Student-Sponsored Projects in a Capstone Course

Reflections and Lessons Learned

Ethan Bütt
University of Nebraska - Lincoln
Lincoln, Nebraska, USA
iethan@huskers.unl.edu

Suzette Person
University of Nebraska - Lincoln
Lincoln, Nebraska, USA
sperson@cse.unl.edu

Christopher Bohn
University of Nebraska - Lincoln
Lincoln, Nebraska, USA
bohn@unl.edu

ABSTRACT

Undergraduate students in all disciplines sometimes struggle to see the connections between what they learn in the classroom and how it will be used in practice. Capstone courses, internships, and co-op experiences can help students bridge this gap by allowing students to put theory into practice on real-world projects and by connecting students to industry practices and partners. These experiences can better prepare students for their careers, but they may also shield students from real-world risks, such as lack of product feasibility or viability, and from the non-technical tasks of business planning and startup tasks, in order to make the academic aspects of the course manageable and scalable. In this work, we present our experience offering undergraduate Software Engineering students the opportunity to self-sponsor a year-long capstone project. Students began the course with four software product ideas, none of which had been assessed for feasibility or viability. During the academic year, students were responsible for product discovery, technology selection, and technical development following good Software Engineering practices. They also created business and marketing plans, and formed an LLC. This paper[1] describes the processes, practices and timeline we followed for a student-sponsored capstone team. We also share our lessons learned and offer suggestions for student-sponsored projects in computing capstone courses.

KEYWORDS

Software Engineering, Computer Science, Entrepreneurship, Capstone, Startup

ACM Reference Format:
Ethan Bütt, Suzette Person, and Christopher Bohn. 2022. Student-Sponsored Projects in a Capstone Course: Reflections and Lessons Learned. In *44nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '22), May 21–29, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3510456.3514160

[1]The authors are one of the students, the director of the School of Computing's undergraduate Software Engineering program, and the director of the School of Computing's capstone program; this paper draws from all three perspectives.

ICSE-SEET '22, May 21–29, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9225-9/22/05…$15.00
https://doi.org/10.1145/3510456.3514160

1 INTRODUCTION

Active learning has been an important part of undergraduate curricula for many years [5, 22]. Experiential learning is a more intensive form of active learning that has been shown to lead to deeper understanding. Students view it as more enjoyable than a didactic approach, and employers prefer to hire students who have an experiential learning background [23]. Project-based learning, a type of experiential learning, has been shown to be effective for improving student engagement and performance in STEM [7]. Capstone courses are often included in undergraduate programs as a means to provide students with project-based learning opportunities. These courses are intended to help students integrate material learned in previous courses while applying their knowledge and skills to a realistic project. Computing capstone courses have been shown to benefit students in various ways, including improving students' estimation and planning skills [13] and by helping students understand the importance and difficulty of team collaboration and communication [18].

For roughly the past ten years, the computing capstone course at the University of Nebraska - Lincoln has offered a combination of industry, government, non-profit, and faculty-sponsored projects. Each year, the majority of project sponsors pay a fee to help cover the costs of resources required to support their projects (e.g., hardware, tools, project management support), while other projects are offered to the sponsor at no cost. The projects begin at the start of the academic year and continue through the spring semester, providing the students with a substantial project experience that spans an academic year. Teams of 5–6 students work closely with the project sponsor and follow an Agile-based software development process. While all students work as developers, some students take on additional roles such as Development Manager or Product Manager. Student teams are monitored by a team of faculty, volunteer coaches, and professional project managers. Instruction in relevant topics is delivered in workshop format (e.g., teams watch a video and then work through an exercise during their team's work time) so that most class meetings are used for weekly sponsor meetings or as "team time" for students to work on their projects. At the end of the academic year, students transition the product to the project sponsor after presenting their work at a capstone showcase.

In Fall 2020, we piloted a single student-sponsored project with nine Software Engineering students. These students were taking the capstone course for the second time, as required by our school's Software Engineering program. In the first year of the capstone course, the students participated in a traditional capstone project, working with their teams and external sponsors. The students' objective in the second year of capstone was to experience the broader scope of tasks and activities that support a software development

project, such as project discovery, market analysis, technology selection, intellectual property protection, and business startup tasks (e.g., formation of an LLC). As a student-sponsored project, the team was not assigned an external project sponsor and they were responsible for providing the necessary resources to support product development. This arrangement protected the team's rights to pursue their product at the end of the course without legal or financial ties to the university.

Our capstone course has a well-defined framework and processes to support a large number (25+) of software development capstone projects each year. However, it was not designed to instruct students in the broader scope of tasks and activities surrounding software product development. Since this extended project scope was beyond what the course normally covers, and also extended beyond the scope of the instructors' professional experiences, we partnered with Don't Panic Labs, a local company that mentors 30–40 entrepreneurs each year in the process of taking an idea for a software product from concept to market. Company volunteers with roles in engineering and product development guided the students through the initial phases of product discovery to the point where students selected a product to focus on for the remainder of the course.

Once a single product was selected for development, the students were able to leverage their previous capstone experience to carry out the development of the software product. The students also received help from our university's legal clinic to work through the legal side of forming a business and preparing for their product's release. These tasks included establishing company bylaws and branding, forming an LLC under which to operate, and writing terms of service for their application. By the end of the academic year, students had developed a business plan, marketing plan, and formed an LLC, all of which helped the students experience many of the activities necessary to create and market a software product. They also made substantial progress on product development, including selecting development technologies and creating and testing interfaces.

2 BACKGROUND & MOTIVATION

Students pursuing an undergraduate degree in Software Engineering at our university follow a core curriculum that is different from our other computing majors. [14] The first two years (four semesters) teach fundamental computer sciences topics (i.e., topics covered in traditional CS1, CS2, and CS3 courses) integrated with Software Engineering topics. Each core Software Engineering course includes a weekly lab section in which students work in pairs to practice concepts taught in class. At the end of the first three core course, students complete a team software development project. The fourth core course incorporates communication skills and provides a one-semester capstone experience in which students work in teams to contribute to an open source software project. The program follows a cohort model and student teams are set by the instructors so that students have an opportunity to work with many of the students in their cohort. After students complete the core courses, they take two years of the capstone course (our other computing majors are required to complete only a single year of this course). In parallel with the capstone course, Software Engineering students also take four additional Software Engineering courses covering topics such as requirements elicitation, software testing, and software design and architecture. Students in our other computing majors complete a traditional computer science curriculum (i.e., CS1, CS2, CS3) and a Software Engineering course prior to enrolling in the capstone course.

In recent years, our capstone course has evolved from instructor-defined projects to paid and unpaid projects sponsored by industry, government, non-profits, and faculty from our department and across the university. A team of faculty (tenure-track and practice faculty), professional project management staff, and a director (who is a member of the faculty) are responsible for soliciting projects, as well as guiding and assessing student work. Volunteer team coaches, usually from industry, provide their business, project, or technical expertise to mentor students as they complete their projects. The program typically solicits 25–30 projects each year based on projected enrollment. Students are assigned to projects based on their project preferences submitted at the beginning of the academic year. Project sponsors are expected to spend 2–3 hours each week meeting or working with their student team.

To ensure the faculty and staff can manage a large number of students and projects each year, all projects follow a standardized development process based on Agile practices taught in previous Software Engineering courses. During the fall semester, the teams of 5–6 students complete three release cycles. At the beginning of each release cycle, students estimate how much work they can complete, and negotiate the deliverables with the project sponsor. At the end of each release cycle, students hold a release meeting to share their progress with the project sponsor. Students are expected to deliver working software at the end of each release cycle. During the spring semester, students complete three additional release cycles. At the end of the year, students present their work at a showcase event for faculty, students, project sponsors, and members of the community.

Working with a project sponsor provides students with an opportunity to learn industry practices and software development in the context of changing and conflicting objectives. They also get the chance to practice non-technical skills, including time management, negotiation, communication, and leadership. Students also experience collaborating with a development team, delegating work, and filling a variety of roles. Students in our capstone course complete projects in web development, mobile apps, data analytics, virtual reality, and a host of other technologies. Software Engineering students taking the capstone for the second time are expected to mentor first-year capstone students in the practices, processes, and technologies used for capstone projects. They are also encouraged to pursue a leadership role, to choose a project with unfamiliar technologies, or to choose a project in a different industry (e.g., agriculture, banking).

In Fall 2020, a group of nine Software Engineering students enrolled in the capstone course for the second time proposed the idea of a student-sponsored project in which they assumed responsibility for the entire project, including the project sponsor role. The students argued they would receive the same benefits in terms of software development experience, but with the added experiences of product discovery and business startup tasks – business and marketing plan development, branding, and navigating the legal

hurdles of forming a corporation. Students wanted to experience a project from the sponsor's perspective to better understand the tasks and activities necessary to develop and deploy software, and ultimately to experience a "startup environment" in its infancy. The students argued that if they were making decisions regarding which product to work on and the technologies used to the build the product, they would be more motivated to work on the project and to give it their best efforts.

Given this expanded scope, our capstone course needed additional resources if we were to allow a student-sponsored project. The business college offers courses in business law, entrepreneurship, and other relevant topics, but requiring students to take additional courses was not practical given the short timeframe between when the students proposed the idea of a student-sponsored project and the start of the fall term. Moreover, our undergraduate Software Engineering program is highly prescriptive and does not include open electives for students to take courses outside their required courses. Instead of using university resources for the product discovery process, we secured the support of a local industry partner, Don't Panic Labs (DPL) [12]. DPL specializes in helping technology entrepreneurs through the product discovery process. Members of the company served as volunteer coaches to help the students determine the "best" idea to pursue as their company's first software product. To help with the legal aspects of their project, the students contacted the university's law school and were connected with a legal clinic that provides third-year law students with practical experience handing early-stage legal matters for startup businesses throughout the state. With this team of faculty, legal experts, and software startup specialists in place, the student-sponsored team had the support necessary to move forward with their software product ideas.

3 RELATED WORK

Many universities offer (and often require) a capstone course for undergraduate students, especially for students in computing programs [8, 10, 19]. However, student-sponsored projects within a capstone course appear to be rare. Among the universities in our academic conference (2020 Big 10 Universities), none list student-sponsored projects on their website or senior capstone course descriptions, as shown in Table 1. After expanding our search, we found only one university among roughly 40 that we explored with this option. This is somewhat surprising, given the reported success from the seven "student-initiated" projects (out of 20 total projects) at a university in Colorado: all reported a great value to student participants, and all seven delivered a valuable product [15].

There are many experience reports describing how to support and deliver capstone experiences and the benefits of capstone projects [3], and how these experiences can narrow the gap between education and industry [1]. However, Devadiga notes that Software Engineering education is often focused on providing students with knowledge and skills to work in mature organizations, whereas startup organizations require different skillsets that are not necessarily covered in a traditional Software Engineering curriculum [4]. This has led a number of universities with Computer Science programs to pursue new ways to connect their capstone projects with industry work. Our work specifically addresses this issue by providing students with an opportunity to experience an expanded project scope that includes product discovery and other startup activities typically handled by a capstone project sponsor. By pursuing this option in their second year of the capstone course, students not only get a second opportunity to apply their Software Engineering knowledge and skills, but they also learn how the uncertainties of a startup organization impact the engineering of software.

In a report by Chanin et al., four Mathematics and Computer Science students at Western Carolina University report how their CS program spent three years developing a two-semester capstone project that places an emphasis on project management and product domain research, in addition to technical abilities [2]. They describe how these projects can be used to better prepare students for all steps of the software development cycle: software testing, version control, pitch presentations, etc. The students credited their capstone experience with helping them improve their soft skills and with their job search. Although their overall experience was positive, they noted that "a number of students... felt the capstone course could be improved in regards to development skills." Primarily, many students noted the lack of guidance or mentoring from faculty or others outside their team. Our work is similar in terms of what was expected of students; however, we actively sought help from other specialists to provide mentoring support in areas such as business law, marketing, and project discovery, to ensure students had sufficient resources to complete a student-sponsored project.

In another capstone experience report written from a student's perspective [6], the author examines his own experience in a Software Engineering capstone course and suggests possible improvements. One of the most significant points for improvement made by the author centers on the value added by expanding the breadth of a student team's project. Although the capstone experience described by the author is dissimilar to our student-sponsored team in significant ways– for instance, our capstone course is two semesters instead of a single semester, the argument to expand projects to allow students more freedom and learning opportunities in order to give them "stronger Software Engineering skills" and better "real-world training," is similar to the argument the students on our student-sponsored project used to justify an expanded project scope. In some ways, our report stands to validate some of the proposed improvements in [6].

Factors influencing student motivation and student success in capstone projects are addressed in several studies. For instance, Ikonen et al. sought to distill so-called "High Impact Success Factors" for capstone projects – properties of teams or operational decisions that lead teams to successful outcomes, identified the makeup and experience of teams as a crucial factor in that team's success [9]. Two key elements called out by this report are a team's ability to communicate and remain unified in their goals. The students on our student-sponsored team had the advantage of three years working together in prior teams and courses; this allowed them to begin their project with an established standard of communication and other processes. Many of the "high-impact success factors" outlined in [9] were present in our student-sponsored team to some degree, including alignment of prior skills and good communication, allowing the team to operate with cohesion. These factors did not

University	CSE Major and Senior Capstone Outline/Requirements
University of Illinois Urbana-Champagne	https://mechse.illinois.edu/undergraduate/senior-capstone-design-program
Indiana University - Bloomington	https://engineering.indiana.edu/courses/class/iub-fall-2020-engr-e490
University of Iowa	https://cs.uiowa.edu/undergraduate-programs/cse-requirements
University of Maryland	https://undergrad.cs.umd.edu/degree-requirements-cs-major
University of Michigan	https://cse.engin.umich.edu/wp-content/uploads/sites/3/2020/10/cse_guide.pdf
Michigan State University	https://www.cse.msu.edu/~cse498/2021-01/home/
University of Minnesota	https://cse.umn.edu/cs/sponsor-capstone-project-course
Northwestern University	https://www.mccormick.northwestern.edu/computer-science/academics/courses/descriptions/362.html
Ohio State University	https://engineering.osu.edu/corporate-engagement/capstone-projects
Penn State	https://www.ime.psu.edu/industry/capstone-design-projects.aspx
Purdue University	https://engineering.purdue.edu/IE/corporatepartners/capstoneprojects
Rutgers School of Engineering	https://mae.rutgers.edu/capstone-design-project
University of Wisconsin-Madison	https://www.engr.wisc.edu/news/the-full-package-capstone-course-for-seniors-ties-materials-science-together/
University of Nebraska-Lincoln	http://cse.unl.edu/senior-design

Table 1: Big 10 Computer Science and Engineering Capstone Course Links (all resources accessed in Dec 2020 - Jan 2021)

guarantee the team's success in every facet of the project, but it did help them start development quickly and effectively.

In another report, the authors describe a capstone course involving over 100 students across 14 teams [17]. While the report is largely focused on the companies that work with student teams, it aims to address the motivation of all capstone participants. Based on student and industry surveys, the report highlights the different motivating factors for capstone participants and how they affect performance. The report notes that most students are highly motivated to perform well in their projects because of the practical experience they can gain; the relationship seems to go both ways: those students that are highly motivated tend to reap greater results from their experience. The capstone experience reported in this work had similar results. Our students experienced greater motivation in their project, as they saw the practical application of their work. Additionally, students reported learning more from the experience when they were more highly motivated to engage in it. Overall, this led to a better experience for students and a better end product.

Paasivaara et al. found that students' attitudes toward their work was shown to improve after students participated in realistic capstone projects [18]. The study notes that "[they] have observed that participating in such a course seems to have an effect on student attitudes related to software engineering topics, as they gain experience from a simulated, yet very 'real' project." Students in the study spent six months working on their projects, learning new technical and other skills such as how to participate in Scrum ceremonies. The study also revealed a lot about which aspects of an immersive capstone course were helpful or difficult for students. Although the work of students in our report was done without an industry sponsor, it is similar in terms of working in a simulated, yet realistic environment, and students in both studies experienced a positive impact on their attitudes towards their future careers.

In an analysis of ten years of teaching a computing capstone course offerings, Khmelevsky describes the course outcomes and feedback from students who participated in these courses [11]. The report indicates many of the most difficult aspects of these capstone courses for students resulted from a lack of team cohesion (e.g., teamwork, leadership, communication) and lack of Software Engineering experience (e.g., software testing, project management). In our computing programs, students are required to complete at least one Software Engineering course before enrolling in a capstone course. Based on our experience, we recommend that students who want to pursue a student-sponsored project also complete a year of capstone with an external sponsor before joining a student-sponsored team. These two recommendations are primarily due to the results noted in [11], as these experiences set up student teams to succeed in their own projects.

It has been argued by some that undergraduate Software Engineering curricula should begin by exposing students to the 'big picture' early to help students establish a context for what they are learning, rather than delivering foundational courses (such as Math) [20]. The same principle can be applied to a capstone course. Students who work on an industry-sponsored project are exposed to industry processes, tools and practices, but are typically shielded from the business side of the project and rarely see how software development fits into the broader context of the sponsor's business activities and processes, nor how their work impacts the sponsor's organization. By providing the opportunity for a student-sponsored project in a capstone course, students gain exposure not only to software engineering practices, but they also have opportunities to work on the broader scope of activities performed by a project sponsor prior to entering the workforce. Moreover, they learn to manage the complexities of the business context surrounding software development and enjoy a significant amount of control over the organization's future and success, but in a low-risk and supportive environment.

4 KEY FEATURES OF OUR STUDENT-SPONSORED CAPSTONE PROJECT

Our capstone timeline, practices, and processes have been fine-tuned over time to create a framework that efficiently supports a large number of students working in teams to complete projects with external project sponsors (i.e., sponsors who are typically not the course instructors). Consistent deadlines and deliverable requirements across projects help faculty and professional project managers manage course and student expectations, and the demands of fair and timely grading. However, our capstone framework was designed to support only the software development aspects of the projects, leaving the project sponsor to provide support for the broader scope of tasks related to how the software developed by the students fits into their organization. Requiring a student-sponsored team, with the additional responsibilities typically handled by an external project sponsor, to adhere to existing deadlines and deliverables was unrealistic.

In Table 2, we summarize the key differences between a traditional capstone team and the approach the student-sponsored team chose (with faculty approval) in terms of roles performed by students and faculty, student time commitments, and how the two types of teams operate in general. In the remainder of this section, we provide additional details regarding the timeline and activities of the student-sponsored project, and the ways in which we approached assessing the students' achievement of the learning outcomes.

4.1 Timeline & Activities

A key difference between the student-sponsored project and a traditional capstone project is the timeline of activities and project deliverables. Traditional capstone projects follow a pre-defined release cycle defined by the course instructional team – three releases in the fall and three project releases in the spring. The deliverables for each release are negotiated with the project sponsor. The student-sponsored team did not follow a pre-defined release cycle, nor did they have pre-defined deliverables. Moreover, the student-sponsored team served as both the producer and the consumer of their software product, and therefore had control over whether to continue development of an artifact or to scrap what they had built. They also had the flexibility to choose whether or not they would continue to develop and market their software at the end of the year. Despite this flexibility, the student-sponsored project team was still required to produce artifacts for assessment and to show sufficient progress was being made on their software product. The student-sponsored team was also required to meet weekly with faculty advisors, similar to traditional capstone teams meeting weekly with project sponsors. At each faculty advisor meeting, students were required to demo product functionality and project documentation (e.g., JIRA issues, task boards).

A traditional capstone team typically spends the fall semester getting to know the members of the team, building a working relationship with the project sponsor, soliciting project requirements, becoming familiar with technologies that will be used to develop the product. Towards the end of the semester, students begin developing fixes or features for the sponsor. Unlike a traditional capstone project, the students on the student-sponsored team already knew each other because they had studied together for three years and had worked on course projects together in their Software Engineering and Computer Science courses. As a result, the team formation process had already been completed prior to the beginning of the semester. The student-sponsored team also did not need to spend time getting to know the project sponsor or the sponsor's expectations. Instead, they spent the majority of the fall semester (August–December) on project discovery, as shown in Figure 1.

The students began the fall semester with four ideas for software products. The students believed each idea was feasible, but had not performed a product analysis nor started to develop the product. To help the team identify a single product to pursue during the capstone course, volunteers from Don't Panic Labs led the students through a product discovery process. The DPL product discovery process is based on their experience helping 30–40 local entrepreneurs assess the feasibility and viability of software project ideas each year.

To expedite the process of evaluating four product ideas, the students divided themselves into smaller teams of 2–3 students, and each team worked through the process for one of the four ideas. While the project discovery process looked slightly different for each product, each team conducted market research, competitor analysis, and prospective user interviews. Teams researched similar products or solutions to their own, and sat down with prospective users to determine their priorities and needs. Their goal was to determine if each product was profitable or marketable as well as technically possible. This research would lay the foundation for setting development priorities for the team. Two project ideas were dismissed early in the discovery process because they did not appear to be as marketable or financially viable as originally thought. The remaining two product ideas were deemed viable and were assessed further in the next phase of the project: business plan development.

During business plan development, the students selected which software product to pursue. Their decision was based heavily on user and market research, which allowed them to move forward confidently with their selected product. Since the team did not complete the product selection task until near the end the fall semester, the team did not produce any software deliverables in the fall semester. Instead, the team prepared a set of objectives and expected deliverables for the spring semester. These objectives and deliverables were similar to what an external sponsor would provide to a traditional capstone team at the beginning of the academic year. As a result, the student-sponsored team ended the first semester in a state similar to the state a typical capstone team would begin the fall semester. This left the student-sponsored team with only one semester to develop their chosen product.

During the spring semester (January–May), the student-sponsored team focused on software development tasks. One of the challenges early in the semester was aligning team members' view of the end goal. Students who had spent the discovery phase working on the product idea that was ultimately selected for development had a much deeper understanding of the product vision. To help develop

	Traditional Capstone Team	Student-sponsored Team
Student Roles	Development Manager, Product Manager, Developers	Sponsors, Development Focus Managers, Product Managers, Developers
Non-student Roles	Project Manager, Tribe Lead, Project Sponsor(s)	Faculty Advisor(s)
Releases (across Academic Year)	6 pre-set, evenly-spaced releases	1 final release, with option for smaller releases as set by students and faculty advisors
Business Goals	Not addressed by student team	Created entirely by student team
Starting Point	Idea provided to students by sponsor; sponsor may also provide existing software	Students start from scratch
Development Practices	Set by student team	Set by student team
Team Worktimes	Entire team meets 8-10 hours per week	Entire team meets 2-3 hours per week; smaller sub teams meet an additional 8-10 hours per week
Development Technology	May be provided by sponsor or set by team, depending on project	Selected by student team
Sponsor Meetings	Student team meets with sponsor weekly	Student team meets with faculty advisor(s) weekly
Macro-goals	Set by sponsor(s)	Set by student team
Day-to-day Project Management	Managed by student team	Managed by student team
Course Lectures & Workshops	Students attend	Students do not attend (they took part in their first year)
Resources	Capstone or sponsor provide resources (e.g., software & hardware)	Capstone does not provide resources

Table 2: A Comparison of a traditional capstone project with a student-sponsored team

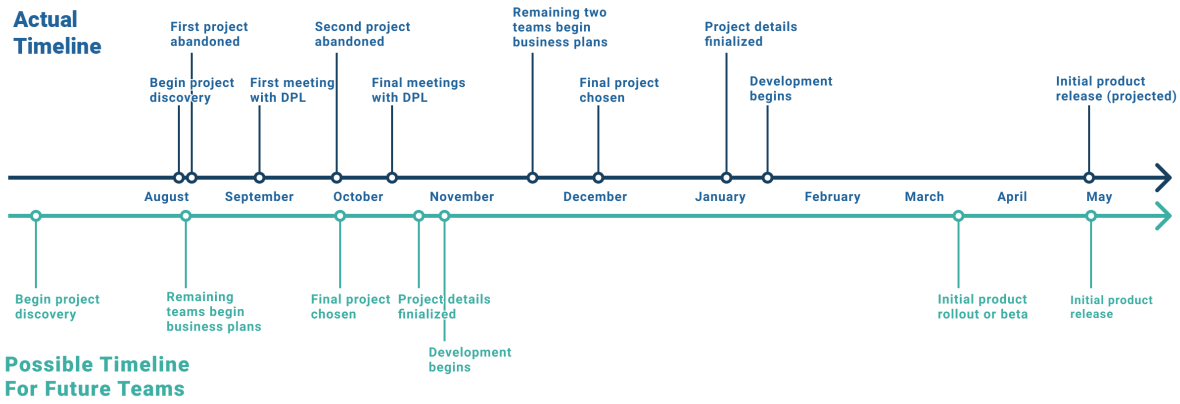

Figure 1: Student-sponsored project actual timeline versus proposed future timeline

a shared understanding of the product across all nine team members, the team engaged in product and story mapping activities. Once the project vision and scope were agreed upon, technologies were selected, development roles decided, and the project management framework (e.g., Jira environment) was established before starting concrete development. Technical tasks completed during the spring semester included the development of user interfaces, database design and implementation, and web and mobile interface development. The goal at the beginning of the semester was to deliver a minimal viable product (MVP) by the end of the term (in approximately four months). During this time, the team also conducted user interviews and usability tests, studied business formation processes, and continued to manage their project goals and timeline – tasks often completed by the project sponsor for a traditional capstone team. At the end of the semester, the team delivered prototypes of their product in the form of a web app and a mobile app. Similar to traditional capstone teams, the student-sponsored team also delivered a release presentation at the end of the semester highlighting the features of their software product and their value to the customer. They also presented their strategies for marketing and selling their product, which represented the business tasks completed as part of their expanded project scope.

4.2 Assessment of a Student-sponsored Team

The students on the student-sponsored team were assessed similarly to the students on a traditional capstone team. Course grades are based on performance evaluations (45%), team behaviors and process (20%), role assignment (20%), individual instructor assessment (10%), and technical presentation (5%). Performance assessment used the same general model we do for other captone projects, in which we consider the team's communication, execution, planning, and vision. During the product discovery phase, the assessment was

tailored to the tasks and deliverables that we and our industry partners asked for; after product discovery, the assessment more-closely resembled our assessments of traditional capstone teams. Although we observed typical capstone behaviors, we focused on behaviors associated with being a "startup" organization: market analysis, data-driven product selection, business planning, performing the necessary steps to form an LLC, etc. For the student-sponsored team, we also maintained the grading cycle of three grading periods per semester, though this was more for grading convenience than for purposes of tying grades to release milestones.

It should be noted that the formation of an LLC was not a requirement for the student-sponsored team. The students could have chosen another organizational structure or business entity. The primary objective was for the students to pursue business objectives in support of the software product they were planning to develop. These objectives included forming an organizational structure to support their product, and selecting an appropriate mechanism to protect their intellectual property. For instance, the students could have selected to provide their software as an open-source project or a non-profit, but concluded that an LLC was the best path based on their market and product research.

4.3 Other Considerations for a Student-sponsored Project

Capstone projects teach students many valuable non-technical skills, such as teamwork, time management, conflict management, and communication skills. Since students on the student-sponsored project team had already completed one year of the capstone course, they had already been exposed to, and learned how to, manage many of the challenges of team software development. However, there were still valuable non-technical skills to be learned in a student-sponsored project. Notably, students learned the importance of project management in the context of financial concerns, the challenges of working with end-users who do not have a stake in the project, and the challenge of balancing academic, business, and technical tasks within a tight timeframe.

In a traditional capstone project, students have little to no concern for financial support or profit, since the sponsor manages the financial concerns associated with the product. However, in the case of a student-sponsored project, financial issues impact many aspects of software product development. Students must be cognizant of these impacts and also be willing to work within their financial constraints (i.e., when ideas are viable or valuable but not financially feasible). Since our student-sponsored team chose to work without the benefit of university resources in order to retain full rights to their software product, they were financially responsible for several startup costs. For instance, the establishment of an LLC in Nebraska requires a number of fees be paid, including filing fees, the publication of the founding in a public newspaper, and the rental of office space within the state. Students may also have to pay for services such as JIRA project management software and development resources such as cloud services or web hosting for their product. In our experience, students were able to start with free versions of these services while enrolled in the capstone course, as they did not yet need to release their product or account for a large pool of users yet, so their primary financial investment was related to the formation of an LLC.

Students on a student-sponsored team must also be willing to seek out potential customers to help identify product requirements, since they do not have a sponsor to provide project scope and project requirements. Finding potential customers and getting good input from potential uses who do not have a vested interest in the product can be especially challenging and time consuming, especially when the project timeline is bound by the constraints of an academic calendar and students have other academic requirements and demands on their time.

One aspect of a traditional capstone experience that may appear to be lacking in a student-sponsored project is the connection with industry. Although the lack of an industry sponsor removes one of the typical connections students build with industry in a capstone course, using industry mentors to guide the students through the product discovery phase can provide an important industry connection that would otherwise be lost if students work with the university's business school or other university organization to provide this support. The university's legal clinic also provided a valuable new connection for students as they enter their careers in technology. Capstone faculty also benefited from these relationships, as they can be leveraged to help support future student-sponsored teams.

The lack of connection with industry is also a curricular issue. We did not allow students in their first capstone year to participate in the self-sponsored project because the IEEE Computer Society's and ACM's *Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering* states that a project should have a "customer" other than the supervising faculty member[21] (and, by extension, other than themselves). We could best guarantee that experience by requiring students' projects in their first capstone year to have an external sponsor.

Finally, intellectual property is a critical component for a student-sponsored project. Because the students on our student-sponsored team wanted to commercialize their project, we consulted and followed our university's policies to ensure students would retain full rights to their software products. Our university's policy is that if the students made substantial use of university resources on their project then the university would own the work.[16] As a result, the students chose to forego much of the support that we normally provide to capstone projects, and we confirmed with the university's general counsel beforehand, that the remaining support we provided would not be construed as "substantial use" of university resources.

5 LESSONS LEARNED

In this section, we provide the lessons learned from both the student's perspective and the faculty's perspective. Despite the faculty's inexperience with entrepreneurship projects and the activities involved with forming a software startup, the overall capstone experience for both the students and the faculty was positive and we plan to continue to offer this experience to future second-year capstone students.

5.1 Student Observations

A traditional capstone experience provides students with valuable experience learning new technologies and practicing software development in teams. Students on a student-sponsored project continue to grow in these areas, but they also learn about new product development and about the work necessary to create a new business entity. In previous coursework and capstone experiences, students' assignments focused on software development on projects with requirements and objectives specified by their instructors or project sponsors. The greater level of autonomy this team had over the deliverables and development of their software product, and the ability to explore technologies, conduct market research, and make contacts with local industry were novel and valuable experiences.

Lesson #1: *Practicing teamwork and collaborative development in a self-organized team prepares students to work in industry or a startup post-graduation.* Previously, whether in course assignments or other capstone projects, students were typically assigned to their teams (and possibly even their roles). When this student-sponsored project began, students had an opportunity to self-organize, which enabled them to discover their leadership and technical skills. Each member of the student team was able to find a role in which they could learn and grow, and make unique and valuable contributions to their project. As a student-sponsored team, they also had to work together on tasks typically handled by project sponsors, such as establishing development guidelines and project priorities. Because the team had a common educational and technical background, they were able to reach a point of stability and productivity faster than if they had not had a shared background. Overall, students shared that participating in this student-sponsored capstone project helped prepare them for work in software development, marketing, and other areas of business after graduation, ultimately narrowing the gap between their academic course work and their careers post-graduation.

Takeaways:

- Students see an increased benefit from academic experiences that closely parallel real work in industry.
- With the assumption that they have sufficient prior experience and technical knowledge, students benefit when given greater autonomy.

Lesson #2: *Balancing technical development and business tasks is challenging for students.* In a traditional capstone project, sponsors typically manage the business side of the project, while the capstone team uses information provided by the sponsor to complete the technical tasks associated with developing the product. For a student-sponsored project, the business tasks are now in the scope of work completed by the students, in addition to the work of developing the product. This means that the conflicts and interdependencies between technical and business tasks now have to be managed by the students, just like any startup organization. This led to a novel experience for the students on the team, who noticed a feedback loop during the project in which business decisions influenced the technical direction of the product, and technical opportunities (and limitations) influenced business decisions. To help balance the demands of business and technical work, and also to accommodate students' preferences to focus on business or technical tasks, the students elected to divide the team into business and technical sub-teams. The business team focused on the business, marketing, and market research involved in the creation of our LLC and product, and other team members focused more on the software development tasks. Students who chose to be part of the business-focused team also worked on the technical development, but to a lesser degree, which provided first-hand experience helping the team attempt to strike a balance between the business considerations and development priorities. While it was both novel and, at times, overwhelming, working with these competing demands was an engaging and valuable experience of how competing priorities can influence the evolution of a product. Additionally, students realized that their assumption when starting the project was incorrect: they thought that they would be able to complete their project, with its added scope, while retaining all the same benefits and experience of a traditional capstone project. But as the time for the project was the same, adding scope in the form of research and business tasks necessarily took time from development tasks. This is not to say that the time was not fruitful or well-spent, but the students were not able to develop as much of the project as they would have given the full academic year without the business tasks.

Takeaways:

- When developing a software product, business opportunities and limitations inform development priorities.
- Similarly, technical progress and limitations can inform business priorities.
- Giving students a view into the business and technical side of their decisions and development allows them to see how these two sides influence the other and give them a more complete picture of the product design and development process.

Lesson #3: *The benefits students receive from their capstone experience correlate directly with what the students are willing to invest.* Students can derive considerable benefits from working on a traditional project in a familiar domain or using familiar technologies. At a minimum, these projects allow students to learn how to work in teams, how to manage projects, and how to communicate in a professional environment. However, once students have completed the project and have learned these basic skills, completing another project under similar conditions limits the ways in which students can grow. The students on the student-sponsored team chose to leverage the safety of an academic environment to push themselves outside their comfort zone of working for a project sponsor. They chose to start a new software company and product. Because students were given a lot of freedom in how they accomplished these goals, they also had a lot of responsibility to push themselves to learn what they needed to know. They also had to manage their time and project priorities, given the expanded project scope but same timeframe as a traditional capstone project. And, they had to be willing to push themselves to take risks when it would have been easier to choose options they were familiar with. In the end, this was the greatest benefit of participating in a student-sponsored team.

Takeaways:

- When students are working with a greater degree of autonomy, they have greater control over the benefits they can gain from their work.

- Self-disciplined and self-motivated students stand to benefit greatly from being involved in student-sponsored projects, as they have to be ready to put in a full effort to realize all the benefits of such a project.

5.2 Faculty Observations

Our Software Engineering program has many unique qualities that allowed us to experiment with a student-sponsored capstone project. These include the fact that our program follows a cohort model in which students complete many of their major classes together. Our students also have many opportunities to practice software engineering in teams by completing team capstone projects in their core courses, completing two years of a year-long capstone course, and through the completion of at least one internship for credit. However, we believe our experiences with a student-sponsored project can be used to inform other computing capstone courses that would like to offer a similar opportunity to their students. In the remainder of this section, we highlight the key lessons the faculty learned from our first experience with a student-sponsored project.

Lesson #1: *Team cooperation and cohesion are paramount in a self-sponsored team.* Team dynamics are a crucial component of any team; however, in a student-sponsored team, the additional pressure resulting from the increased project scope, the uncertainties of what product will be developed, and the lack of an external sponsor to guide the team means the team's success is even more heavily dependent on their ability to quickly form into a cohesive unit, and then to operate in a cooperative manner. On the one hand, our students had the advantages of studying and working together for three years so they were able to quickly organize their team, assign roles, and begin working on product discovery. They also self-selected their team and thus began the project with a sense of trust in each other's skills and knowledge. On the other hand, the students' shared academic history may have meant they were less creative in their approach to solving problems when compared with a group of students with diverse academic backgrounds. Moreover, the rapid team formation may have nudged students into team roles they successfully performed on previous projects, rather than allowing the students to explore a new role in this team.

Takeaways:

- Specialized teamwork instruction, practice, and tool support may be necessary to help students with the challenges of working in a team in an uncertain and unstructured environment.
- Team members' prior experiences working together can help the team form more quickly but may also be a limiting factor at both the individual and the team level.

Lesson #2: *If possible, have the students complete project discovery prior to the beginning of the fall term.* The timeline and deliverables for our first student-sponsored team were determined on-the-fly as the academic year progressed. This was partly due to our lack of experience with mentoring a team through the early stages of project discovery, and partly due to scheduling issues with our industry partner. Ideally, students will have completed their project selection by the beginning of the term, or at the latest, during the time when other capstone teams are going through the formation process and getting to know their project sponsors. Planning for early discovery optimistically allows students to start development early, but it also affords more time if the team needs to start over or continue product discovery.

Takeaways:

- Identify and build your support team prior to the start of the project.
- Identify project deliverables and rubrics for assessment.
- Identify project resources (e.g., sample business and marketing plans or templates).
- Create a flexible timeline that specifies deliverables and dates to guide the project and to give the support team an idea of when their support and services may be needed.

Lesson #3: *Be willing to let the students fail along the way (and reassure students it is OK to fail).* In an academic setting, the objective is typically to provide students with the resources necessary to help them succeed. In a student-sponsored project, it is important for students to work with less structure and fewer instructions so they can learn the importance of initiative, and the challenges of solving problems that are not well-defined, or that cannot be easily solved. Failures along the way can provide important lessons for students. To encourage students to take calculated risks that could lead to failure, reassure students this is an important component of the learning objectives.

Takeaways:

- Create a framework for monitoring the teams's decisions and progress to ensure students are making adequate progress, even if it is not on a direct path to their goals.
- Have a plan in place if product discovery fails. Our plan was to ask the students to begin product discovery again with a new set of ideas.

Lesson #4: *Finally, be willing to model how to step out of your comfort zone for your students.* We had the option to decline the students' request because we did not have practical experience taking a software product from concept to development, nor did we have time to plan a new process for a student-sponsored project. We instead chose to use this opportunity to step outside our comfort zone to fill our knowledge gaps. This opportunity also taught us just how much our students are capable of accomplishing in an academic year.

Takeaways:

- Let students know this is an experiment and you do not have all the answers–this is an important message for students to hear.
- Be patient and flexible. Things will likely not fall into place the first time, or even the second time. Based on our experience, it will likely take several student-sponsored projects before we have a process, timeline and set of expected deliverables we are comfortable with, and before we feel truly comfortable mentoring a student-sponsored project.
- Solicit volunteers to help when you lack expertise. We found industry volunteers who were happy to help the students (and faculty) with their project, treating them as any other client who seeks help with a software product idea.

6 CONCLUSIONS & FUTURE WORK

Experiential learning opportunities are a critical component of an undergraduate curriculum. They provide students with practical experience working in their field and making connections to the real-world. In this work, we describe how a student-sponsored computing capstone project benefits students by exposing them to the intersection of software development and business development. Although the student-sponsored team did not deploy a product at the end of the academic year, they did complete many of the business tasks that are outside the scope of a traditional computing capstone course, in addition to the software development tasks. Students on the team found the overall project experience beneficial, especially when compared with working on a traditional capstone project for a second year. In particular, students felt they had gained a more realistic view of how software is developed in industry, and they developed a greater appreciation for how software development activities fit in the broader context of business practices, especially in a startup or small business environment. Students also found the autonomy of a self-sponsored capstone project to be beneficial by providing them with opportunities to research and choose technologies themselves, and to make decisions regarding which features their product would support.

If a traditional capstone project is viewed as a way to introduce students to software development practices, processes, and technologies used in practice, then a student-sponsored project is a way to introduce students to the business practices of goal setting, investigating market viability, and performing other tasks typically handled by a capstone project sponsor – while also providing students with additional opportunities to practice their Software Engineering, teamwork, and communication skills. Students gain experience working in a software startup with less financial risk and more time to dedicate to product development by pursuing their software product ideas within the context of an academic course. Although many startup businesses dedicate a large amount of time to securing investments from outside sources, the student-sponsored project chose not pursue outside investors as part of their capstone project. This was due, in part, to the fact that outside investors would introduce additional stakeholders in the project and the students wanted autonomy in their project decisions. The students also did not want to further expand the project scope to include the time consuming process of investor acquisition. Now that this group of students has completed the capstone course, they plan to continue development of their product, and they plan to release the product to the market in the near future.

Student team members are not the only beneficiaries of the student-sponsored project. Their future employers–even if they are not startup companies–will also benefit from the knowledge and experience the students bring with them into their careers. The startup community in general can also benefit from working with Software Engineering students who have already experienced the challenges of starting a software company. Finally, the faculty who mentored the students have benefited from the experience through their exposure to entrepreneurial activities. This knowledge can be carried forward to the benefit of future capstone students.

Although our experience is based on a single team and a single academic year, student feedback indicates a student-sponsored capstone project is a valuable experience that can expose students to a broad range of product development activities – such as business, marketing, and branding – that would not have been included in a traditional capstone project. Although the students did not take the product to market during the academic year, the work they were able to complete left the students feeling more prepared for their post-graduation careers – whether or not they choose to participate in a software startup.

Scalability of a capstone course is always an important topic when considering a new types of projects. While two faculty members had oversight of the student-sponsored project, we believe that once the relevant partnerships are well-established and the lesson material is stable, the per-faculty workload will be no greater than that for a traditional capstone team (i.e., one faculty member for 5–6 capstone teams), and that the model will scale to accommodate additional entrepreneurial teams until factors other than faculty workload become the limiting factors.

In the future, we plan to improve the learning experience for students on a student-sponsored team, by 1) continuing to develop our collaborations with local industry to mentor students through the product discovery phase of the project, 2) placing a limit on the time for product discovery to provide more time for students to work on business and software development tasks, 3) creating a set of resources specifically for startup-style capstone projects, 4) creating a parallel framework of processes, activities and deliverables for student-sponsored projects that provide similar support to our traditional capstone projects, and by 5) developing an evaluation system for student-sponsored projects to inform future improvements.

ACKNOWLEDGMENT

The authors would like to thank the team at Don't Panic Labs for their support of the student-sponsored capstone project.

REFERENCES

[1] Gursimran Walia Alex Radermacher and Dean Knudson. 2014. Investigating the Skill Gap between Graduating Students and Industry Expectations. _. [Online].
[2] Rafael Chanin, Afonso Sales, Leandro Pompermaier, and Rafael Prikladnicki. 2018. Challenge Based Startup Learning: A Framework to Teach Software Startup. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) *(ITiCSE 2018)*. Association for Computing Machinery, New York, NY, USA, 6 pages. https://doi.org/10.1145/3197091.3197122
[3] Stefan C. Christov and Mark E. Hoffman. 2019. Experiential Learning of Software Project Management and Software Development via Course Collaboration. _. [Online].
[4] Nitish M. Devadiga. 2017. Software Engineering Education: Converging with the Startup Industry. In *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*. 192–196. https://doi.org/10.1109/CSEET.2017.38
[5] John. Dewey. 1938. *Experience and Education.*
[6] John G. Flowers. 2008. Improving the Capstone Project Experience: a Case in Software Engineering. https://dl.acm.org/doi/10.1145/1593105.1593167. [Online].
[7] Scott Freeman, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences* 111, 23 (2014), 8410–8415.
[8] Nicole Herbert. 2018. Reflections on 17 Years of ICT Capstone Project Coordination: Effective Strategies for Managing Clients, Teams and AssessmentReflections on 17 Years of ICT Capstone Project Coordination: Effective Strategies for Managing Clients, Teams and Assessment. _. [Online].
[9] Marko Ikonen and Jaakko Kurhila. 2009. Discovering High-impact Success Factors in Capstone Software Projects. https://dl.acm.org/doi/10.1145/1631728.1631803. [Online].

[10] Kevin Lee Feifei Chen Andrew Cain Mohamed Abdelrazek Jean-Guy Schneider, Peter W. Eklund. 2020. Adopting Industry Agile Practices in Large-scale Capstone Education. _. [Online].

[11] Youry Khmelevsky. 2019. Ten Years of Capstone Projects at Okanagan College: A Retrospective Analysis. https://dl.acm.org/doi/10.1145/2910925.2910949. [Online].

[12] Don't Panic Labs. [n.d.]. Don't Panic Labs. http://dontpaniclabs.com. [Online; accessed January 2022].

[13] Viljan Mahnic. 2012. A Capstone Course on Agile Software Development Using Scrum. *IEEE Transactions on Education* 55, 1 (2012), 99–106.

[14] Colin Maly, Suzette Person, and Leen-Kiat Soh. 2021. SE-First: A New Approach to Software Engineering Education. In *2021 IEEE Frontiers in Education Conference (FIE)*. 1–9. https://doi.org/10.1109/FIE49875.2021.9637325

[15] University of Colorado Boulder. 2020. 2020 Senior Design Projects. http://www.colorado.edu/ecee/undergraduate-program/senior-design/2020-senior-design-projects#team-initiated/. [Online; accessed March 2021].

[16] University of Nebraska Board of Regents. 2001. Intellectual Property Policy of the University of Nebraska. https://digitalcommons.unl.edu/ir_information/14/. [Online].

[17] Maria Paasivaara, Jari Vanhanen, and Casper Lassenius. 2019. Collaborating with Industrial Customers in a Capstone Project Course: The Customers' Perspective. https://dl.acm.org/doi/10.1109/ICSE-SEET.2019.00010. [Online].

[18] Maria Paasivaara, Dragoş Vodă, Ville T. Heikkilä, Jari Vanhanen, and Casper Lassenius. 2018. How Does Participating in a Capstone Project with Industrial Customers Affect Student Attitudes?. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training* (Gothenburg, Sweden) *(ICSE-SEET '18)*. Association for Computing Machinery, New York, NY, USA, 49–57.

[19] Jacob R. Miller Sandra Gorka and Brandon J. Howe. 2007. Developing Realistic Capstone Projects in Conjunction with Industry. _. [Online].

[20] Andrew Scott, William Kreahling, Mark Holliday, and Scott Barlowe. 2017. A Holistic Capstone Experience: Beyond Technical Ability.

[21] IEEE Computer Society and Association for Computing Machinery Joint Task Force on Computing Curricula. 2015. Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. https://www.acm.org/binaries/content/assets/education/se2014.pdf. [Online].

[22] D.A. Umphress, T.D. Hendrix, and J.H. Cross. 2002. Software process in the classroom: the Capstone project experience. *IEEE Software* 19, 5 (2002), 78–81.

[23] Scott Wurdinger and Pete Allison. 2017. Faculty Perceptions and Use of Experiential Learning In Higher Education. *Journal of e-Learning and Knowledge Society* 13, 1 (January 2017).