student project. As a result, preparations began to determine how to merge the capstones into a single capstone experience that breaks down the departmental boundaries and sets the standard for multidisciplinary capstones in the college. The new course is based upon the computer and software engineering course’s model, but structured with three instructors: one computer engineering, one software engineering, and one electrical engineering.

Supporting the merged capstone design course, the mini-projects were continued and expanded to address several new objectives. First, the mini-projects were continued to be used to introduce the students to the agile design process by allowing them to complete several low-risk activities following this process. Second, the mini-projects allowed students to work on multidisciplinary teams. Third, the mini-projects are structured so that each project focused upon a particular discipline’s skill set (software, electrical, and computer engineering), while building upon the previous project. This was meant to encourage students to see how other disciplines work, the particular toolset of students in those disciplines, etc. Fourth, the mini-projects were meant to introduce students to the high-level of expectations placed upon them by the course instructors, and an example of the level of intensity that can be expected as the course progresses.

Agile methodologies are processes used in software engineering, systems engineering, and possibly other engineering disciplines. The Agile Manifesto defined the high-level conceptualization of agile. Developers focus less on heavyweight processes such as spiral or waterfall and more on delivery of quality products to the customer. The teams develop in sprints or iterations with frequent delivery to the customer. The teams often adapt their methodology based upon existing agile methodologies such as Crystal Clear, Extreme Programming, Scrum, etc. At ERAU, the Crystal Clear agile methodology has been utilized with some adaptations for the past four years as part of its capstone senior design course for computer and software engineering students.

The mini-projects were first conceptualized for this course based upon the Process Miniature described by Alistair Cockburn in Agile Software Development. In Cockburn’s review of this concept, there were several examples where one or more aspects of the process were executed by the team in either using a toy problem or canned data so that they could gain experiential knowledge of how the process executes. For example, Cockburn describes a company that he studied using a one week project to take new employees through the development cycle. Similarly, for our capstone design experience, this activity was used, but with the purpose of also teaching the students about the technologies to be used and how the different disciplines can work together.
The mini-projects also borrow upon Cockburn’s concept of the technology spike\(^3\). This technique involves having the team build to some proof of concept to determine the validity of a particular design approach. Each mini-project is designed with the idea of demonstrating some integrated functionality rather than on a specific end-user product.

The typical script of a process miniature is as follows:
1. Students receive a detailed problem description and/or requirements
2. Students break down the problem into a set of tasks.
3. Tasks are distributed to each team member.
4. Students have a standup meeting to plan out development and integration.
5. Students work using side-by-side development to build the solution.
6. Students frequently integrate and test the developed components.
7. Students demonstrate the completed work to the customer who provides feedback.
8. The students have a reflection meeting to identify what process issues were encountered, what process elements were useful and worth keeping, and what possible solutions exist to ensure the team performs better on future iterations.

**Description of Mini-Projects**

The mini-project sequence consists of three consecutive two-week modules. These modules are designed as a guided sequence for the design of a hand-held portable GPS receiver. Each module places emphasis on a unique aspect of the system and each emphasizes one of the three core disciplines represented in the class, including electrical, computer, and software engineering. Each module consists of a compartmentalized task that is simple in-concept, but is framed to require the students to perform external research to accomplish the task.

Each module adds additional functionality to the system. The first module creates a prototype software NMEA decoder, performing real analysis on simulated data in a non-real-time, debuggable environment. The second provides input data with a GPS circuit to show real-time analysis on real data using a laptop. The third replaces the laptop with a microcontroller, rendering the device portable and hand-held.

**Mini-project One**

The first mini-project module tasks the students with writing a software program to parse NMEA 0183\(^6\) strings containing GPS messages. This is a software-oriented task, and favors the skills of the software engineers and computer science students. Students are instructed to obtain specific information from the GPS messages, stored in an input ascii text file, and provide specific outputs, written to an output ascii text file. The groups are required to write their program in the standard C language.

Several elements of this project required external research. First, students were provided guidance on the NMEA messages and example messages, but were not given the definitions explicitly, so had to look them up. Second, output formats were not always in the input format, so students must determine the appropriate conversions. For example, the programs must convert between DMS and decimal degree formats, and display time in TAI format rather than
UTC. The formats were specified by the instructors, but the meanings and conversions must be discovered independently by the students.

