CapStone: A Cloud-Based Platform for Multi-Party Collaboration on Capstone Projects

Xiaocong Fan
School of Engineering, Behrend College
Pennsylvania State University
Erie, PA 16563
Email: xfan@psu.edu

***Abstract*—Many undergraduate engineering programs require senior students to take capstone design courses to work on industry-sponsored projects. Such a project often involves multiple parties of people playing various roles: students as project workers, faculty members as coordinator or advisors, as well as mentors from the sponsoring industry. In order to promote student success, highly demanded are platforms that can effectively facilitate multi-party collaboration during the development process of capstone projects. CapStone is such a platform that has been developed and deployed at the Behrend College, Pennsylvania State University. In this paper, we present the design rationale of CapStone, explaining the responsibilities of different roles involved in a capstone project and how those responsibilities are meshed into one common platform such that all parties could effectively orchestrate their activities. We have also conducted a study on how well CapStone could help student users in their project development. The responses showed that the adoption of CapStone has significantly improved student success in many ways. Particularly, over 80% of the students found CapStone helpful in project planning/scheduling, reporting weekly progress to advisors, collecting feedbacks on project works, and in writing their project reports.**

***Keywords*—Capstone projects, teamwork, process control, cloud computing, multi-party collaboration, ABET.**

I. Introduction

Many undergraduate engineering programs require senior students to take capstone design courses, where students have the opportunities to work on capstone projects with realistic requirements that are typically proposed by industry sponsors [1]–[6]. Most, if not all, capstone projects are team-based, requiring multiple students from the same or closely-related disciplines to work as a team [7]–[9]. This helps students gain essential teamwork skills and experience that can launch them into successful future careers [10].

In practice, the process of capstone projects may vary in length, from one semester to two- or three-semester course sequence. In addition, a capstone project typically involves people playing different roles: students as project workers, faculty members as advisors or coordinators, as well as mentors from the sponsoring industry. Obviously, it is a big challenge to successfully orchestrate multi-party activities in such a "role-playing game" that spans one or more semesters. The question is, are there any collaboration platforms that can be put into use such that student success and faculty involvement can be largely promoted?

Actually, lack of effective collaboration platforms is repeatedly identified as a common facet of poor teamwork. This issue has been recognized by many, and tools have been developed to support capstone project management. For example, WIER [11] is a web-based tool developed for managing capstone projects in the School of Computer Science and Software Engineering at Monash University. WIER is organized by user types (client, student, supervisor and coordinator), providing several functionalities that are pertinent to project management: template repository, student activity logging, discussion forum, past project searching, and resource sharing. Another web-based application, named easyCapstone [12], was adopted by the Department of Computer Science and Engineering at Qatar University to support senior design activities such as project registration, submitting deliverables, scheduling project presentations, assessing student works and providing timely feedback to students.

There are also web-based tools under development. As reported in [13], an online collaboration system was designed to enhance the capstone projects in the Department of Computer Science and Software Engineering at Southern Polytechnic State University, attempting to enable students, faculty, customers, and IAB members to share project information, monitor project progress, and evaluate project quality. Another work-in-progress was reported in [14], which describes a tool to facilitate students and faculty advisors in the management of capstone projects.

The advance of cloud computing and the ubiquitousness of smart devices make it possible to launch innovative Web applications to facilitate a variety of collaborative tasks. In this paper, we present CapStone, a web-based application that has been developed and deployed at the Behrend College, Pennsylvania State University, to allow users playing various roles in a capstone project to collaborate anytime anywhere.

In comparison with the above-mentioned tools, CapStone differs in one way or another. Here is a list of key features implemented in CapStone: (a) CapStone supports five user types, allowing the course instructor, capstone coordinator, industry mentor, student teams, faculty advisors and co-advisors to "dance on the same wavelength" within a common platform; (b) CapStone has a feedback subsystem where faculty advisors and industry mentors can closely monitor students' progress on their capstone projects and provide timely feedbacks for

further improvement; (c) CapStone has an evaluation subsystem where all evaluation rubrics are carefully designed with performance indicators that are tied with the ABET program outcomes; (d) To enforce uniform quality control in their project report writings, CapStone integrates a few useful CASE (Computer-aided software engineering) tools for Computer Science and Software Engineering students to document and export project artifacts such as requirements cards, use cases, agile reports, and test cases; (e) To facilitate project management, CapStone is a center for reporting weekly progress, for submitting and version-controlling project works, for scheduling presentations, and for tracking project schedules, agile plans, meeting minutes, and requests for changes from industry mentors.

The rest of this paper is organized as follows. In Section II, we introduce the general workflows of the capstone program in the School of Engineering at Penn State Behrend. In Section III, we present the key features implemented in CapStone. In Section IV, we report students' experience of using CapStone, and Section V concludes the paper.

II. Background

For over two decades, industry-sponsored senior design has been a signature program of the School of Engineering at Penn State Behrend [2], and it has been continuously evolved by faculty members in different engineering programs to accommodate discipline-specific requirements.

In general, our capstone program spans two semesters, requiring engineering students to form teams to design and implement systems to meet realistic requirements originated from industrial customers. The whole process includes the following milestones:

1) Project proposal: in summer, call for project proposals are sent to potential industry sponsors, who are requested to submit their project ideas by the end of August.
2) Project selection: in early September, the capstone coordinator of each engineering program should evaluate the proposals and select a set of candidate projects that are appropriate for students in that program.
3) Bid for projects: once candidate projects are selected, they are open to students, who can then read the detail of each proposal and bid for projects they would like to work on.
4) Team formation: by mid-September, student teams are formed based on their common interests in a project as shown in their bids. Each team consists of three or four students [1]. Once a student team is formed, their project is also fixed. To each team, the capstone coordinator also assigns one faculty advisor and two faculty co-advisors. The person who has proposed the project will serve, or designate his/her colleague to serve, as the industry mentor for the student team.
5) Project progress review I: in mid-October, each student team submits their phase-I project report for review, and gives a technical presentation to a 3-person evaluation committee (consisting of faculty advisor and co-advisors).
6) Project progress review II: in mid-December, each team submits their phase-II project report, and gives a technical presentation to their faculty evaluation committee.
7) Project progress review III: in late February, each team submits their phase-III project report, and gives a technical presentation to their faculty evaluation committee.
8) Project progress review IV: in late-April, each team submits their complete project report, and gives the last technical presentation to their faculty committee.
9) Senior design conference: in late-April, each team gives a presentation to public with the audience including the industry mentor, their relatives and friends, and junior students who are encouraged to attend and envision what they are expected to do so that they can plan ahead by identifying potential project opportunities in their summer internships.
10) Project delivery: in early May, each team should deliver their project report and the final product or prototype to their industry sponsor.

The above process involves multiple roles: capstone coordinator, industry mentors, students, faculty (co-)advisors, as well as the course instructor who oversees all capstone projects to ensure that engineering standards and principles are consistently applied. Obviously, the success of a capstone project largely relies on the intentional integration of multi-party collaboration.

However, due to lack of collaboration platforms for capstone projects, our faculty members (as instructors or advisors) have experienced difficulties in multi-party collaboration, many of which are also echoed by the practices shared in the literature [12]–[15]. These difficulties exhibit in several aspects:

1) Process control: As the development process goes on, students can easily lose track of important project schedules; faculty advisors often wonder whether his advised team is making expected progress.
2) Quality control: To increase the quality of project deliverables, it is suggested to give students access to grading rubrics, best design practices, report writing guidelines and templates. In addition, students should be able to access advisors' feedback on their work-in-progress in a timely manner so that changes can be implemented accordingly. The effective management of these documents and the fast delivery of paper-based feedback can raise big overheads to instructors and advisors, which would inevitably impede the quality of project outcomes.
3) Project management: Along the development process, many artifacts can be generated for a project, such as requirements, change requests, use cases, reviews, meeting minutes, test cases, test execution reports, project reports, etc. The lack of an appropriate project portfolio oftentimes makes it difficult to monitor, collaborate, and review the project development [13]. Year after year, it becomes even more difficult to maintain those ever-

[1]More often than not, multi-disciplinary teams are formed to work on projects that demand a wide range of skills.

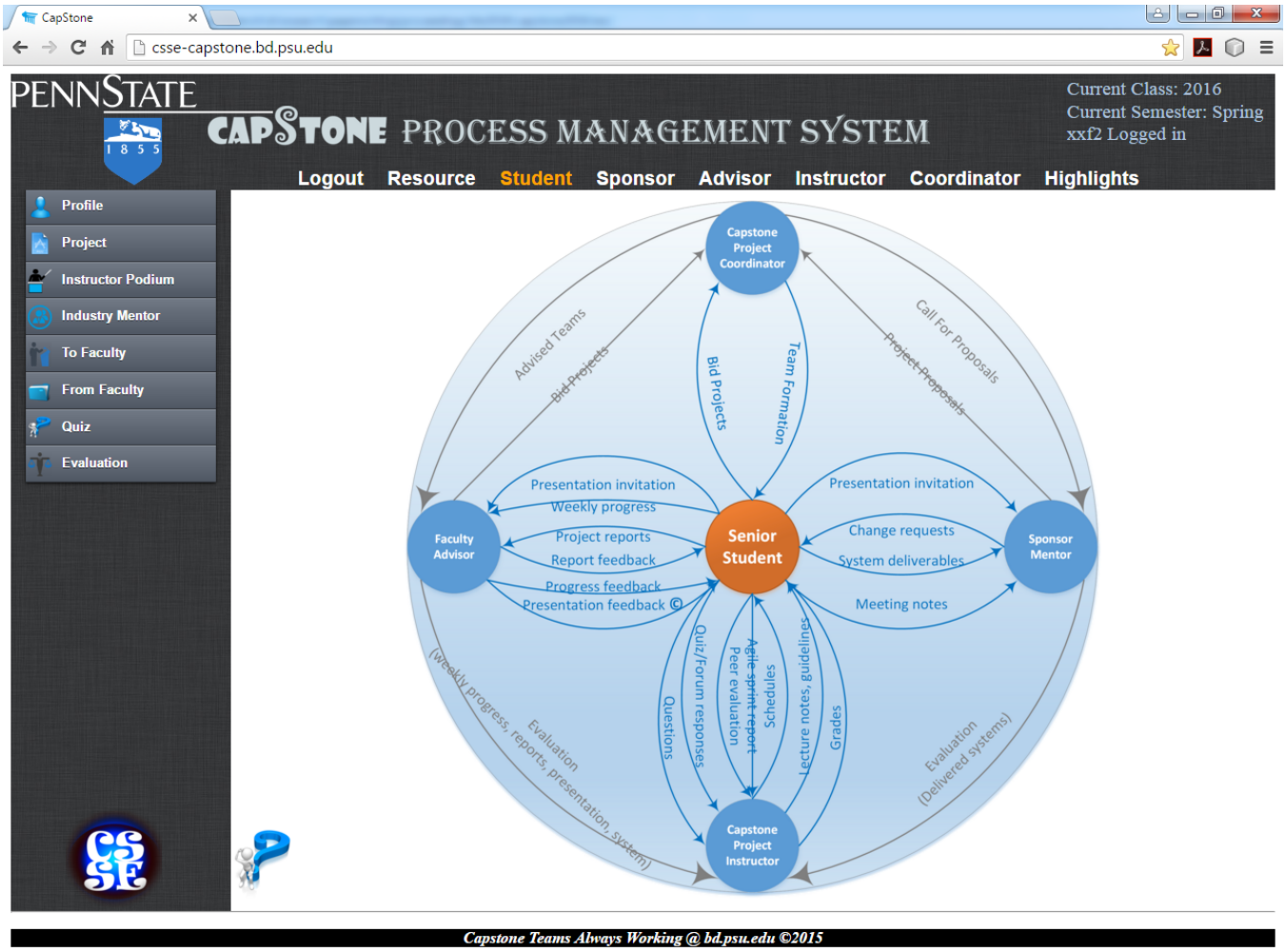

Fig. 1. Mesh of Role Responsibilities in the CapStone Platform.

increasing number of paper-based project artifacts.

4) Project traceability: Without a CASE tool, it's extremely time-consuming for both students and faculty to trace related project artifacts.
5) Teamwork: Key characteristics of high-performing human teams include progress awareness, task awareness and proactive communication [16], [17]. Obviously, lack of proactive communication is common in capstone teams. For instance, it should be the students, but in many cases it's the advisor who must take the initiative to gather information on students' progress [14]. Even worse, many students do not know exactly where they or their teammates are in the development process.

In order to promote student success, highly demanded are platforms that can effectively facilitate multi-party collaboration during the development process of capstone projects.

III. CapStone: A Collaboration Platform

CapStone–a web-based collaboration platform–is our response to the difficulties as explained in Section II.

The design rationale of CapStone is to mesh most, if not all, of the role responsibilities into one common platform such that all parties involved in a capstone project could effectively orchestrate their activities during the development process.

A screenshot of the CapStone platform is given in Figure 1, where the center part illustrates inter-role relationships. Table I also summarizes the inter-role activities implemented in CapStone. We next explain in detail the key features for supporting different types of users.

A. CapStone Supports for Coordinator

The capstone coordinator has the following responsibilities:

(1) sending call-for-proposals to potential sponsors,
(2) managing incoming project proposals,
(3) sifting the proposals for students to bid, and
(4) forming capstone teams upon bids for projects received from both students and faculty advisors.

Every year before the fall semester starts, the capstone coordinator needs to send out call-for-proposals to potential industry sponsors. CapStone maintains the contact information of all the industry sponsors who have indicated interests in, or

TABLE I
INTER-ROLE ACTIVITIES AND SHARED PROJECT ARTIFACTS

	Capstone Coordinator	*Industry Mentor*	*Student Teammate*	*Faculty Advisor*	*Faculty Co-Advisor*	*Instructor*
Capstone Coordinator	Ⓐ	Ⓑ Ⓕ	Ⓕ	Ⓕ	Ⓕ	
Industry Mentor	Ⓒ		Ⓜ Ⓝ			Ⓜ
Student Teammate		Ⓖ Ⓣ	Ⓙ Ⓞ Ⓟ Ⓡ Ⓢ	Ⓙ Ⓛ Ⓚ	Ⓚ	Ⓙ Ⓛ Ⓞ Ⓠ
Faculty Advisor			Ⓜ			Ⓜ
Faculty Co-Advisor			Ⓜ			Ⓜ
Instructor			Ⓓ Ⓔ Ⓗ Ⓘ	Ⓓ Ⓗ	Ⓓ Ⓗ	Ⓤ
Legend (artifacts or activities)	Ⓐ: user management Ⓓ: project schedules & guidelines Ⓖ: student-mentor meeting minutes Ⓙ: weekly progress reports Ⓜ: evaluation feedbacks Ⓟ: cross-project sharing/learning Ⓢ: design artifacts (use cases, test cases, etc.)		Ⓑ: call for proposals Ⓔ: software engineering principles Ⓗ: evaluation rubrics Ⓚ: project presentation slides Ⓝ: requests for changes Ⓠ: post-project reflection Ⓣ: project deliverables		Ⓒ: project proposals Ⓕ: capstone team formation Ⓘ: quizzes Ⓛ: project technical reports Ⓞ: agile sprint reports Ⓡ: peer evaluation Ⓤ: ABET report generation	

have sponsored, our capstone program before. Via CapStone, sending a call for proposals is as easy as a button click.

In early September, the coordinator reviews all the submitted proposals and select a set of candidate projects for further consideration. A proposal can be filtered out if it demands knowledge and skills that are far beyond the capability of our students, but on the other hand, we highly recommend those proposals that impose a reasonable level of difficulties and challenges.

The set of selected projects are then open to students and faculty members for them to bid. Upon receiving bids from students and faculty, the coordinator will then form student teams. For each team, a project is assigned and the sponsor is contacted to have an industry mentor appointed; also assigned is a faculty advisor and two co-advisors. Project assignment is by no means an easy task [18]–[20]. Factors considered by the coordinator include not only student interests, but also whether they may complement each other in terms of skills and personality characteristics. In particular, to meet the ABET requirements on multi-disciplinary teams, team formation is carefully conducted such that each team consists of both CS and SE major students.

B. CapStone Supports for Industry Mentors

An industry mentor, serving as a real customer of the system under development, plays a significant role in the success of the capstone project [21]–[24]. Briefly, an industry mentor has the following responsibilities:

(1) meeting with the student team to clarify user needs and system constraints,
(2) reviewing intermediate/final system deliverables, and
(3) sending change requests for improvement, if applicable.

It has to be acknowledged that industry mentors are fully engaged in their own jobs. So we advise our student teams to contact their industry mentors only on an as-needed basis. According to our experience, most industry mentors are very supportive. Some even could have weekly meetings with the students, strictly sticking to the best agile practices.

Depending on the protocol set up between a student team and their mentor, a mentor has chances to review the intermediate working system to experience high-priority features at an early stage. If defects are detected, via CapStone the mentor can send desirable changes to the students. The change requests are maintained in CapStone as formal records so that both the students and the mentor can easily update and track the status of change implementation.

An industry mentor is deemed as a representative of the sponsoring organization; those mentoring this year's projects are likely to propose new projects in the coming years. This is why in Figure 1 and Table I submitting project proposals is an activity associated with the industry mentor role.

C. CapStone Supports for Faculty Advisors

A faculty advisor has the following responsibilities:

(1) meeting with the student team weekly to monitor their progress and to provide guidance,
(2) reviewing the team's project reports and provide feedback (twice per semester) to students,
(3) attending the team's presentation on project progress and provide feedback (twice per semester) to students, and
(4) sending evaluations on the team's work (presentations/reports/system) to the capstone instructor.

A faculty co-advisor is only responsible for (3).

In order to closely monitor students' progress on their capstone projects, faculty advisors are requested to meet with student teams weekly. To promote proactive communication, every student is asked to submit a weekly progress report to CapStone before meeting his/her advisor. The weekly report contains two parts: a summary of task achievement/challenges in the past 7 days, and a to-do list for the next 7 days. During the meeting time, via CapStone an advisor can (a) load each advisee's weekly report, (b) review technical challenges the student faced and provide guidance, if not resolved, and (c) comment on the student's attitude on teamwork.

To meet the technical writing requirement, each student team needs to submit project reports (twice per semester). After reviewing a project report, the advisor is requested to complete an online evaluation form from CapStone. The advisor's comments become immediately available to the students.

In addition, each student team needs to give presentations (twice per semester) to report their project progress. For each team, an evaluation committee formed by the faculty advisor

and co-advisors need to attend their presentations and provide feedbacks for improvement. Immediately after a presentation, each evaluator is requested to complete an online evaluation form from CapStone. Again, the feedbacks from all three evaluators become immediately available to the students.

The online feedback records as used for both presentation and report evaluation are much easier to maintain than oral or paper based ones. The feedbacks from various sources can be immediately delivered to the students in a combined form. Moreover, this allows an evaluator to easily track whether his/her feedbacks have been considered in the later versions of student work (system/reports).

D. CapStone Supports for Instructor

The capstone instructor has the following responsibilities:

(1) overseeing the general development process and standards enforcement for all capstone projects,
(2) giving lectures on design principles to the whole class of students,
(3) assessing student learning on design principles,
(4) supporting cross-team dissemination of best practices,
(5) collecting project evaluations from faculty (co-)advisors and industry mentors,
(6) collecting teamwork evaluations from each individual student, and
(7) assessing each individual student's capstone performance.

The capstone instructor oversees all capstone projects to ensure that engineering principles and standards are consistently applied. In the fall semester, while all the teams are working on their projects, the instructor needs to give lectures three times a week on design principles and practices, and administer quizzes to evaluate their understanding. Major themes covered in lectures include code of ethics, software process models, UML, design patterns, version control, and automated testing. It's worth mentioning that some topics might have been covered one way or another in lower-level courses. However, senior design course is quite different because it offers the best opportunity to help students mesh everything they have learned and systematically apply their knowledge and skills to a team project with real customers.

In the past, we adopted the waterfall process model for capstone projects for a good reason. Students majored in EE, CE and SE used to form multi-disciplinary teams to work on projects that typically involve hardware design. The waterfall model fits well because it is extremely costly to start implementation before a clear list of requirements is settled and a solid system design is validated.

To foster a better cross-learning environment for software-intensive projects, we now have a new capstone program dedicated to computer science and software engineering disciplines. This allows students majored in CS and SE to practice the agile process model, which is widely adopted in software industry [25], [26].

When it comes to the spring semester, every team should have made a significant progress on their projects. At the same time, each student might have self-learned a few useful techniques or skills that are essential to the project development. Imagine! This is the time when they are eager to share what makes them feel proud. To promote cross-learning, we have scheduled a capstone forum where every week one or two teams have the chance to share their practices.

In order to assist a capstone instructor to fulfill the above-mentioned responsibilities, the following features have been implemented in the CapStone platform.

(1) For process control, the instructor can set key milestones along the two-semester process. For quality control, capStone helps the instructor to consistently maintain rubrics for evaluating student presentations and project reports. All evaluation rubrics in CapStone are carefully designed with performance indicators tied with the ABET program outcomes [27], [28]. In addition, the instructor has access to all the teams' agile development plans and sprint reports periodically updated by students. This allows the instructor to track each team's progress, critical changes and their implementation or lack of.
(2) The instructor can upload lecture notes and resources, which are accessible to all students.
(3) CapStone maintains a question bank. The instructor can add new questions and create new quizzes. ABET reports with grading statistics can be automatically generated immediately after a quiz.
(4) Every time before a capstone forum starts, from CapStone the instructor can gather the presentation slides prepared by the student speakers, and make the materials accessible to all students.
(5) The instructor has access to the evaluations on all the capstone projects submitted by faculty (co-)advisors and industry mentors.
(6) The instructor has access to the peer evaluations made by each individual student on his/her teammates [29].
(7) CapStone can synthesize all the evaluations and recommend a final grade for each individual student.

E. CapStone Supports for Students

Each student has the following responsibilities and they are all supported in the CapStone platform:

(1) taking quizzes on design principles,
(2) bidding for capstone projects,
(3) reporting his/her progress and challenges before the scheduled weekly meeting with the team's faculty advisor,
(4) after each meeting with the team's industry mentor, sharing a formal record of the meeting minutes with both the industry mentor and faculty advisor,
(5) monitoring change requests from the industry mentor, and updating its status as a change has been implemented,
(6) scheduling presentations (twice per semester) to update project progress to faculty advisor and co-advisors,
(7) submitting project technical reports to the faculty advisor for comments,
(8) sending intermediate/final system deliverables to the industry mentor for review,

(9) submitting agile sprint reports bi-weekly to the instructor,
(10) disseminating best practices to the whole class in capstone forums,
(11) evaluating the teamwork performance of each teammate,
(12) reporting lessons learned upon the completion of the capstone project.

As explained in Section III-D, the agile development approach has been adopted for all our capstone projects. The agile process starts in mid September and ends in mid April of the next year. This period of time is further split into 10 agile sprints: one inception sprint and 9 construction sprints. Figure 2 highlights the focus of these agile sprints. The inception sprint (Sprint 0) starts in mid September and ends in mid October. During this sprint, each student team need to work with their industry mentor to elicit an *initial set* of user requirements, which are then transformed into system requirements. The hope is to have most, if not all, of high-priority system features and constraints identified in the early stage of development.

Each construction sprint lasts for about 2 weeks. At the end of each construction sprint, a team is asked to demonstrate their working system to both their faculty advisor and industry mentor. As they see how the system-in-progress works, they are able to provide students with constructive comments, which are then taken into consideration as students plan their design and implementation tasks in the next construction sprint. It is typically at the end-of-sprint review time that the industry mentor would have chance to further clarify the *initial set* of user requirements, and more importantly, to raise new requirements.

In addition, to further encourage cross-team learning, we have coined a term called "agile buddy." Let's say, there are three teams T_1, T_2, and T_3. Each team has one student designated as that team's agile buddy (or ambassador). Let's refer to them by A_1, A_2, and A_3, respectively. A_1 is responsible for reviewing the system-in-progress of T_2, A_2 is responsible for reviewing T_3's system, and A_3 is responsible for reviewing T_1's system. In so doing, each team can acquire an outsider student's opinion on their system. Furthermore, each team, say T_1, has a chance to collect information from two other teams: A_1 can bring back team T_2's progress and practices, and T_1 can get to know T_3's progress and practices from A_3. This not only helps to disseminate best practices, but also to drive teams who have lagged behind their schedules to speed up.

To support development activities in the agile process, CapStone has the following features:

(1) Students can create and update their agile plans (including task splittings among team members).
(2) At the end of each sprint, students can input their reflections on the agile practices, and collect system review comments from faculty advisors, industry mentors, as well as agile buddies, if available.
(3) Students can document and update the design artifacts of their systems, including requirements, use case, test cases, test execution reports, etc.

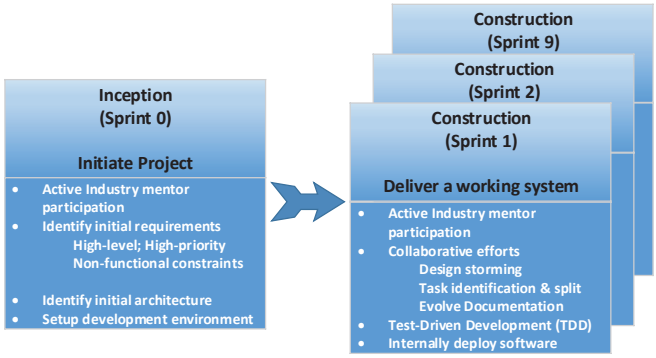

Fig. 2. Focus of Agile Sprints.

Along the development process, each team need to work on a series of technical reports to document important project achievement. Required are 4 major versions (V1.0, V2.0, V3.0, V4.0) and 4 minor versions (1.5, 2.5, 3.5, 5.0). Each major version represents a milestone, with a significant amount of new materials added. A minor version is used as a means to change enforcement: the team earn extra credits if in it they have made changes as suggested by their faculty advisor in the latest major version. From project management perspective, CapStone makes it easier for students and advisors to locate and review various versions of project reports.

Another objective of CapStone is to assist a student team to build stronger teamwork awareness. Towards this goal, any inputs from a student are made accessible to all his/her teammates. In particular, weekly progress reports, meeting minutes, agile plans and sprint reports are all shared, from which a team oriented student can always obtain a reasonable level of awareness about another teammate's task assignments, progress, and existing challenges. As far as proactive communication is concerned, CapStone cannot do anything better than human habits. But CapStone does have a reminder subsystem such that messages about important deadlines can be automatically sent out to the right persons a few days ahead.

IV. CapStone in Use: Student Experiences

In August 2015, the CapStone platform has been deployed in the CSSE department at Behrend College, Pennsylvania State University. Since then, all types of users involved in a capstone project have been using the system to support them in fulfilling their respective responsibilities. In this paper, we focus on the student role and report the experience of our students on CapStone.

There are 33 students in the class, where 18 of them are Software Engineering students, and the rest are Computer Science students. In total, 11 teams were formed, each consisting of both CS and SE major students. A survey was conducted in Feb, 2016, when the students had intensively used the CapStone platform for about 6 months. All students except one responded. Figures 3 and 4 plots student responses to the survey questions.

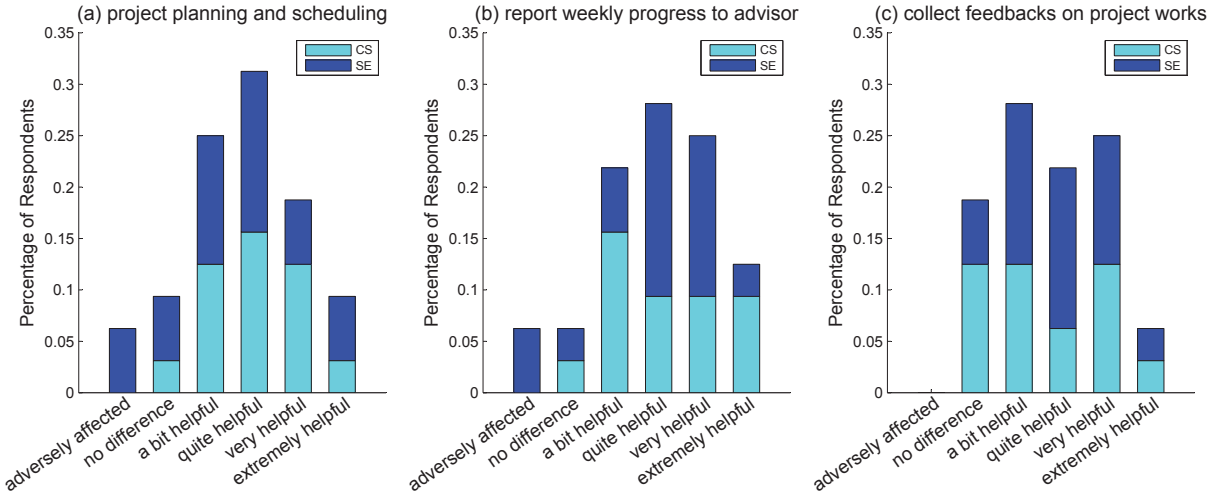

Fig. 3. Student Responses to Survey Questions.

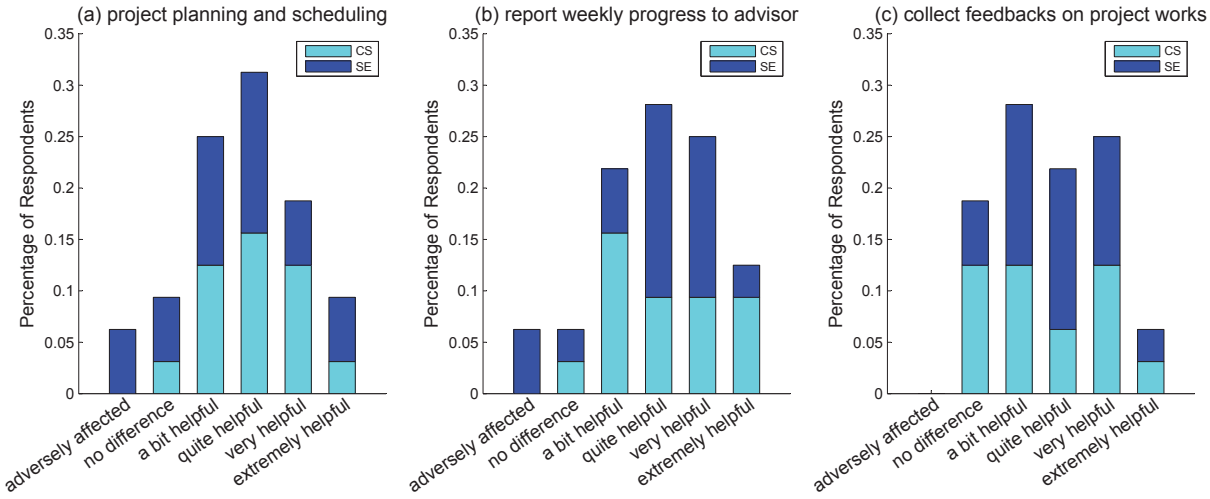

Fig. 4. Student Responses to Survey Questions.

In Figure 3(a), in response to how well CapStone helped in project planning and scheduling, 2 students said they were adversely affected, 3 were neutral, and 27 (about 84.4%) were positive. It's interesting to note that CS and SE students concurred at several rating levels; in particular, the majority opinions were both at the 'quite useful' level.

Plotted in Figure 3(b) are responses to how well CapStone helped students to report weekly progress to advisors. While majority of SE students rated CapStone as 'quite useful', majority of CS students rated CapStone at a lower level: 'a bit useful'. Overall, 87.5% of students had positive experiences.

Plotted in Figure 3(c) are responses to how well CapStone helped students to collect feedbacks on project works. There was no negative rating, and about 84.4% students rated CapStone as useful. In comparison, more SE students liked CapStone at each of the four positive levels.

Plotted in Figure 4(a) are responses to how well CapStone helped students to keep track of the progress of teammates. As far as progress tracking is concerned, the supports from CapStone include the sharing of weekly progress report, meeting minutes, and agile sprint reports. It's a surprise that only about half of the students reported positive whereas the other half felt neutral. Some students admitted that their teams actually used some social network applications such as Facebook to update each other's progress. While the students were encouraged to use any means of communication, they didn't recognize the need to sync with other people, the instructor and faculty advisors, who might have been using CapStone as the only platform to monitor their progress.

As shown in Figure 4(b), in response to how well CapStone

helped them to collaborate on project design tasks, about 60% reported positive, among which 35% of the responses were at the level of 'quite helpful' or higher. Additionally, CS students concurred with SE students at all levels.

Plotted in Figure 4(c) are responses to how well CapStone helped students to write project technical reports. CapStone allows students to export their design artifacts, which can be inserted into their project reports. About 85% students reported it as a useful support, among which 50% of the responses were at the level of 'quite helpful' or higher.

V. Conclusion

In sum, CapStone is a cloud-based application that has been developed and deployed at the Behrend College, Pennsylvania State University, to facilitate multi-party collaboration during the development process of capstone projects.

The design rationale of CapStone is to mesh most, if not all, of the role responsibilities into one common platform such that all parties involved in a capstone project could effectively orchestrate their activities.

To facilitate project management, CapStone is a center for reporting weekly progress, for submitting and version-controlling project works, for scheduling presentations, and for tracking project schedules, agile plans, meeting minutes, and requests for changes from industry mentors. CapStone has a feedback subsystem where faculty advisors and industry mentors can closely monitor students' progress on their capstone projects and provide timely feedbacks for further improvement. CapStone also has an evaluation subsystem where all evaluation rubrics are carefully designed with performance indicators that are tied with the ABET program outcomes.

Last but not the least, CapStone has an open interface that allows the integration of tools for supporting major-specific design activities. For instance, in the Department of Computer Science and Software Engineering, the main output of a capstone project is a software system that can meet the user needs as described in the project proposal. To support software development activities, CapStone has integrated a few useful CASE tools for students to document and export project artifacts such as requirements cards, use cases, agile reports, test cases, and test execution reports. This can help to enforce uniform quality control in project report writings.

We have also reported a study on how well CapStone could help student users in their project development. In many aspects, our students had quite positive experiences. In particular, over 80% of the students found CapStone helpful in project planning/scheduling, reporting weekly progress to advisors, collecting feedbacks on project works, and in writing project reports. The study also suggested some areas for improvement. For example, in its support for teamwork progress tracking and design task collaboration, CapStone can be further enhanced such that those students who felt neutral could benefit more from it.

References

[1] J. L. Ray, "Industry-academic partnerships for successful capstone projects," in *Frontiers in Education, 2003. FIE 2003 33rd Annual*, vol. 3, Nov 2003, pp. S2B–24–9 vol.3.

[2] R. M. Ford and W. C. Lasher, "Processes for ensuring quality capstone design projects," in *Frontiers in Education, 2004. FIE 2004. 34th Annual*, Oct 2004, pp. S2G–12–17 Vol. 3.

[3] D. Ingalsbe and J. Godbey, "Project-oriented capstone course: Integrating curriculum assessment utilizing industry partner and student input," in *Proceedings of the ASEE Annual Conference and Exposition*, Austin, 2005.

[4] J. V. Tocco and D. D. Carpenter, "Re-engineering the captsone: Melding an industry oriented framework and the BOK2," in *Proceedings of the ASEE Annual Conference and Exposition*, Vancouver, BC, Canada, 2011.

[5] R. K. Stanfill and A. Rigby, "The professional guide: A resource for preparing capstone design students to function effectively on industry-sponsored project teams," in *Proceedings of the ASEE Annual Conference and Exhibition*, Indianapolis, 2014, p. 22.

