Enabling Scalability, Richer Experiences and ABET-Accreditable Learning Outcomes in Computer Science Capstone Courses Through Inversion of Control

Thomas Bihari, Igor Malkiman, Moez Chaabouni, Joe Bolinger, Jay Ramanathan Rajiv Ramnath, Michael Herold
The Ohio State University, bihari.5@osu.edu, igorm@cse.ohio-state.edu, chaaboun@cse.ohio-state.edu, bolinger@cse.ohio-state.edu, jayram@cse.ohio-state.edu, ramnath@cse.ohio-state.edu, herold@cse.ohio-state.edu

Abstract **- Capstone courses are expected to prepare students for the "real world" by putting them into a microcosm of the real world. In these courses, students are given a problem of some complexity, and are expected to exercise and develop problem-solving skills as they address the problem. Within our Computer Science and Engineering program we have, over the past eight years, successfully scaled up the Capstone courses. Doing so has required innovative thinking about the role of the students, faculty, and project sponsors.**

In this paper, we will present and discuss issues with scaling up the components that have made this program successful. These include housing the courses in an NSF IUCRC that enabled the cultivation of highly-committed industry partners, the creation of strong pre-requisite courses, careful development of faculty resources through the selective hiring and mentoring of clinical faculty, a commitment of the faculty to give up close management and control, strong partnerships with other organizations within the university to provide students greater access to resources, an emphasis on cross-team knowledge sharing and learning, and the development of unique assessment and evaluation tools so as to be able to monitor, measure and fairly assess a wide-spectrum of projects.

Index Terms – Capstone program, Software engineering education, Workshops

INTRODUCTION

Capstone and similar programs [1] [2] are expected to prepare students for the "real world" by putting them into a microcosm of the real world. As with internships [3], in these courses, students are given a problem of some complexity, and are expected to exercise and develop problem-solving skills as they address the problem. The course and its expectations are carefully managed, in order to provide a safe place for students to fail. And it is understood that students cannot really do much in a Capstone course (after all, being able to deal with complexity and open-endedness is a life-long journey).

Or can't they? Within our Computer Science and Engineering program we have, over the years, and, step-by-step, unshackled and greatly scaled up the Capstone courses. Faculty keep a watchful eye, but provide only guidelines, and mentor from a distance. Students work directly with the project sponsors (i.e., the customers), who expect teams to deliver close to commercial-grade products. Students are expected to recognize the essence of situations – both technical and non-technical – and learn more about then. Finally, they are expected to find solutions by adapting the teachings of their previous classes, by developing new capabilities and applying them, and by evaluating their effectiveness and making corrections, all during the course of their projects.

The results have been striking. Beginning with one project with an external sponsor in Spring of 2003, we now execute thirty projects a year. Most of the eighty projects done to date have been taken over by their sponsors after the course and put into production use. Student-created smart-phone applications are available on the Apple App Store and the Microsoft Marketplace. Two projects resulted in products that became commercial successes. Our assessment and evaluation tools show that student-learning outcomes have been comprehensively met.

Successfully building, maintaining, and scaling this program has raised many issues, and has required attention to key success factors. These include housing the courses in an NSF IUCRC that enabled the cultivation of highly-committed industry partners, the creation of strong pre-requisite courses [4], careful development of faculty resources through the selective hiring and mentoring of clinical faculty (see [5] for a discussion on faculty development), a commitment of the faculty to give up close management and control, strong partnerships with other organizations within the university to provide students greater access to resources, an emphasis on cross-team knowledge sharing and learning [6], and the development of unique assessment and evaluation tools so as to be able to

monitor, measure and fairly assess a wide-spectrum of projects.

THE CAPSTONE PROGRAM

The Capstone program consists of a set of project courses. While they differ in content, each course builds on the foundations of its pre-requisite courses and requires the students to complete a project showcasing those skills. The courses are:

- CSE 758: Software Engineering
- CSE 762: Web-Services-Based Distributed Systems
- CSE 772: Information System
- CSE 776: Hardware/Software Interface Design
- CSE 786: Game Design and Development

Each of the Capstone courses has a different focus, and each is run slightly differently. However, the CSE 758 (Software Engineering Project) course is a good example.

RUNNING THE CAPSTONE COURSES EXAMPLE: CSE 758

The main objective of this course is to prepare students for the software engineering profession. The prerequisites for the course are CSE 560, 601, and 757. CSE 560 provides students with a first serious experience in software design and implementation, working in teams, and in producing documentation. CSE 757 provides additional knowledge of software development practices, methods for implementation and maintenance of software, as well as the importance of reliability of software and ways of achieving it. CSE 601 introduces students to the ethical issues in computing and provides practice in developing communication skills. All of this knowledge is called upon in CSE 758.

Learning Outcomes

The learning outcomes for this course are in the process of being revised to better connect with its pre-requisite, CSE 757. A draft set of revised learning outcomes is as follows:

- LO1. Be familiar with frameworks for analyzing the business context of enterprise IT systems, the concept of Business-IT alignment and related issues, and Enterprise Architecture frameworks for analyzing and achieving Business-IT alignment.
- LO2. Master the principles underlying structured and agile software engineering frameworks, specifically structured and agile software engineering methodologies for requirements identification, analysis, architecture, design, deployment, testing, and project management.
- LO3. Be familiar with the application of structured and agile software engineering frameworks, specifically structured and agile software engineering methodologies for requirements identification, analysis, architecture, design, deployment, testing, and project management.
- LO4. Be familiar with the application of at least one industry-standard technology framework.
- LO5. Master professional and formal presentations and communications to a varied set of stakeholders – customers, peers and superiors.

Students in CSE 758 achieve these outcomes by engaging in a quarter-long industry sponsored project and interacting with real customers.

Before the Term Begins

The Course Coordinator is responsible for maintaining a list of sponsors and projects. The projects have been varied, and are usually provided by industry organizations. Recent examples of the projects in the course are:

- an iPhone application for location-aware social networking;
- a touch-screen instructional system for autistic children;
- an iPad application for teaching algebra;
- a disaster recovery content management system;
- A Windows 7 Phone application for teaching math to primary school students.

Several weeks before the term begins, the Course Coordinator meets with the Instructors for the upcoming term's CSE 758 (and other Capstone) sections to review the project list. The list is prioritized by the suitability for the focus of the course (e.g., CSE 762 has a web-services focus), likely student interest (e.g., in the specific technology or business area), sponsor requests (e.g., related to schedule), and general fairness to sponsors.

Based on the number of students enrolled, a subset of the projects is chosen for each section. Typically, if a section has 25 enrollees, we choose about 10 potential projects, of which three or four are "must do" based on the above criteria. Of that list, five to six projects might be executed by five to six teams of four to five students each.

During the Course

The instructor delivers a single, context-setting lecture at the beginning of the quarter. Thereafter, the instructor leads class discussions on various topics that come up during the execution of the projects (e.g., risk management, requirement prioritization).

At the first class, students are presented with the list of projects. They subsequently fill out a Questionnaire in which the individual students note their first, second, and third project choices on the list of potential projects for the quarter. The instructor makes a good attempt to assign the students to projects they prefer. Each team works jointly on all phases of their project.

Students are also made aware of ethical, social and professional considerations -- both in their dealings with the customer sponsors (e.g. having to sign non-disclosure agreements), and because of the characteristics of the applications themselves (e.g. ADA-compliance, privacy laws). The nature of some projects (such as the one concerning the design of a public assistance web site) is such that multi-disciplinary considerations are also part of the overall picture.

The external organizations assign one or more of their personnel to work with the students -- mostly in the provision of requirements, but also as mentors. These sponsors are expected to meet with the student teams frequently (e.g., weekly), on a schedule and in a manner negotiated with students. The sponsors are invited to all presentations made by students, to provide appropriate feedback.

Although the projects vary considerably in scope, the course maintains a consistent flow during the term. Borrowing from the Unified Process [7], the projects pass through several stages during the quarter:

- Inception Stage: Approximate vision, business case, scope, high-level estimates
- Elaboration Stage: Refined vision, iterative implementation of the core architecture, resolution of high risks, requirements identification, more realistic estimates
- Construction Stage: Iterative implementation of remaining features, preparation for deployment
- Transition Stage: Beta test, deployment

Each team is required to make three formal presentations during the quarter to showcase their progress and solicit feedback:

- A "Risk Checkpoint" presentation (30 minutes) in the third week of the term, in which the team describes the project scope, risks, etc., and demonstrates a simple "hello world" application in the chosen technology (e.g., a simple iPhone app, to demonstrate that the team has configures the iPhone development environment successfully). This presentation roughly coincides with the end of the Inception Phase.
- An "Interim" presentation (30 minutes) in the seventh week of the term, in which the team describes the progress to date, and the plans for completing the project. This presentation roughly coincides with the end of the Elaboration Phase.
- A "Final" presentation (two hours) during finals week of the term, in which the team showcases the completed project and discusses plans to transition the deliverables to the sponsor.

The Risk Checkpoint and Interim presentations are made to the entire class, the instructor and the project sponsor and/or an external evaluator. The Final presentation is made to the instructor and the sponsor and/or evaluator because of the difficulty of scheduling times when the entire class can attend. Typically, each student will present at least 4 times during a quarter.

Students also develop a 48" x 36" color poster and participate in a final poster presentation, where they present their work to invited guests from industry and academia. These presentations typically draw 50 guests.

Each week, typically on the first class-day of the week, each team gives a standup report approximately 10 minutes in length. Over the course of the quarter, each member of each team is expected to be the "lead presenter" at one or more weekly stand-ups. The topics are:

- What the team did in the previous week,
- What the team plans to do in the upcoming week,
- What risks or issues ("blockers") the team is experiencing.

If blockers or risks are identified, the entire class brainstorms possible solutions. After all teams have presented, the instructor meets with each team to discuss the progress of each project in detail. As appropriate, the instructor may take a more active role in resolving issues (e.g., by speaking directly to the sponsor).

Early in the term, the student teams choose methodology, including roles, activities, and work products appropriate for their particular project. The specific domains include:

- Project Management
- Requirements and Analysis
- Architecture and Design
- Development and Testing
- Deployment and Transition

The students then maintain an electronic Project Workbook (Repository) containing the chosen artifacts, including, for example a Risk Plan, Story Cards, Test Scripts, Source Code, and other artifacts determined by their methodology. This workbook forms the core of the materials turned over to the sponsor at the end of the project.

Students are also required, at the end of the project, to write an Individual Report describing the work they personally performed, what they learned from their experience with the project, who mentored them, and who they mentored. The students must also write an evaluation of some tool, process, or technology they encountered during the project.

Students are evaluated by a combination of the following:

- Risk Checkpoint, Interim Presentation, and Final Presentation: (team oral presentation, with Powerpoint slides), graded via a standard Oral Rubric.
- In-Class Status Reports: Grading of the weekly stand-ups is by attendance and participation.

- Customer Feedback: The team's sponsor is encouraged to give feedback throughout the quarter. At the end of the quarter, we ask each sponsor for written feedback, as well as a satisfaction score on a scale of 0-5.
- Project Workbook: (team written document), graded by section (see below) on content (quality of artifact and activities that created it) and reasoning (why decisions were made).
- Poster: (informal team oral presentation, with 2' x 3' poster), graded on general quality of poster and presentation.
- Individual Report: (individual written document), graded by level of contribution to team and by quality of writing (generally follows a standard Writing Rubric).

This course is well-received by the students. Students have reported that the course has helped them significantly in their job interviews. Industry sponsors have also generally been pleased with what the students have delivered.

A Successful Variation

In a successful variation of this approach, in a section of CSE 772 (Information System Project), a single, larger project was decomposed into sub-projects and was assigned to the entire class. In that case, the class was organized into four teams of four people each. We had one project to work on. Each team worked on a stand-alone component of the project—and provided a separate workbook that included elements of each of the phases of software development—project management, requirements collection, analysis, user-interface design, architecture, system design, and implementation. Individual workbooks also served as a "portfolio" to showcase each team's capabilities. Teams were selected based on a combination of student preferences and a desire to equalize capabilities based on the responses given in the class Questionnaire. In addition, each team had to designate an "Architect" (usually the lead developer) who was a member of an "Architecture Group" (AG). The AG was responsible for setting the general architectural constraints and specifying the interface points across all teams. This was very necessary for the success of the class and the project(s). It was essentially the glue that held all together.

CRITICAL ENABLERS AND CAPABILITIES

There are a number of requisite characteristics of a program in order to effectively use this technique: an adequate supply of sponsors willing to work within the constraints of the course, instructors with exceptional mentoring skills, and suitable projects for completion within the finite time available in one term. The presence of these characteristics is necessary for this technique to be viable, but it is not sufficient for practicability.

Committed Partners Supplying Sponsors and Projects

Sponsors form the heart of this technique and thus require continuous, active recruitment. We petition our contacts from our NSF Industry-University Collaborative Research Center (IUCRC) for many of our sponsored projects. By asking our industry contacts for well-defined, concrete problems they would be willing to work on with our students, we are able to provide the majority of our groups with industry-sponsored projects.

Not all industry-sponsored projects are suitable for use in the venue of a capstone course. As with all courses, time constraints play a large part in the suitability of a project for the course. Many projects cannot be completed in a single quarter, within a single class. Often, however, the sponsor is able to partition a smaller part of their project out for use in the course. In fact, it is possible for a single sponsor to provide different pieces of a single project in consecutive quarters for use in the course. This type of relationship is beneficial for both the sponsor and the university; the sponsor gets development work done by the students at very small cost and the university gets industry experience for its students.

In addition to this, our center is very multidisciplinary. We work with departments throughout our university on various projects. Graduate students from several departments have acted as sponsors for several capstone projects.

However, merely being willing to serve as a sponsor does not make one an ideal candidate. Setting and having the sponsor commit to clear expectations is key [8]. An ideal sponsor must be willing to hand over development tasks to the students without being over-bearing and unwilling to let the students lead the project. If a sponsor cannot let go of the project enough to allow the students to build their project management skills as well as their programming abilities, the sponsor can severely limit both the students' learning outcomes and their satisfaction with the course.

Maintaining relationships with sponsoring organizations and maintaining a suitable backlog list of projects initially required a significant time commitment from the Course Coordinator – not unlike the "Account Manager" role in a consulting company. However, as the program has developed a successful track record, and has become more well-known, the landscape has shifted. Today, ¾ of the projects now come to the program unsolicited, by word of mouth. Former students who are now employed in industry have become sponsors. A new challenge is in managing that demand fairly and appropriately for the sponsors and the courses.

Well-Prepared Students

The pre-requisite courses provide the students with a strong tool-box of skills to apply to the given project. In particular, the prerequisite courses emphasize problem-solving, the ability to locate knowledge and resources, the use of logical /

methodical approaches, cross-mentoring, and other life-long learning skills that allow the teams to remove blockers and successfully complete the project.

Experienced Instructors

Careful development of faculty resources through the selective hiring and mentoring of clinical faculty has allowed us to staff the Capstone courses with experienced instructors with good mentoring skills. Most of the instructors have extensive experience in industry in addition to academic credentials. Just as sponsors must be willing to cede control of their projects to the students, so must the instructors be able and willing to step back and only provide a mentoring or consultation role. The instructors base this on expert judgment on how much help, and what kind of help will best allow the teams to "stretch" without floundering.

Supporting Resources

The Capstone program has strong partnerships with other organizations within the university to provide students greater access to resources. These include a well-stocked computer lab with staff that helps the student teams set up critical infrastructure (e.g., virtual servers with appropriate tools – databases, content management systems, etc.). The program also maintains a set of Mac laptops with the iPhone/iPad development environment, for developing mobile apps.

ASSESSMENT AND EVALUATION TOOLS

The development of appropriate assessment and evaluation tools, able to monitor, measure and fairly assess a wide-spectrum of projects, is critical for the program [9]. Students must be given clear expectations, and fairly assessed, in the face of significant variability[10].

Variability in Projects

The projects executed by the Capstone courses vary considerably. In a recent term, a project submission from Eaton Corporation for a Disaster Recovery content management system was accompanied by a four page project charter, complete with milestones, gate review dates, and role definitions. A submission from Microsoft consisted of several examples of interesting Windows 7 Phone applications the team might want to build, but no specific guidance. Both were valid requests, and both were executed successfully by the project teams. However, each project team had to adjust the project activities and deliverables to suit the project and sponsor. The Microsoft team had to propose several specific applications, one of which the sponsor chose, and had to submit the final application to the Microsoft Marketplace. The Eaton team had to work with corporate security constraints, and develop detailed transition plans.

Variability in Sponsors

The sponsors as a whole are extremely dedicated. Most take on the role because they have a desire to mentor the students and pass on knowledge and skills they have acquired. However, each sponsor is an individual, with a different personality, background, expectations and skills. Some are very prescriptive and "hands-on" (dare we say "micro-managers"), while others provide only high-level guidance, and only when asked. Most sponsors are very busy with their primary jobs, and must make time in their schedules to meet with the students. Sometimes, business trips or other commitments interfere with weekly meetings. Some sponsors are local and can meet with the students face-to-face. Some are three time zones away and can only meet at odd hours via telephone or Skype.

Students are encouraged to negotiate a communication plan with the sponsor, agreeing on the frequency and method of communication, how decisions will be made and recorded, etc.

Variability in Technology

Learning a specific technology is explicitly *not* a goal of the Capstone courses. Successful completion of a team's project frequently requires the students to learn tools, languages, frameworks, etc, that they have never used before – for example, the iPhone development environment, the C# language, the FaceBook API. Students are encouraged to use "life-long learning" skills to quickly come up to speed on these technologies, and a significant amount of the weekly standup discussion relates to managing these risks. Students from other teams frequently offer advice to teams new to a technology.

In their final reports, students frequently mention the need to learn new technologies and skills as one of their main risks for the term, and also as one of things they are most proud of at the end.

Expectation Setting and Performance Assessment

Given the variability of the projects, it is important that students have a clear understanding of what is expected of them, and that student performance be assessed as clearly as possible. This is an area of active research within the program. We are developing and piloting improved rubrics for evaluating specific project deliverables. We are particularly interested in rubrics for the Project Workbook. Our current rubric evaluates each area (e.g., Project Management, Requirements), based on a combination of a "justifiable" choice of method, and good execution of the chosen method. For example:

- There is situational awareness at the high level about the software development process, and why this content is important and appropriate for this specific project.
- The content is both well presented and informative, given the expected audience. There is appropriate justification for its inclusion.
- There is evidence that the content reflects actual performance during development and helped define and execute the project.

We do not prescribe a specific set of methods, but provide suggestions. For example, for the Requirements and Analysis area:

- Describes problem or opportunity, with supporting material (problem statement, business case, domain analysis).
- Defines project scope.
- Selects beneficial work products and presents them (class models, sequence diagrams, scenarios/use case diagrams, class diagrams, relational diagrams).
- Identifies functional/non-functional requirements (use cases, scenarios, user stories).
- Specifies complexity and/or order of priority of requirements.

This balancing act between evaluating the quality of the process (reasonable choices) and the quality of the outcome is central to appropriate evaluation of student performance.

CONCLUSIONS

As successful as the Capstone program has been, we are continually making improvements to the courses, based feedback from students, faculty, and sponsors.

One of our main focus areas at this time is further refinement of the aforementioned rubrics, to better set expectations and to evaluate student performance in the face of the breadth of projects the students execute. We feel that this is a fertile area of investigation that will benefit all of the parties.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under NSF CCLI Grant No. 0837555 and the CERCS IUCRC Center for Enterprise Transformation and Innovation (CETI), supported by the NSF-IUCRC Program, Grant No. 0630188.

REFERENCES

[1] Zidek, L., "Engineering Service Learning, Engineering Entrepreneurship and Assessment: Building a Program That Works", *40th ASEE/IEEE Frontiers in Education Conference*, October 27 - 30, 2010, Washington, DC, 978-1-4244-6262-9/10.

[2] Tan, J., Phillips, J., "Incorporating service learning into computer science courses, *Journal of Computing Sciences in Colleges*, Volume 20 Issue 4 April 2005.

[3] Cutler, S., Borrego, M., "Developing Global Competence in Graduate Engineering and Science Students through an IGERT International Internship Program", *40th ASEE/IEEE Frontiers in Education Conference*, October 27 - 30, 2010, Washington, DC, 978-1-4244-6262-9/10.

[4] Ikonen, M., Kurhila, J., "Discovering high-impact success factors in capstone software projects, *Proceedings of the 10th ACM conference on SIG-information technology education*, October 2009.

[5] Hurtig, J., Estell, J., "A common framework for diverse capstone experiences", *Proceedings of the 39th IEEE international conference on Frontiers in education conference*, October 2009.

[6] Sundheim, N., Asquith, J., "First Year Cross-Functional Design Teams: Engineering and Business", *40th ASEE/IEEE Frontiers in Education Conference*, October 27 - 30, 2010, Washington, DC, 978-1-4244-6262-9/10.

[7] Kruchten, P., "The Rational Unified Process: An Introduction", Addison Wesley, 2004.

[8] Gorka, S., Miller, J., Howe, B., "Developing realistic capstone projects in conjunction with industry", *Proceedings of the 8th ACM SIGITE conference on Information technology education*, 2007.

[9] Clear, T., Goldweber, M., Young, F., Leidig, P., Scott, K., "Resources for instructors of capstone courses in computing", *SIGCSE Bulletin*, Volume 33 Issue 4 December 2001.

[10] Montfort, D., Brown S., Pegg, J., "An investigation of the adoption of an assessment instrument for capstone design courses", *Proceedings of the 39th IEEE international conference on Frontiers in education conference*, October 2009.

AUTHOR INFORMATION

Thomas Bihari, Senior Lecturer, Department of Computer Science and Engineering, The Ohio State University, Senior Consultant, Nationwide Insurance, bihari@cse.ohio-state.edu.

Igor Malkiman, Senior Lecturer, Department of Computer Science and Engineering, The Ohio State University, igorm@cse.ohio-state.edu.

Moez Chaabouni, Senior Lecturer, Department of Computer Science and Engineering, The Ohio State University, chaaboun@cse.ohio-state.edu.

Joe Bolinger, Ph.D. Candidate, Dept. of Computer Science and Engineering, The Ohio State University, bolinger@cse.ohio-state.edu.

Jay Ramanathan, Director, C.E.T.I, Dept. of Computer Science and Engineering, The Ohio State University, jayram@cse.ohio-state.edu

Rajiv Ramnath, Director, C.E.T.I, Associate Professor of Practice, Dept. of Computer Science and Engineering, The Ohio State University, ramnath@cse.ohio-state.edu.

Michael Herold, Ph.D. Student, Dept. of Computer Science and Engineering, The Ohio State University, herold@cse.ohio-state.edu.