In the spirit of engineering for safety and reliability, important concepts in engineering and especially in the aerospace industry that ERAU serves, students were informed that instructor test messages would contain deliberate errors. Students were instructed that errors must be detected and that no data error could cause the program to crash. This introduced an additional layer of the module in which the students must anticipate and detect potential errors (including bad CRCs, gibberish data, and maliciously-designed messages containing impossible data with good CRCs).

**Mini-Project Two**

The second mini-project module tasks the students with designing and assembling a physical GPS circuit and utilizing the software from the first module to decode messages from this GPS in real-time. This is an electrically-focused module and favors the skills of the electrical engineering students. However, software updates are required so that the program reads from the computer’s serial port.

Teams are provided with a kit of parts and given a short lecture outlining the block diagram of the intended circuit. Major elements of the circuit include a surface-mount GPS module (which must be soldered by the students to a DIP adapter board), power regulation to provide 3.3V from the battery assembly, level translation to translate the 3.3V GPS logic to the 12V RS-232 logic, and pulse-extender circuitry to render the short 1PPS pulse visible to the eye on an LED. Independent research is required to obtain the data sheets for the individual components and then render a detailed (component-level) block diagram prior to assembly.

At the conclusion of the second module, students must demonstrate the real-time acquisition and decoding of GPS data. Typically, this includes taking the circuit outside with a laptop computer. GPS lock is shown by the activity of the 1PPS timing signal on the hardware module. The laptop interface must render the correct position and time.

**Mini-Project Three**

The final mini-project module tasks the students with replacing the laptop computer of their system with a functional microcontroller and LCD to create a hand-held unit. This project emphasizes the skills of computer engineers in creating embedded systems utilizing microcontrollers, and provides opportunities for software engineers and electrical engineers as well. Each team is provided an Arduino microcontroller and serially-driven LCD. The microcontroller is based on a 5V bus, necessitating the preservation of the multiple-bus and level-translation design. Students are expected to re-use the hardware GPS from module two, and to re-use C-code from module one, to the greatest extent possible.

Like the previous modules, this project requires external research in determining how to address the LCD and how to port code to the microcontroller. The LCD selected by the instructors deliberately had fewer character spaces than the amount of data that is required to be displayed to
the user. This choice forced the students to design their display to show the most relevant information in a user-friendly way, and provide a method to cycle through different screens of data.

This module ends with a demonstration of the functional system consisting of data input (the GPS), processing (the microcontroller), and user output (the LCD). Students are expected to demonstrate a working system, including video of the system in operation. The system is tested for resilience to faults by inducing faults in the serial connection between the GPS and the microcontroller (by, for example, intermittently disconnecting the data line). As in the previous modules, the system must be robust to data errors.

Feedback From Fall 2011

A survey was conducted upon completion of the mini-projects to solicit student feedback. At the end-of-term course evaluations, some additional student feedback regarding the mini-projects was provided, which is summarized here. This section also provides feedback from the faculty point-of-view.

Student Feedback - Survey

A survey was created using Blackboard and shared with the students for one week. Participation was voluntary. The majority of the survey was comprised of statements in which the student could respond via a set of options based on a Likert scale (i.e. Strongly Disagree, Disagree, Neither Agree or Disagree, Agree, and Strongly agree). There were also several short answer questions (as noted below).

Survey Questions:

The survey questions are as follows:

Mini-Project #1

MP1-1. Mini-project #1 provided me with a software development experience that will better prepare me for the senior design project. (scale)

MP1-2. Do you have any feedback for Mini-Project #1? (short answer)

Mini-Project #2

MP2-1. I feel that the reflection workshop from mini-project #1 allowed the team to identify its process problems, and formulate reasonable solutions for mini-project #2. (scale)

MP2-2. The solutions formulated from the first reflection workshop helped the team perform better for mini-project #2. (scale)

MP2-3. Mini-project #2 provided me with a hardware development experience that will better prepare me for the senior design project. (scale)

