A 15-year Case Study of a Real-World-Projects Capstone Course in Computing

Charles C. Tappert
ctappert@pace.edu

Andreea Cotoranu
acotoranu@pace.edu

Jonathan Hill
jhill@pace.edu

John V. Monaco
john.v.monaco@pace.edu

Seidenberg School of Computer Science
and Information Systems
Pace University
Pleasantville, NY 10570

A 15-year Case Study of a Real-World-Projects Capstone Course in Computing

Abstract

The capstone computing course at Pace University provides students with experience working on computing and information systems projects with real-world aspects. Over one course or a two-course sequence, students have the opportunity to develop both the hard and soft skills that are sought after by industry. Since the course was introduced fifteen years ago, significant changes in the instructional environment and delivery method have taken place to meet shifting demands in distance education. The structure of the course has evolved from a traditional face-to-face, to a dispersed hybrid, to an online format. Despite these changes, student satisfaction and quality of project deliverables has remained high. A unique peer evaluation system and course management tools have been developed to facilitate the course delivery. This paper surveys the current landscape of capstone courses through a survey of 49 institutions and provides a review of the course at Pace University.

Keywords: Capstone computing courses, project-oriented courses, distance education, collaborative skills, student assessment.

1. INTRODUCTION

A capstone course is a course offered as part of an academic major aiming to bring together major aspects of the academic discipline(s) related to the said major (Ford, 2002). The aim of our capstone course in computing is to familiarize students with how their trade is plied in organizations, so that the program of study delivers "the practice" part of the promised "theory and practice." The projects are "real world" in every respect as they entail the development of an application desired by a real world customer. As in industry, applications are developed by a small, collaborative team which needs to communicate with the customer, coordinate its activity, attend to internal decision-making, and, as observed by Denning and Dunham (2001), be sensitive to delivering value. The applications press into service current technology. This is technology with which the students are often unacquainted inasmuch as it may be specialized, new, or at least new to them. Students learn about real-world technology through their own group's experiences as well as through the reports from other groups. A soft skill of transcending importance, emphasized by activities throughout the capstone, is the ability to communicate on technical concepts and issues; orally, in written reports, and via Web media; to both peers and lay people. The soft skills acquired through a capstone course are perhaps one of the greatest rewards of this academic experience. These include problem solving, communication, and teamwork skills which are becoming essential for work in industry (Gardiner, 2015).

Following a ten-year review (Tappert & Stix, 2011), this paper provides a fifteen-year review of Pace University's capstone course in computing. In the fifteen years (2001-2015) since the capstone course assumed its project-based form, the most significant change has been in its presentation. For the first five years the course spanned the fall and spring semesters and was face-to-face. In 2006 the course was condensed into a one semester offering. For projects, this meant accelerating requirements elicitation, system development, and testing. We responded with agile methodology. In 2006 the course delivery shifted from face-to-face to "hybrid", where students collaborate remotely except for three meetings – at the beginning of the semester for orientation, at the middle of the semester for midterm project status reports to the class, and at the end of the semester for final project presentations. By 2009 the format was entirely online for those students for whom attendance was geographically infeasible.

The remaining sections of the paper cover the following material: section 2 surveys the landscape of existing capstone courses in computing; section 3 describes the current team and project-oriented capstone course as a case study; and conclusions are drawn in Section 4.

2. SURVEY OF CAPSTONE COURSES IN COMPUTING

To get a sense of how various academic institutions implement their capstone courses, 84 universities having capstone courses in computer science or engineering were contacted by a spring 2015 capstone project team (Brewer et al., 2015). Of the 84 universities polled with a set of 16 questions, 34 responded, including CMU, U. California, U. Maryland, U. Tech. Sydney, NYU Wagner, U.S. Air Force Academy, and U.S. Naval Academy. For schools that did not respond, information in some cases was obtained from the internet, resulting in appropriate information from up to 49 institutions, depending on the information. The key information obtained is described here.

2.1 Determination of Projects and Customers

Capstone course projects were determined in three different ways: by the project customers, by the students, and by the instructor. The vast majority of the projects – 33 of the 48 universities where this information was found – were generated by the project customers. In some cases the students were required to sign non-disclosure agreements (Baker, 2011). Occasionally, professional associations built competitions between schools (Formula, 2015).

