Preparing Students for a Capstone Design Course

Lisa Burnell Ball
l.ball@tcu.edu

Antonio Sanchez
a.sanchez-aguilar@tcu.edu

Donnell Payne
d.payne@tcu.edu

Department of Computer Science
Texas Christian University
Fort Worth, Texas, 76129, U.S.A.

Abstract

Students in computer science and information technology can be surprised and overwhelmed by the difficulty of their capstone design project. To bridge the gap from small team, course-topic centric activities to capstone design projects, we recently experimented with what students dubbed the "mini-capstone project". A local university contacted our department with an urgent request. They needed software to provide logistics support for a symposium that introduces young women to careers in science, technology, engineering, and mathematics (STEM) fields. Their existing software had been developed by a capstone team at our university ten years ago but no longer functioned in their new computing environment. Once this need presented itself, a suitable junior level course was modified to support the project. In this paper we describe the course project, how the course was modified to support it, and student responses.

Keywords: capstone projects, integrating computing skills, database course assignments, case study.

1. INTRODUCTION

Capstone design projects in Computer Science (CS) and Information Technology (IT) are purposefully challenging, requiring student teams to develop a significant real-world product. Students must learn how to specify requirements, design and implement software systems, perform detailed testing, and deliver a working product. In the course of doing so, students must learn many new skills including evaluating new technologies (languages, frameworks, hardware, etc.), integrating existing knowledge and skills, and, perhaps most importantly, working with customers and each other. The starting point is often vaguely defined and requirements are fluid as the system evolves. In the final project presentations, almost every team reports some degree of surprise at the difficulty of their task.

Team projects are widely accepted as an effective educational tool. Students learn how to work in teams of course. They learn technical skills from one another, and gain experience in the soft-skills of collaboration, writing, and oral communication. Our industry advisory board tells us that all these are essential abilities of employees. Our department uses team projects in several upper-division courses. These typically are two or three person projects with clearly defined requirements and are small enough to be completed in two to three weeks. Some of these require student presentations. While valuable, they may not offer enough of a stepping stone to the capstone project.

An interesting alternative viewpoint exists regarding capstone projects. Some have argued that traditional capstone design projects are not that beneficial. Wright (2010) claims traditional capstones have failed. In their place, multiple internships developing many skill-sets, including system administration, help desk support, and programming are recommended. In essence, this means replacing capstones with apprenticeships. This is an interesting argument and may be true, but may be unfeasible at most schools. What company wants to work with a student with a 2.5 GPA? Our experience is that these students do get jobs, but not ones that require the breadth of experience suggested. Moreover, these skills (and some others) need a significant amount of oversight to ensure accomplishment of learning objectives. Faculty members need to be the ones that invest this effort.

Walker and Slotterbeck (2006) also have explored an alternative to a traditional capstone design project. They use what they call "Integrated Research Components" (IRC) in which each student takes two related courses that together can be used to satisfy the requirement for a capstone experience. These courses are designated as having an "IRC component". Projects are related to the selected courses thus constraining the choice of research topics. "Every student participates, though the balance between research and application varies from student to student and project to project". For example, a game was developed in an Artificial Intelligence course and the graphics component was added in the Computer Graphics course. Another example could be a research experience in which work is performed in a networks course and continued in a security course. At the time of their publication, they had studied their approach over a five year period. Unlike the previous recommendation, smaller institutions with limited resources can try this variation of student research projects, with reasonable demands on faculty time.

Whether it is in preparation for a capstone project or an apprenticeship, we believe that the approach described herein helps prepare students for industry. Students need exposure to the learning objectives we describe. For us, the Database Systems course offers a suitable course in which to apply our "mini-capstone project" approach.

To bridge the gap from small team activities to the capstone project, we recently experimented with what the students dubbed "mini-capstone design project". In this paper we describe the project and student responses.

Our department offers two degrees: a traditional ABET-accredited Computer Science (CS) degree and Computer Information Technology (CIT) degree, the later geared for those students interested in directly entering the workforce as system administrators, web developers, and the like. Both majors are required to take a junior-level course in Database Systems. This course covers both theory (e.g. relational algebra, functional dependencies) and practice (e.g. database design, SQL). Prerequisites for all students include two semesters of programming, data structures, and one semester of discrete mathematics. In their senior year, both CIT and CS majors enroll in the same capstone projects course. Thus it is in the Database Systems course in which some new pedagogical approaches for all students can be explored.

2. STUDENT LEARNING OUTCOMES

Several of the course objectives fit nicely with the incorporation of a software development project (Table 1). Keep in mind that these are not all the course objectives; they are ones that can be met in part by the project we describe. Additional homework assignments, in-class practice, and exams further evaluate student mastery.

Construct and evaluate conceptual designs using the Entity Relationship (ER) model.
Map an ER design into relational tables and evaluate a mapping.
Define tables, constraints, and triggers in SQL.
Construct and understand queries expressed in SQL.
Write a program that interfaces with an SQL database.

Table 1. Database Systems course objectives that can be satisfied through a team project

To incorporate the desired learning to prepare students for their design course, additional objectives are shown in Table 2. As an example of an objective in Table 2, teams were allowed to select the programming language from among Java, C++, and Python. Technical presentations were a good way for students to get feedback and for other teams to see alternative solutions to a problem. Observing the behavior of an existing production system is a skill students need. Anecdotally, the authors and industry colleagues report the necessity of this analytical ability (often source code is available, but the program is hundreds of thousands of lines long and impractical to study in detail). One example from the defense industry was some engineering thermal analysis software for which modifications were required but not all software could be retrieved because it was classified.

Make software development choices with minimal instructor direction.
Become familiar with a real-world application (domain analysis and requirements).
Validate software.
Conduct technical presentations to the class and instructor.
Visit customer site for installation and training

Table 2. Newly added Database Systems course objectives to be satisfied via the team project.

As expected, time allotted for topics traditionally covered must be reduced. Given changing technologies and coverage in other courses, we chose to reduce time spent on traditional disk-based processing and optimization. In the Elmasri & Navathe (2011) textbook we use, this meant reducing coverage in some chapters or sections without loss of continuity. Query optimization lectures provided to the students replaced the more detailed coverage in the text.

3. AN IMPORTANT SOCIAL ISSUE PROJECT

Studies show that exposing middle school girls to STEM fields impacts their interest, ability, understanding of STEM careers. (Valla, 2012), (Demetry, et al.,2009), (Wyss, Heulskamp, & Siebert, 2012). Expanding Your Horizons (EYH) is among a number of national initiatives to increase the number of female professionals working in STEM professions via introduction to current career women. This need has been expressed at the highest levels of U.S. government:

> *President Obama understands that increasing the number of women engaged in science, technology, engineering, and math (STEM) fields is critical to our Nation's ability to out-build, out-educate, and out-innovate future competitors. Jump-starting girls' interest in STEM subjects, [...] and giving greater prominence to strong role models is not just the right thing to do, but the smart thing to do.*
>
> Executive Office of the President (2013)

The mission statement of The Expanding Your Horizons (EYH) Network is "inspiring girls to recognize their potential and pursue opportunities in science, technology, engineering and mathematics." (EYH, 2012). Middle and high school girls take part in local programs in which they participate in hands-on activities led by professional women in STEM fields. Example disciplines include environmental engineering, computer science, and chemistry. In the annual program held at a nearby university, approximately 400 middle school girls and 40 professional women participate.

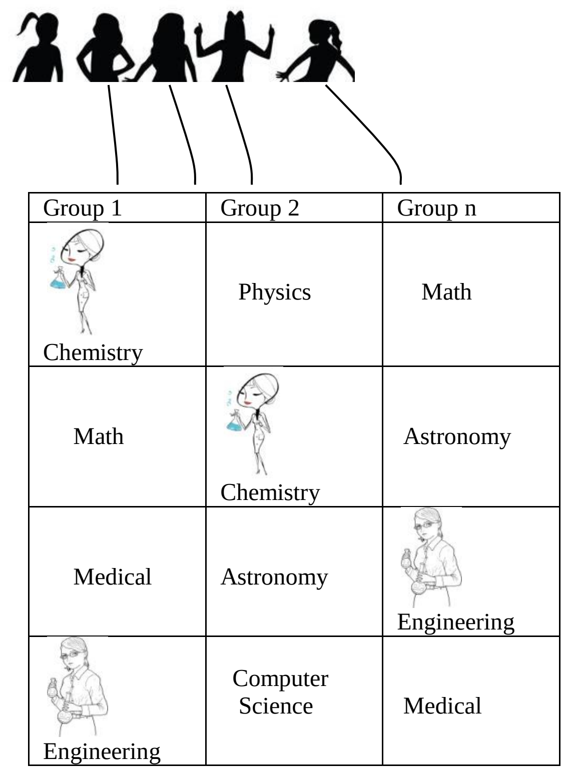

Fig. 1. EYH scheduling.

Trying to schedule this many participants in a manner that meets a number of constraints is a challenging task. Here, we give a very brief description. Students choose a first and second topic of interest. Groups are formed in which a given group will attend four sessions each by a different presenter in a unique discipline. A goal is to assign a student to a group in which her first or second choice is among the sessions. Further constraints are to have roughly 20 students per group (additional presenters are recruited if needed), and to limit the maximum number of students in a group that are from the same school. Figure 1 outlines the concept.

4. STUDENT PROJECT DESCRIPTION

As with the rest of the production system, students could run the executable to observe the user interfaces (the source code was no longer available). For the course objectives, this was useful in informing students how they might choose to implement their tables, define data constraints, and update the database. Moreover, they could see an example interface with which the client was happy. This provided some structure so students did not need to feel lost in a sea of decisions. Figure 2 shows an example screen for entering, changing, and deleting EYH student data.

After class discussion and demonstrations, the project was summarized for the students in the first assignment:

> *A local university needs a new version of their existing system that is used for planning the annual EYH event. The final result of your project will be a new software program that will replace the existing version. The functionality should match that of the current system (you can discuss possible improvements with me). We want to make the transition as smooth as possible. This is a real project upon which the EYH event coordinators depend. You may be asked to accompany me on this and future assignments to their university for demonstrations, training, and evaluation.*

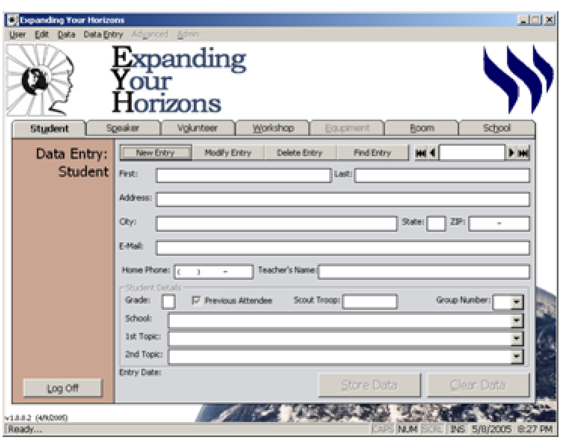

Fig. 2. Data entry screen for student data.

An abridged version of the first assignment is shown below. It meets several course objectives, including embedded SQL programming, technical writing, and domain analysis.

Requirements & tasks for assignment 1:

1. Get familiar with the operation of the existing version of the software.
2. Develop an ER model and map it to relational tables (with constraints)
3. Write software to allow the customer to manage <u>student information.</u>
    a. Enter, update, and delete data
    b. Prepare reports to view the data according to customer-specified requests.
4. Write user and developer guides, following the format provided to you.
5. Present to the class

Example reports include listing students alphabetically, listing students from one school, and listing students by 1st or 2nd topic choice (see the existing documentation/code for all reports not related to scheduling girls into groups or assigning presenters).

Additional Requirements (Constraints)

1. The new system needs to run on Windows 8. Use SQLite for the DBMS. For code, you may choose from C++, Java, or Python. Keep in mind that this will be used for many years and should need little to no maintenance.
2. The new system should be easy to install and operate. To make the transition as easy as possible, it should operate like the existing system. Minimize the amount of free text as much as possible in order to reduce data entry and scheduling errors (e.g. enter valid schools into a separate table so you can use a menu on the data entry screen for students).
3. The system should allow the database to be backed up easily.

Additional Notes

In this assignment, you are not doing any scheduling (e.g., assigning students to groups). You will be doing this in the next homework. Please keep this in mind as you are completing this assignment.

An abridged version of part 2 is given below. The 2 assignments together meet the objectives stated in Tables 1 and 2. Students are provided with a short outline of the automated scheduling algorithm. At a minimum they must fill in the details, but they were strongly encouraged to explore improved algorithms. The metric used was based on the number of students receiving their first or second topic choice.

Requirements & tasks for this assignment: Project Completion

1. Complete the project, in particular the scheduling portion and anything you didn't complete from part 1
2. Allow the customer to manage all student, presenter, and related data
3. Implement automated and manual scheduling
4. Prepare reports to view the data according to customer-specified requests. Examples include listing the schedule, group assignments, and listing students by 1st or 2nd topic choice. You do not have to print the output exactly as given in the existing system, but it should be easy to understand!
5. Present to class and customer
6. Write final user and developer guides.

---

5. EVALUATION OF STUDENT PERFORMANCE

A detailed rubric is difficult to define for this sort of project. As in industry, project and individual assessments include subjective components. First, we focused on the completion of core functionality: did the program store data properly, create a satisfactory schedule, and create accurate reports? Second was the usability of the system. Third was quality of user and developer documentation, with emphasis on user documentation. Fourth was quality of presentation materials and delivery of the presentation (including to the customer). The first criterion is the one in which key learning objectives of this course are satisfied so obviously it had to be worth the most (80%). Assessment here focused most on design, implementation, querying, and updating of the database. A formal rubric for assessing capstone projects, abridged for this smaller project, is planned for future semesters. Estell & Hurtif (2006) describe the use and outcomes of a detailed rubric that could be used as a starting point. For the semester discussed here, most was communicated in class. Examples of graded written work with instructor comments were provided early in semester.

Just as we had to focus on some aspects of a software development project, the same held true for student assessment. Formal peer reviews were not used (as they are in the capstone project). With eleven students in the class, it was easy enough to assess the quality and degree of individual student participation. With a small department such as ours, faculty members have taught earlier courses with these students and so were familiar with their prior performance (e.g., skills, work-ethic).

The project was not part of the initial course syllabus. A new version of the EYH system was urgently needed because changes in the computing environment at the host university rendered their existing system unusable. Because our university developed the original system under the instruction of one of the authors, we were asked for help.

The instructor and students discussed as a group how to modify course weighting (e.g. how much the project was worth versus homework assignments and exams). Private discussion, and finally, anonymous votes resulted in the final weight distribution. Additionally, through a similar negotiation process, the students had the ability to get a good grade even if the 1st deliverable (part 1) did not meet all requirements. This seems unfair at first, but given that the goal was a working project at the end of part 2, those that did not complete part 1 had to do more work in a shorter period of time. While unquantifiable for this group, peer pressure and pride appeared to have motivated a team that had failed to complete several part 1 tasks. Two other teams were engaged in a friendly rivalry, which was fun to see.

The distribution of final course grades for the students were similar to previous semesters in which such a project was not given. This indicates that student grades were not negatively impacted by the course modification.

6. STUDENT RESPONSES

Through conversations with students in other courses, we learned that the project was dubbed the "mini-capstone design project". Student responses were mostly positive. The aggregate formal evaluation scores exceeded the department, college, and university average in 80% of the metrics. Students found the project challenging and worthwhile. Most were surprised at the level of difficulty resulting from developing a nearly complete software project. This, in part, was a desired learning experience to prepare the students for their capstone projects. Some of the most common positive comments from the university's formal student evaluation system include:

- "The EYH Project assignment helped me learn how to efficiently implement a database".
- "Keep up with projects such as EYH."
- "The project taught me how to put together many of the subjects taught in other courses. I expect this will help with capstone course."