MP2-4. Do you have any additional feedback for mini-project #2? (short answer)

Mini-Project #3
MP3-1. I feel that the reflection workshop from mini-project #2 allowed the team to identify its process problems, and formulate reasonable solutions for mini-project #3. (scale)

MP3-2. The solutions formulated from the second reflection workshop helped the team perform better for mini-project #3. (scale)

MP3-3. Mini-project #3 provided me with hardware AND software development experience that will better prepare me for the senior design project. (scale)

MP3-4. Do you have any additional feedback for mini-project #3? (short answer)

Mini-Projects Overall
O-1. From the mini-projects, I feel that I am better prepared to work in a team in the ECSSE senior design class. (scale)

O-2. From the mini-projects, I feel that I have a better understanding of the faculty’s expectations for the students in the ECSSE senior design course. (scale)

O-3. I believe that my experience on the mini-project will add value toward my work in the senior design course. (scale)

O-4. I learned something(s) from these mini-projects about my discipline of study. (scale)

O-5. I learned something(s) from these mini-projects about other disciplines of study. (scale)

O-6. I learned how to better communicate with people from other disciplines after completing these mini-projects. (scale)

O-7. Any additional feedback? (short answer)

Survey Results for Likert-scale questions

Figure 1 captures the results of the survey for each of the questions that utilized the Likert scale. The stacked bar chart shows the total response for each choice for each question. The options are color coded and shown in the legend. The survey was completed by 27 out of 28 students. Note for mini-project #3’s questions 1 and 2 one student chose not to respond.
Figure 1: Survey Results for Likert-scale Questions.
**Additional Student Feedback on Mini-Project #1 (representative comments, spelling and grammar corrected)**

- Good C refresher and I can now add basic GPS systems to my skills set.
- The mini projects helped me understand C better, however I still feel that mini project one should have been two weeks long at minimum for the learning curve of C.
- The timeline was a bit tight. But as I've said before, by planning ahead and starting work early, it was a reasonable assignment.
- I think a longer timeline would have allowed us to get together better to deliver a better project. It was difficult to learn a language that's new for everyone and learn our teammates' strengths and weaknesses.
- It took much longer than the time given. Perhaps two weeks would be a better option for the future.
- Due to being on a new team, working with a language many people had not used before, it would have been beneficial to have a little more time. I was not competent enough in C to provide a lot of help with the programming aspect of MP1.
- I think limiting the project to C was a little unfair since it was moving to an Arduino which supports C++.

**Additional Student Feedback on Mini-Project #2 (representative comments)**

- Only issue was with the hardware problems. I definitely understand why more expensive GPSs were purchased though since it was only for mini-projects.
- Just one thing: we really weren’t prepared for the soldering of the GPS units, and as a result many teams struggled through the rest of the mini-project.
- Even in the student who should have known how to solder correctly there were too many mistakes resulting in a lot of burnt GPS. We could have chosen 1 person in each group (who know how to solder) and put them together for a little training and to help each other doing all the soldering.
- For the second mini project I learned some hands on hardware basics that I as a SE never had the chance to experience.
- This project is a little easier than project1. We finished this project on time, and we really did a lot effort. Once we took a team meeting till 3am. Also, I think to do mini-project2 before mini-project1 will be better for our team.
- As far as hardware diagrams go, we were never taught in any other class what design “best practices” were. Perhaps a helpful resource would be a document or website that we could refer to for that. Also, it would be nice for some recommendations for programs to use.

**Additional Student Feedback on Mini-Project #3 (representative comments)**

- Combining hardware and software in the same development was a really good experience that we don't usually have. Like we could see, we need a good coordination between the two subteams to be able to finish (some team couldn't test their program because of hardware problem)
• I liked that we had extra time to work on Mini-project 3. It really allowed us to make a better product in the end to deliver to the customers.
• The development on a representative microcontroller was a good experience. I hadn't used an Arduino before and I greatly enjoyed the opportunity to develop for it. I feel that this mini-project should be kept.
• We finish this project some days before the due date. We can see that during the process of these three programs, we did better and better.