Rarely are students allowed to determine their own projects – only 9 of 45 universities take this approach – and usually in such cases the projects must relate to a specific subject matter. Also, instructors rarely designed the projects – only 5 of 45 universities. However, in a number of cases – 10 of 45 universities – projects were determined jointly by the real-world project customers and the instructor, and in most cases involving external customers the instructor had to approve the projects.

2.2 Student Team Selection

Students are typically assigned to a team in one of three ways: the students form their own teams, the students are randomly assigned to teams, or the instructor forms the teams.

Self-selected teams allow students to have more control over project development. Team members are often committed to and enthusiastic about a shared goal. This method may also be appropriate for students who know each other well enough to support and hold each other accountable in a productive manner.

For randomly-selected team members, there exist team-generating websites, such as http://www.randomlists.com/team-generator/. Apple also has an iOS app called Team Shake that will randomly select names to form teams.

Instructor-selected teams are formed when the instructor uses his/her judgment to assign team members to a particular project. Teams may be formed based on expressed interest by students for particular projects, geographical location, or student age and work history. At Pace University, for example, a project preference rank survey is used to inform team formation. Teams are formed by the instructor based on project preferences, technical capabilities and geographic location. Research suggests that instructor selection is the most beneficial method to establish groups (Nilson, 2010). The instructor can ensure a diverse group of students with regard to, for example, academic performance, location, gender, and nationality. The diversity achieved by this selection process prevents cliques of students from forming and helps students develop important social skills which can result in learning the material better (Nilson, 2010). In addition, the diversity in student backgrounds is expected to contribute to a rich set of ideas and innovative solutions for the projects at hand.

2.3 Dispersed Teams

A geographically dispersed team (also known as a virtual team, distributed team, or remote team) is a group of individuals who work across time, space, and organizational boundaries with links strengthened by webs of communication technology (Virtual Team, 2015). These teams are similar to traditional teams but are geographically dispersed and rely heavily on virtual methods such as email and virtual conference applications. Instructors believe experience working on a virtual team prepares students for the growing business demand (Goldberg, 2014). Of the schools surveyed, roughly 80% of capstone courses are offered in-person or as a hybrid compared to 20% offered entirely online. With most of the online offerings, teams are dispersed both nationally and internationally (Figure 1).

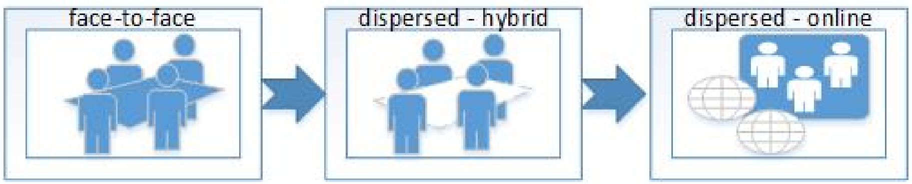

Figure 1. Evolution of capstone course delivery.

Management of dispersed teams presents unique challenges not limited to culture, language, time zone, and geography. One of the biggest challenges is the ability to build trust among team members. The term “jelled team” has been used to describe a strongly knit team that relies on trust. The probability of project success for “jelled teams” increases significantly when compared to other teams (Pressman & Maxim, 2014).

Time zone differences add an extra layer of complexity when scheduling virtual meetings and tracking communications. This also limits the mutual availability of team members. Of the schools with dispersed teams, 65% indicated difficulty with communication. Teams were often found to utilize video conferencing applications, such as Skype, GoToMeeting, and Google Hangouts. File management also proves to be an ongoing challenge as many teams resort to email attachments with versioned filenames, despite the availability of distributed version-control software such as *Git*. The team leader is typically tasked with managing project data on behalf of the team members.

Sixty percent of schools with dispersed teams held weekly or biweekly team meetings via phone conference, chat room, or virtual conference (Skype, Lync, etc.). The team leader maintained communication with team members to ensure project advancement. Most teams found it best to maintain one current version of project drafts with tracked changes and updates. Some teams designate a team manager to keep track of all changes.

2.4 Student Assessment

Most of the capstone instructors polled required some sort of progress report to be turned in regularly. The most common frequency was weekly at 48% since this coincides with a typical course schedule and is frequent enough to allow the instructor to track progress. Bi-weekly progress reports were required in 17% of the polled schools, and 9% allowed more than two week intervals between progress reports. Surprisingly, 26% of universities did not require progress reports, indicating that time management was the responsibility of the student team.