Approximately 15% of the students expressed displeasure with either the level of work or that the project was unexpected, as evidence by the following:

- "The project [...] was extremely difficult to manage on top of the other course projects which had been discussed well in advance. Simply put, the project was difficult to complete because of the necessary outside material we had to learn to complete it, on top of the competing projects from our other Computer Science classes"

In response to such a comment, we believe the learning experience emulates those they will likely have once they graduate. Furthermore, we expect responses in their exit interviews (given the semester they graduate) will be positive. Because this is a junior level course, this data will not be available until next spring. As can be the case, one does not realize the benefit of an activity until some time has passed.

7. CONCLUSIONS

Preparing students for a senior capstone design project can be improved by giving students integrated experience involving several of the processes and skills they will need to be successful. We have described an experiment in which the prerequisite Database Systems course can give them this experience. The team project described herein was an initial exploration. While student evaluations were performed in the Database Systems course, the more important indicator of success will be to again evaluate student success once they complete their capstone project at the end of this academic year.

All student teams produced software that met most of the requirements. Teams, accompanied by the instructor, visited the site for installation and demonstration of their test systems. Just as the customer appreciated our efforts, we appreciated the opportunity for our students to get this experience. In the future it would be useful to employ written customer assessment, rather than rely on just verbal discussion. However, it is important not to over-burden the customer. After the teams submitted their final product, the course instructor selected the best system and continued working on a deliverable product.

Students worked hard, learned new skills, and are looking forward to being better prepared for their capstone projects. Provided that assessment of the students' senior capstone performance is positive, we plan to incorporate similar projects in the future.

8. REFERENCES

Demetry, C., Hubelbank, J., Stephanie L. Blaisdell, Sontgerath, S., Nicholson, M. E., Rosenthal, E., & Quinn, P. (2009). Supporting Young Women to Enter Engineering: Long-Term Effects of a Middle School Engineering Outreach Program for Girls. *Journal of Women and Minorities in Science and Engineering*. 15(2):119-142.

Dutson, A. J., Todd, R. H., Magleby, S. P., & Sorensen, C. D. (1997). A Review of Literature on Teaching Engineering Design Through Project-Oriented Capstone Courses. *Journal of Engineering Education*, *86*(1), 17-28.

Elmasri, R. & Navathe, S. B. (2011). Fundamentals of Database Systems (6th Edition). Addison-Wesley Longman Publishing, Inc., Boston, MA, USA.

Estell, J. K. & Hurtif, J. (2006). Using Rubrics for the Assessment of Senior Design Projects. *2006 ASEE Annual Conference Proceedings*. American Society for Engineering Education. Chicago, IL.

EYH. The Expanding Your Horizons Network. (2012). Retrieved June 1, 2015 from http://www.expandingyourhorizons.org/about-us.html.

Executive Office of the President (2013). Women and Girls in Science, Technology, Engineering, and Math (STEM). Retrieved May 3, 2015 from https://www.whitehouse.gov/sites/default/files/microsites/ostp/stem_factsheet_2013_07232013.pdf

Valla. J. (2012). Increase achievement and higher education representation of under-represented groups in science, technology, engineering, and mathematics. A review of current K-12 intervention programs. *Journal of Women and Minorities in Science and Engineering,* 18(1), 25-53.

Walker, E. L. & Slotterbeck, O. A. (2006). Integrated research components: a practical and effective alternative to senior projects. *J. Computing Science in College 22(1).* (October 2006), 72-83.

Wright, K. (2010). Capstone Programming Courses Considered Harmful. *Communications of the ACM, 53*(4), 124-127.

Wyss, V. L., Heulskamp, D., & Siebert, C.J. (2012). Increasing Middle School Student Interest in STEM Careers with Videos of Scientists. *International Journal of Environmental and Science Education.* 7(4), 501-502.