Additional Student Feedback on Mini-Projects Overall (representative comments)

• Overall the projects were not that bad, looking back that is.
• Overall the experience of the mini projects was enjoyable!
• I think the Mini project let me know the teamwork better!
• Overall, the goal of the projects was good. However, there were a few aspects outside of the control of the students, such as the bad chips, that significantly hurt many of the teams' progress. The time distributions were not as well spaced as they could have been: the first project needed a little more time, whereas the third did not need as much as was given.
• All the questions would have been answered with the best score if I myself did the entire project alone. I know budget is a problem, but have a team of 5 people which each team member having to complete the whole project from start to finish. This way everyone will understand a little bit about every discipline, and it could still be a team effort if the team grade for the projects was based off of everyone completing their own mini project in that team. (e.g. If i don't understand something about circuit design, I ask the EE student to help me, and he would because his grade depended on it)
• I really enjoyed the Mini-Projects. I thought they were a good introduction to the course and were fun to do. The course has been a little subjective and I am concerned about the major project because I have no idea if what we have done for our mini-projects is good. We haven't received any grades, so I'm not sure if I should go about the major project the same way. I am also concerned because the grades seem quite subjective. I am not sure what or how I am going to be graded on.
• I would say do these mini-projects again for the next senior design class. Definitely a good intro to the course and other disciplines.
• No, I really didn't think they were all that bad. Looking forward to the rest of the year.
• I think our team is the best team. We help each other and work on our projects together. The guys who are good at programming help me a lot and I really feel I improved a lot during this process. I am more good at the hardware part than the software part. In the following project, I think I can do better and better.

Analysis of Survey Results

From the survey results, several conclusions can be made. First, across all of the mini-projects a majority of students responded favorably (“Agree” or “Strongly Agree”). Despite many misgivings on the workload or various challenges, the students seemed to acknowledge that the mini-projects met their goals of introducing the students to the project. No student strongly disagreed with any of the questions.
Questions MP1-1, MP2-3, and MP3-3 assess the ability to teach the desired skills for the project (software development, hardware development, and embedded systems development). In all three cases, the majority of the students agreed or strongly agreed that the mini-project support their improvement in these areas such that they were more prepared for the course’s design project.

When assessing the mini-projects as a whole, a noteworthy result appeared from the survey. A large number of students (14 out of 28) strongly agreed that the project better explained the faculty’s expectations for student performance in the course. This is one of the major goals of the mini-project.

The additional feedback did provide some noteworthy statements about each mini-project. Students were not given enough time on mini-project #1 due to a lack of familiarity with the C programming language. The learning curve of the language and working with a team was too much. For the second mini-project, the students seem to desire a better primer to hardware assembly. The second project’s comments also revealed an overwhelmingly large frustration toward the components selected, which unfortunately were prone to failure. The third project had significantly more positive feedback with an emphasis on the students appreciating that the Arduino microcontroller was used, and that additional time was given. When assessing the mini-projects overall, despite voicing a lot of frustration to the faculty during the process, a large number commented that they were overall “not that bad” in hindsight.

Student Feedback – Course Evaluations

Additional feedback was received on the mini-projects during the end-of-term course evaluations. These anonymous comments highlight some additional opinions regarding the mini-projects given the context of the remaining half of the semester. Note: these comments were written by the students and the typographic errors or their own.

**Question: What elements in the course MOST helped you learn the course content?**
- The miniprojects were a good way to learn how to handle a software/hardware project in a short period of time.
- The mini projects were a great way to see how people operated under pressure, what they were capable of, and what roles they might be best suited for.
- The mini project let me know about NEMA standard.
- The mini projects we did at the beginning were fun and it was great to work with people outside the norm.