[6] C. Steinlicht and B. G. Garry, "Capstone project challenges: How industry sponsored projects offer new learning experiences," in *Proceedings of the ASEE Annual Conference and Exposition*, Indianapolis, 2014, p. 11.

[7] P. M. Griffin, S. O. Griffin, and D. C. Llewellyn, "The impact of group size and project duration on capstone design," *Journal of Engineering Education*, pp. 185–193, 2004.

[8] K. F. Li, A. Zielinski, and F. Gebali, "Capstone team design projects in engineering curriculum: Content and management," in *Teaching, Assessment and Learning for Engineering (TALE), 2012 IEEE International Conference on*, Aug 2012, pp. T1C–1–T1C–6.

[9] J. W. Jones and M. Mezo, "Capstone = team teaching + team learning + industry," in *Proceedings of the ASEE Annual Conference and Exposition*, Indianapolis, 2014, p. 7.

[10] C. W. Ferguson and P. A. Sanger, "Facilitating student professional readiness through industry sponsored senior capstone projects," in *Proceedings of the ASEE Annual Conference and Exposition*, Vancouver, BC, Canada, 2011.

[11] J. Ceddia and J. Sheard, "Evaluation of WIER - a capstone project management tool," in *Computers in Education, 2002. Proceedings. International Conference on*, Dec 2002, pp. 777–781 vol.1.

[12] A. Erradi, "easyCapstone: A framework for managing and assessing capstone design projects," in *Computer Science Education (ICCSE), 2012 7th International Conference on*, July 2012, pp. 1345–1350.

[13] D. C. T. Lo and O. Karam, "Enhance capstone projects with a new online collaboration system," in *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*, July 2013, pp. 217–218.

[14] J. J. Olarte, C. Domnguez, A. Jaime, and F. J. Garca-Izquierdo, "A tool for capstone project management in computer science engineering," in *Computers in Education (SIIE), 2014 International Symposium on*, Nov 2014, pp. 65–68.

[15] S. S. Moor and B. D. Drake, "Addressing common problems in engineering design projects: A project management approach," *Journal of Engineering Education*, pp. 389–395, 2001.

[16] M. Borrego, J. Karlin, L. McNair, and K. Beddoes, "Team effectiveness theory from industrial and organizational pychology applied to engineering student project teams: A research review," *Journal of Engineering*, vol. 102, no. 4, pp. 472–512, 2002.

[17] X. Fan, J. Yen, and R. A. Volz, "A theoretical framework on proactive information exchange in agent teamwork," *Artificial Intelligence*, vol. 169, pp. 23–97, 2005.

[18] Y. C. Lan and J. A. Ginige, "Towards criteria based allocation of capstone projects for an enhanced learning experience," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 5, Dec 2008, pp. 121–124.

[19] L. Lopez, M. Aronson, and G. Carstensen, "Optimizing support for senior design project assignments," *Interfaces*, vol. 38, no. 6, pp. 448–464, 2008.

[20] J. E. Burge and G. C. Gannod, "Dimensions for categorizing capstone projects," in *Software Engineering Education and Training, 2009. CSEET '09. 22nd Conference on*, Feb 2009, pp. 166–173.

[21] D. Knudson and A. Radermacher, "Updating cs capstone projects to incorporate new agile methodologies used in industry," in *Software Engineering Education and Training (CSEE T), 2011 24th IEEE-CS Conference on*, May 2011, pp. 444–448.

[22] V. Isomttnen and T. Krkkinen, "The value of a real customer in a capstone project," in *Software Engineering Education and Training, 2008. CSEET '08. IEEE 21st Conference on*, April 2008, pp. 85–92.

[23] J. Vanhanen, T. O. A. Lehtinen, and C. Lassenius, "Teaching real-world software engineering through a capstone project course with industrial customers," in *Software Engineering Education based on Real-World Experiences (EduRex), 2012 First International Workshop on*, June 2012, pp. 29–32.

[24] A. Rusu and M. Swenson, "An industry-academia team-teaching case study for software engineering capstone courses," in *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual*, Oct 2008, pp. F4C–18–F4C–23.

[25] D. A. Umphress, T. D. Hendrix, and J. H. Cross, "Software process in the classroom: the capstone project experience," *IEEE Software*, vol. 19, no. 5, pp. 78–81, Sep 2002.

[26] D. Rover, C. Ullerich, R. Scheel, J. Wegter, and C. Whipple, "Advantages of agile methodologies for software and product development in a capstone design project," in *Frontiers in Education Conference (FIE), 2014 IEEE*, Oct 2014, pp. 1–9.

[27] D. Stevens and A. Levi, *Introduction to Rubrics: An Assessment Tool to Save Grading Time, Convey Effective Feedback, and Promote Student Learning*. Sterling, VA: Stylus Publishing, 2005.

[28] E. Ahmad, B. Raza, and R. Feldt, "Assessment and support for software capstone projects at the undergraduate level: A survey and rubrics," in *Frontiers of Information Technology (FIT), 2011*, Dec 2011, pp. 25–32.

[29] F. Fagerholm and A. Vihavainen, "Peer assessment in experiential learning assessing tacit and explicit skills in agile software engineering capstone projects," in *Frontiers in Education Conference, 2013 IEEE*, Oct 2013, pp. 1723–1729.