Capstone project grades are usually based in part on peer reviews designed to evaluate the performance of team members as seen through the eyes of team members. Peer reviews take into consideration the project goals, team communication, and division of labor. Geographically near teams have the opportunity to meet in person on a regular basis. Team members can take cues from one another through in-person interaction and the responsibility of each team member is clear. Mid-semester peer reviews assist the instructor in identifying contribution or issues among the team members that need to be addressed before the project can be impacted negatively.

The weekly time required of students varied greatly from two hours per week to forty hours per week. A majority (60%) of programs expected more than ten but less than twenty hours weekly.

3. CASE STUDY OF A CAPSTONE COURSE IN COMPUTING

We use team projects modeled on real-world development practice to provide students with the educational experience of collaborative efforts, similar to what is done in industry, in order to design, build, and test computer information systems.

3.1 History

The capstone computing course at Pace University has been offering students experience with the development of real-world computing projects for the last fifteen years. The course has evolved from face-to-face delivery in the first five years to a hybrid course involving dispersed teams today.

Beginning in 2006, the projects course was migrated from a traditional face-to-face format to an online format. The face-to-face pedagogical mechanisms employed had to be adapted for distributed student project teams. The online format precludes automatic, weekly meetings that act as a safety net to the teams' interaction and smooth functioning.

As the ability for impromptu team discussions before and after class disappeared and online communication became dominant, the team dynamics became more complicated. In addition, we needed to revisit the way we graded the performance of team members. It is well known that projects undertaken by groups lacking co-presence presuppose a higher level of organizational and process skills among their members (Cusumano, 2008).

For the past ten years the capstone course has been a project-oriented, one-semester, web-assisted course for masters-level computing students in which student teams develop real-world computer information systems for actual customers. Students learn the importance of a systematic approach in the process of developing robust systems, the management of projects, how to interact with customers and conduct requirements analysis, how to build and test systems, and the related technical and soft skills. Emphasis is placed on developing skills and knowledge in technical areas that have practical value in the workplace. In addition to technical skills, students develop problem-solving, critical thinking, communication, and teamwork skills. By working on real-world systems with actual customers, the students learn the appropriate skills for filling meaningful roles in the professional IT workplace.

3.2 Project and Research Interplay

A critical aspect and the signature of this course is the interplay of student projects and research done by students and/or faculty. One of the approaches we use is to support doctoral student dissertation and faculty research to create research-supporting projects in several of our courses. We teach our dissertation students how to conduct research in a number of areas of computing, and our student project teams how to develop real-world computer information systems. In recent years, we have experimented with the interplay of dissertation research and projects created specifically to develop the supporting software infrastructure for that research. Some of the project customers are faculty members or dissertation students who need supporting software infrastructures to conduct their research. Thus, there is interplay between the project and research activities.

We have found this interplay between research and project activities to be exciting and productive. The main benefits have been to increase faculty research productivity, to facilitate the completion of doctoral dissertations, and to strengthen capstone classes in the master's programs. The mechanism has been using research problems to provide projects, and using projects to supply computing infrastructure. We term this symbiotic relationship the research/project interplay.

3.3 Team Project Categories and Publications

The team project focuses on developing a computer information system that meets an actual customer's needs.

Table 1 presents the 142 projects conducted over the last fifteen years together with the resulting 185 publications. In recent years, many projects focused on biometrics systems, and an incubator containing system code and databases has been created as a source of material primarily for dissertation studies. Web applications include for example a web interface to a backend database. Pervasive systems are typically mobile device (e.g., smartphone) applications – an example is an interactive human-machine, flower identification system that outperforms either the human or machine alone. PC applications are standalone PC applications. The artificial intelligence (AI) systems include various AI and machine learning systems such as modeling Hubel-Wiesel-like line and edge detectors in a character recognition problem. The "Other" category includes a variety of projects, such as literature reviews and other non-system-creating projects, and about a third of these are quality assurance projects that assist the instructor with the quality of the work in semesters having a large number of projects.

Project Category	Number Projects	Project Semesters	Project Related Pubs	Offshoot Pubs
Web Applications	27	37	30	
Pervasive Systems	18	29	26	1
PC Applications	14	21	16	
Artificial Intelligence	19	25	48	26
Biometric Systems	48	74	94	25
Other	16	25	21	10
Totals	**142**	**211**	**235**	**62**

Table 1. Summary of projects and publications.

Table 2 lists the project sources – faculty research, doctoral student research, external community systems (e.g., systems for local hospitals, collaborative research with other universities, etc.), and internal university needs (e.g., a university-wide IRB system).

Project Source	Number
Faculty Ideas or Research	58
Student Ideas or Research	44
External Community	19
Internal University Needs	21
Totals	**142**

Table 2. Project sources.

Table 3 lists the publication categories – the largest being an annual internal conference, next largest being external conferences, then doctoral dissertations and masters theses (there were only 5 masters theses since most students choose to take the capstone projects course rather than write a thesis), and journal articles and book chapters (only two book chapters). A detailed list of the publications for the first ten years, 2001-2010, were provided in the Appendix of an earlier paper (Tappert & Stix, 2011). Of the 297 resulting publications, 235 were directly project-related, and 62 were similar in kind and designated “offshoot publications” (Table 1).

Publication Type	Number
External Conference Papers	67
Journal Articles & Book Chapters	15
Doctoral Diss. & Masters Theses	31
Internal Conf. Papers & Tech Rep.	184
Totals	**297**

Table 3. Publication categories.

3.4 Project Examples

Figure 2 is a partial list of the spring 2015 semester projects. Most of the project customers were doctoral students in our PhD in Computer Science and Doctor of Professional Studies (DPS) in Computing programs. The Projects page lists the projects and contains, for each project, the project ID number, the project customer(s) with links to detailed contact information, a link to a detailed project description, and the student team (listing the team leader first).

A continuing line of research, and one that brought forth many projects, is behavioral biometrics. Over the last five years we have had ten semesters of masters-level project work, four doctoral dissertations, three external conference papers, a book chapter, and a journal article.

Project Information			
ID	Customer	Project	Student Team
1	Dr Kalyanasundaram, Verizon LeighAnne Clevenger, DPS'16 Javid Maghsoudi, DPS'16 Dr. Tappert, CSIS	Mobile Phone Security	Sara Siddiqui Nikhita Gopidi Nishant Patel Nitish Pisal Tanya Sahin Shreyansh Shah
2	Dr. Beigi, Recognition Tech. Hugh Eng, DPS'16 Dr. Tappert, CSIS	Smartphone/Laptop Security Acceptability/Ease-of-Use Study	Jimmy Leon George Chacko Emmanuel Ruiz Daniel Weise
3	Vinnie Monaco, PhD'15 Dr. Tappert, CSIS	Modeling Terrorist Activity	Ron Williams Kevin Henry Kristie Roman William Wrublewski
4	Javid Maghsoudi, DPS'16 Hui Zhao, PhD'17 Dr. Tappert, CSIS (Vinnie Monaco, PhD'15)	Android Biometric System: Gestures, Motion/Orientation	Chris Carlson Tony Chen Jeremy Cruz
5	Amir Schur, DPS'13 Dr. Tappert, CSIS	Human-Assisted Pattern Classification	Jem Staine Dena Berthelette Alex Lukyanov Alexander Mei Adriel Velazquez
6	LeighAnne Clevenger, DPS'16 Dr. Tappert, CSIS (Vinnie Monaco, PhD'15)	Investigate and Design Multi-Biometric System	Chris Chiffriller Gabriel Dos Santos Chris George Subah Sachdeva

Figure 2. Partial project information on course website, spring 2015.

3.5 Project and Team Selection

Projects come from faculty and dissertation students interested in developing systems to further their research, from other departments or schools of the university needing computer information systems, from non-profit community institutions such as local hospitals, from local research institutions, and from interests of the project students. The instructor sizes and shapes each project to be an appropriate systems development experience for the students, forms the student teams, and assigns each team to a project.

In recent years there have been two significant changes in the source of project customers. The first has been to extend projects to private industry, and we began exploring this direction in 2010 with several small companies. Then, in 2014, we targeted large companies, initiated our first funded projects with Verizon at $50k per project, and are currently negotiating projects with IBM and General Electric. This funding is primarily for student tuition, but also can cover project equipment, as well as student and faculty attendance at conferences to present the project work, often desired by the companies for exposure. The second change in the source of project customers, which has been evolving slowly over more than ten years, has been to have doctoral students serve as customers and have the associated project teams develop the system infrastructure that can be used for running experiments for the doctoral research. For example, we have had many projects develop biometrics systems for experimental purposes and our university is now widely recognized for work in the biometrics area. Now, with the majority of the projects supervised by doctoral students, including some with private industry, this has become the hallmark of our capstone projects.

From the project descriptions posted on the course website the students complete a project preference form during the first two weeks of the course. Students list their current company and job title, number of years of work experience in information technology, work and home locations, whether they can attend the three classroom meetings, preferred communication mode (email, phone, IM, etc.), top five project choices, top five availability time choices for project communication (day of week plus morning, afternoon, or evening), project skills (requirements engineering, system design, programming, databases, web design, networking, communication/leadership, etc.). The instructor uses this information to form teams, to select team leaders, and to assign teams to projects (Figure 3). Due to the complexity and scope of the projects, teams are sometimes subdivided into sub-teams. Some projects also require collaboration between teams, leading to inter- and intra-team coordination.

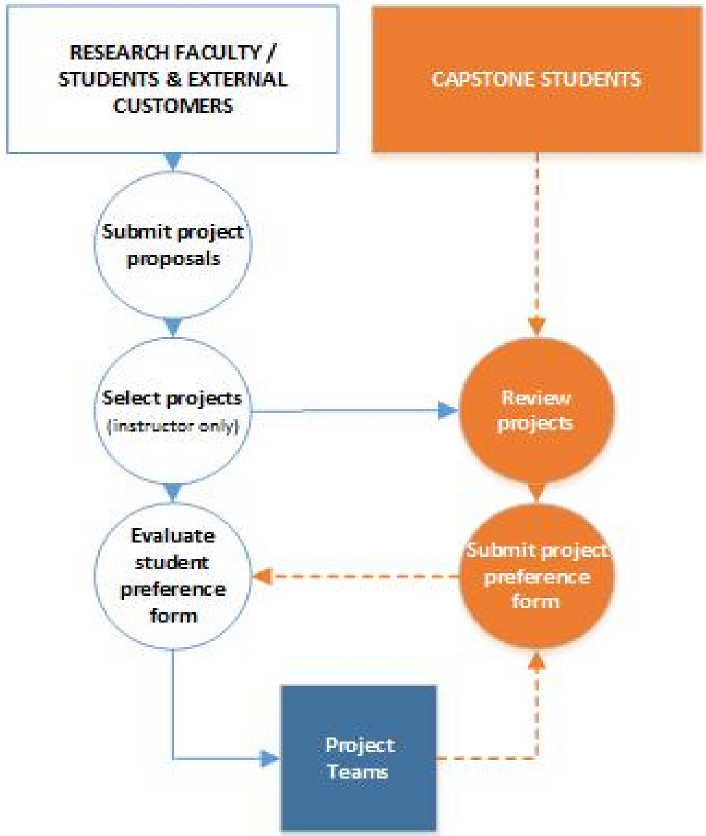

Figure 3. Project and team selection.

3.6 Teams, Roles, and Methods of Work

A team is a group of individuals having the responsibility to jointly accomplish an objective, and in this course the objective is to successfully complete a project. It is widely accepted that work in teams enhances learning by creating an "active learning process." (Bonwell, 1991) Student teams have been found particularly effective when the students actually need each other to complete the project. It is also the norm for employees to work in teams, and teams are pervasive across industry, academia, and government.

Effective teamwork requires the division of responsibility, the coordination of efforts, communications to expedite coordination, and group governance for collective decision making, conflict resolution, and the control of deviance. Denning and Reihle (2009) draw attention to both the importance of group dynamics to software engineering and the traditional failure to accord them proper regard in project development courses.

Most of the systems involve one or more of the following: programming, a database, a computer network, a Web interface. Java is the preferred language for projects that require programming. Non-programmers or weak programmers can contribute in many ways other than programming. A team usually consists of 3-5 students – an Architect-Designer, one or two Developers-Implementers, a Quality Assurance-Tester, and a team Leader-Liaison (Figure 4). For small teams several team member functions can be combined.

Figure 4. Team roles.

Although the requirements for the projects come from the customers, the course instructor is the "boss" or "Chief Information Officer" of each project team, and, as such, the person who makes all the major decisions. The project customer knows what he/she wants as an outcome but may not know the technical aspects of the project work (algorithms, program code, etc.). Some projects have subject matter experts who are knowledgeable about certain domain related aspects of a project. The customer, the subject matter experts, and the instructor can guide the team but are not expected to make major contributions to the actual project development effort.

For project development work we use the agile methodology, particularly Extreme Programming (XP) which involves small releases and fast turnarounds in roughly two-week iterations (Beck, 2000). Each team delivers a prototype system that performs the basic required functions to their customer at the halfway point of the semester. This is possible since, according to the 80-20 rule (Pressman & Maxim, 2014), 80% of the project can be completed in 20% of the time it would take to deliver the complete system. A complete system is delivered at the end of the semester.

3.7 Course Management

Pace University has campuses in New York City and Westchester, NY. Currently about two-thirds of the students live or work in the greater New York City area. The remaining third are mostly from other regions of the East Coast with some from as far away as California and foreign countries. The distributed team issue is handled by a number of mechanisms and guidelines.

To facilitate communication among the project stakeholders, we insist that, except for extenuating circumstances, communication between a team and instructor, and between a team and a customer, be through the team leader, with all team members copied on communication email and given summaries of face-to-face meetings. This reduces communication to the instructor from individual students and keeps all stakeholders updated on project activities. The instructor also creates and uses email distribution lists for the whole class, for each project team including the customer, and for the customers. Project team leaders must be local, either living or working in the greater New York City area, to permit occasional face-to-face meetings with the project customers and instructor.

An extensive course website maintained by the instructor efficiently presents all the course information for convenient centralized access as follows:

- Homepage – instructor information, textbooks, course description and goals, course requirements, and grading system.
- Syllabus – weekly readings and assignments.
- Projects – a table of the semester's projects provides for each project the customer's name and contact information, the project description, the names of the students on the project, and a link to the project team's website.
- Students – contains student photos so students know their classmates and the instructor can recall a student, possibly years later, to provide letters of recommendation.
- Project Deliverables – lists and describes project deliverables.
- Grades – contains table of graded events and the current student grades indexed by the last 4 digits of their university ID number.
- Link to the Blackboard educational software system (Blackboard, 2015) used for quizzes, discussions, and collecting digital assignments.

Three 3-hour classroom meetings are important to bring the local students together so they can meet some of their teammates and form some face-to-face bonding. The first meeting occurs after the first week of the semester. By this time:

    - the students have introduced themselves online through a Blackboard forum, reviewed the course website, and submitted project preference information to the instructor

- the instructor has received the students' project preferences and associated information, formed the student project teams, assigned teams to projects, chosen project team leaders, and posted the information on the project's page of the course website

At this meeting the instructor and students introduce themselves face-to-face (half hour), the instructor gives a lecture on the nature and value of conducting real-world projects in a capstone course (one hour), the instructor reviews the specifics of the course material and describes each of the projects (one hour), and the students group themselves into their project teams and begin planning project activities (half hour).

At the second mid-semester meeting the students make PowerPoint slide presentations of their project prototypes. Material covered in these presentations includes, as appropriate and as time permits, a subset of the following items: brief description of project, summary of project specifications, frequency of meetings with customer/stakeholders and usual method of communication, plans to address changes in customer requirements, summary of user stories collected (if any), analyses accomplished (object-oriented might include defined classes and operations), design decisions and the trade-offs encountered, work breakdown structures, PERT chart, and/or Gantt chart, components built/planned, testing strategy, what was accomplished to complete the prototype, what will be added in the remainder of the semester, what has been easy/difficult during this half of the semester, and a prototype demonstration. Many customers attend the second meeting.

At the third (semester-end) meeting the students present their final project system. This meeting is similar to the second meeting, and most of the customers attend the final presentations.

3.8 Student Assessment

Student assessment is currently as follows: individual quizzes (20%), first quarter team project (10%), second quarter team project (20%), third quarter team project (20%), and fourth quarter team project (30%). The quarterly evaluations include the submission of a team project report that, by the end of the semester, grows into a technical paper that for most teams is published in an annual internal computing conference. Thus, 80% of a student's grade is based on their contribution to the team effort with the quizzes (based primarily on the textbook material) providing the only direct individual assessment. Mid-term and final exams used in earlier two-semester versions of the course were eliminated allowing the students to focus on the project work in this one-semester course. The team has the ultimate responsibility for the project work and is graded accordingly. Grades on team events are determined by first assigning a team grade and then adjusting an individual student's grade up or down based on the peer evaluations along with additional input from the customers and instructor.

Peer evaluations are used to assess the project contributions of each team member. Although used when the course was conducted in the classroom, peer evaluations are even more critical for distributed teams because some team members have minimal, if any, direct contact with the customer and instructor. Obtaining individual student grades on teamwork has been reported in the literature. For example, Clark, Davies, & Skeers (2005) created an elaborate web-based system to record and track self and peer evaluations, Brown (1995) has a system similar to ours but which uses more granular numerical input, and Wilkins & Lawhead (2000) use survey instruments.

The students are required to provide peer evaluations four times during the semester, one at each of the quarterly checkpoints. The recent evolution of peer evaluation summary charts with associated grades is shown in Table 4 for a four-member team. Each of the four evaluation columns shows the evaluation of a team member evaluating the team members. The summary column shows the adjusted sum of each row of evaluations, and the grade column shows the student grades. Here, a team grade of 85% is first determined and then individual grades are adjusted relative to the team grade.

Team Member	Eval 1	Eval 2	Eval 3	Eval 4	Summary	Grade
1	+	=	– –	++++	+++	91
2	=	=	–	– – –	– – – –	77
3	–	=	++++	– –	+	87
4	=	=	–	+	=	85
Average	=	=	=	=	=	**85**

Team Member	Eval 1	Eval 2	Eval 3	Eval 4	Summary	Grade
1		=	– –	++++	++	89
2	+		+	– – –	–	83
3	– –	=		–	– – –	79
4	+	=	+		++	89
Average	=	=	=	=	=	**85**

Team Member	Eval 1	Eval 2	Eval 3	Eval 4	Summary	Grade
1		4	2	8	4	89
2	5		4	0	-1	84
3	0	3		2	-5	80
4	5	3	4		2	87
Total	**10**	**10**	**10**	**10**	**0**	**85**

Table 4. Evolution of team peer evaluation summary charts:
prior to fall 2015 (top), fall 2015 (middle), and current (bottom).

In the top chart, which was used prior to the fall 2015 semester, each column must average "=", and student grades were adjusted up or down 2% for each "+" or "–" sign obtained in the summary row. The middle chart, which was used in the fall 2015 semester, varied slightly from the top chart – the difference being that the students could not evaluate themselves, and in this example the inputs (+, -, =) have been adjusted slightly from the top chart to average "=". This top-chart method favors and thus encourages students (if they understand it, and most of them do) to give themselves high evaluations, and the middle chart shows that the student who did not give himself/herself a high evaluation in the top chart scored better in the middle chart. The bottom chart, which was first used in the spring 2016 semester and will likely be used henceforth, employs a numerical system where each team member allocates a total of 10 points among the other team members (again no self-evaluation) so that each evaluation column adds to 10. Each entry in the summary column is obtained by adding the row values and subtracting 10, so that the average of the summary column adds to zero, and the summary value is used to adjust the team grade for each team member. This numerical method simplifies the awkward accounting for the +, -, and = signs and is the one described by Brooks (2015).

For simplicity, this table shows only the peer evaluations, but customer and instructor evaluations can be included as well, or simply used for adjustment purposes. Team leader and instructor evaluations can also be given extra weight.

Students are also asked a number of general questions for the time interval in question – the number of hours per week spent on project work, their specific contributions, their strengths and how these were used, their areas needing improvement, and what has enhanced and/or challenged their team's performance – and the responses might influence the instructor evaluation of a student's contribution to the team effort. For additional input the instructor can discuss team member contributions with the team leader.

Since this is a project-oriented course with no midterm or final exams, student grades depend mostly on their contribution to the project work. The usual expected time commitment per student for a 3-credit course is three hours per week in class and twice that outside of class, for a total of nine hours per week. However, because this is an online course where students save commuting time, we expect a time commitment of about ten hours per week, and this additional time commitment is one of the advantages of a distance-learning course.

4. CONCLUSIONS

Capstone courses are particularly important to computing and information systems education. Students develop hard and soft skills, are exposed to a wide range of topics, and foster interdisciplinary collaboration. The project deliverables also provide valuable systems for the customers and support student and faculty research. This enhances relationships between the university and local technology companies, and affords students the opportunity to acquire internal and external publications. Our yearly internal conference is complete with a review process and proceedings. We have found that working to produce publications is a strong motivating factor for the students.

The essence of the course has remained the same regardless of changes in its delivery from face-to-face to essentially online format. Course management tools and a unique peer evaluation system have been developed to facilitate course delivery. Despite the changes in structure and delivery, student satisfaction has remained high as recent course evaluations indicate; results are based on a 5-point Likert scale (strongly agree - strongly disagree), and percentages encompass both *strongly agree* and *agree* responses:

- 93% agree that working on real-word projects for actual customers was a good learning experience,
- 86% agree that writing a technical paper for the Research Day Conference was a good learning experience,
- 82% agree that quizzes were an appropriate individual evaluation method for the readings of the course,
- 86% agree that "Projects in Computing and Information Systems" by Dawson was a good book for the course,
- 79% agree that peer evaluations were an appropriate method of determining individual contributions,
- 43% agree that three optional in-class meetings were useful for this course, while another 43% were neutral.

5. ACKNOWLEDGEMENTS

We thank the various project teams that contributed to the information content of this study.

6. REFERENCES

Baker, J. Sr. (2011). Challenging the Traditional Graduate Information Systems Program. *IT Professional*, 13:06.

Beck, K. (2000). *Extreme Programming Explained*. Addison-Wesley.

Blackboard (2015). Courseware product marketed by Blackboard, Inc. Retrieved from http://www.blackboard.com/.

Bonwell, C. & Eison, J. (1991). Active Learning: Creating Excitement in the Classroom AEHE-ERIC Higher Education Report No. 1. Washington, D.C.: Jossey-Bass.

Brooks, F.P., Keynote Address, EDSIG 2015 Conference, Wilmington, N.C., November 2015.

Brewer, T., Campoverde, N., Hanna, J., Humphries, B., & Tappert, C.C. (2015). Analysis of Capstone Computing Courses. Proc. Student-Faculty Research Day, CSIS, Pace University.

Brown, R.W. (1995). Autorating: Getting Individual Marks from Team Marks and Enhancing Teamwork. *Proceedings of the Frontiers in Education Conference*, IEEE/ASEE, Pittsburgh, Pa.

Clark, N., Davies, P., & Skeers, R. (2005). Self and Peer Assessment in Software Engineering Projects. *Proceedings of the 7th Australasian Conference on Computing Education*, Newcastle, Australia.

Cusumano, M. (2008). Managing Software Development in Globally Distributed Teams. *Communications of the ACM*, 51(2), 15-17.

Denning, P.J. & Dunham, R. (2001). The core of the third-wave professional. *Communications of the ACM*, 44(11), 21-25.

Denning, P.J. & Riehle, R.D. (2009). The Profession of IT: Is Software Engineering Engineering? *Communications of the ACM*, 52(3), 24-26.

Ford, M. (2002). *Beyond the Modern University: Toward a Constructive Postmodern University*. Praeger.

Formula Society of Automotive Engineers Competition, York College of Pennsylvania, http://www.ycp.edu/academics/academic-departments/engineering-and-computer-science/student-projects/fsae-capstone-design/, accessed 2015.

Gardiner, B. (2015). Soft skills now an essential focus of IT recruitment. CIO Magazine, 2015.

Goldberg, J. & Howe, S. (2014). Virtual Capstone Design Teams: Preparing for Global Innovation. *Proc. Capstone Design Conf.*, Columbus, Ohio, June 2014.

Nilson, L.B. 2010. *Teaching at Its Best: A Research-Based Resource for College Instructors.* Jossey-Bass, 3rd edition.

Pressman, R. S. & Maxim, B. (2014). *Software Engineering: A Practitioner's Approach* (Eighth Edition). McGraw-Hill.

Survey Monkey (2015). Retrieved from http://www.surveymonkey.com/.

Tappert, C.C. & Stix, A. (2011). A Decade Review of a Masters-Level Real-World-Projects Capstone Course. *Proc. Info. Systems Educators Conf. (ISECON 2011)*, Wilmington N.C., Nov 2011.

Virtual Team, Wikipedia, http://en.wikipedia.org/wiki/Virtual_team#cite_note-1a, accessed 2015.

Wilkins, D.E. & Lawhead, P.B. (2000). Evaluating Individuals in Team Projects. *Proceedings of the 31st SIGCSE Technology Symposium on CS Education*, Austin, Texas, 172-175.