**Question: What elements in the course LEAST helped you learn the course content?**
- We could have done with a little refresher on process before the mini-projects started, as most of us probably lacked the discipline to truly apply what we learned in SE-300.
- The timeline for some of the mini-projects at the beginning could be adjusted better. It would have been nicer to have more time to work on the requirements document.
- Mini-projects seemed to completely derail progress on the actual project, and seemed to be a time filler while research was still being done on what the project was going to be.
Additionally, shoddy equipment and a lack of pre-project research regarding the mini-
projects on the part of the instructors got the entire course off to a bad start.
- The mini projects took a lot of time away from the actual project which has caused the
  work to suffer.
- Miniprojects were good but not worth the time they took away from main project.
  Definitely would of liked that time to work on major aspects like SRS and SDS.
- Miniprojects: They were a lot of tedious work. While there were some lessons learned
  such as a refresher in Software design process and hardware design, I felt that entirely too
  much focus was placed on them. We spent 8 weeks on miniprojects and at the end of all
  that, they were only worth 5% of our grade. That 8 week check we lost from working on
  the main project is severely hurting us in the class right now as we have only had 8 weeks
  to create an entire SRS and SDS along with all the other deliverables that make up the
  other 95% of our grade for the semester. I feel like all of the miniproject lessons could
  have been conveyed in
- The mini-projects were helpful, until we realized how much we needed to do for the final
  project, and then they weren’t so helpful after being 2 months behind on our project.

**Question:** If you could make changes to the course to help you better learn the course content,
then what would they be?
- I think it would be better if the instructor can change the order of mini-project1 and mini-
  project2.
- Never, ever do the mini-projects again. They seemed fine, and I said I enjoyed them in
  the survey, until we realized that they took 8 weeks of our term, but the expectations on
  us were exactly the same is if they had not done them.
- I would shorten up the mini-project, maybe just have 2 and 3
- Drop MPs
- Don't waste as much time on mini projects.
- Fewer miniprojects. Spend more time working on feasibility study, requirements and
  design. Don't waste time on projects that have no bearing on the final project or
  implementation.
- Remove or shorten miniprojects.
- Make the miniprojects shorter (like 3 weeks at most) and only have one miniproject that
  conveys all of the lessons you want to impart.
- No mini-projects.

**Summary and Lessons learned**

As course instructors we believe that the Mini-Projects did serve a great purpose in identifying
students’ strength and weaknesses, which helped us in constructing the teams for the overall
capstone project. It also gave us a chance to get introduced to the different personalities and
Teaching styles among the three instructors. This was significant because this was our first
experience co-teaching a course together.

A major issue with the Mini-Projects was the scheduling of three of such projects. In hindsight,
this seems to be a little exaggerated and it might have been better if we combined projects one
and two together as one. This would have saved some class time and allowed the students more
time to work on the main capstone project.

The weighting of the mini-projects in the final course grade should also be reconsidered. At 5%,
the faculty’s goal was to make it count for something, but be low enough that the students would
not worry if they struggled. Instead, given the workload and short time frame, some students
appeared to care less for success because it was not worth the final payoff in terms of grade.

For future capstone courses, we also intend on making the connection between the Mini-Projects
and main project clearer. One of the complaints from students was that they spent eight weeks
working on these projects that served little purpose in the context of the overall project. While
this assessment might be debatable, it is clear that a more obvious motivation behind the
Mini_Projects and a clearer connection to the main project are needed. In addition, we realize
that sufficient time for must be allowed for completing each of the Mini-Projects when new
concepts are introduced (i.e. C programming) which are new to the majority of the students in
the class. Finally, extra care should be taken in selecting the required hardware material for the
students to work. Having the students work with defected parts leads to frustration and could
taint the whole experience.

We believe that this has been a valuable experience for both students and faculty. We will
continue to make use of Mini-Projects in future capstone courses, while keeping in mind the
lessons we learned as described above.

References:

1 Richard S. Stansbury, Massood Towhidnejad, Jayson F. Clifford, and Michael P. Dop. “Agile Methodologies for
Hardware/Software Teams for a Capstone Design Course: Lessons Learned,” ASEE Annual Conference 2011,
Vancouver, BC, June 27, 2011.

Wesley Professional, 2006.

3 Cockburn, A. Crystal Clear: A Human-Powered Methodology for Small Teams. Upper Saddle River, NJ: Addison-
Wesley Professional, 2004


5 Agile Software Alliance, “Principles behind the Agile Manifesto.” Online at:

6 SiRF Technology “NMEA Reference Manual.” Online at: