Involving Multiple Levels of Students in a Software Capstone Project - A Case Study

Robert Hatch
The University of Virginia's College at Wise
One College Ave.
Wise, VA 24293
rjh7g@mcs.uvawise.edu

Nicholas Setliff
The University of Virginia's College at Wise
One College Ave.
Wise, VA 24293
nrs9m@uvawise.edu

ABSTRACT
This paper documents a joint effort by Software Engineering capstone students and Software Engineering survey course students to create an augmented reality application. The goal for the capstone course was that the soon-to-be graduates could exercise knowledge learned, as well as have a taste of management, and have some help in completing their project. The goal for the survey course was to have students working together in teams on a big software project to give insight into what they would encounter in industry. The paper’s focus is on the issues and lessons learned by everyone involved in the project through qualitative feedback.

Keywords
Software Engineering, capstone, multiple class collaboration, communication skills

1. INTRODUCTION
The Software Engineering Capstone course at the University of Virginia’s College at Wise represents the application of the sequence of Software Engineering courses. The course is a two-semester sequence; during that time, students can engage in the entire lifecycle of software development - from inception to deployment. A recent capstone project was determined by the students with minimal suggestions from the supervising faculty: an augmented reality (AR) application that would help incoming freshman, as well as visitors, to navigate their way around campus using their smartphones. The supervising faculty felt that the low number of team members (two students enrolled for the sequence) would not be consistent with what the students might encounter upon entering the workforce.

The supervising faculty for capstone included the survey Software Engineering course (CSC/SWE 2300) students in Spring 2018 to help the seniors complete their project - the first time that an entire class has been involved in such a deep, supporting role. Students in this course typically work through a software engineering lifecycle model to develop an assigned project over the course of a semester. Struggles occurred along the way, such as the seniors’ ability to communicate effectively with their subordinates, as well as adjusting to the loss of a member of the capstone course midway through the spring semester. This particular iteration of the capstone course also added complexity because students were required to manage other team members. The supervising faculty hoped that the nature and scope of this project would give the capstone students a more realistic view of the actual work environment in the industry upon graduation, if this field is their intended route.

The paper is organized as follows: related research; determining the scope and nature of the project; the lifecycle process of developing the AR app; reactions and suggestions from all students working on the project; and ideas for future work and improvements for future project iterations.

2. RELATED WORK
Knowing the theory is not sufficient to solve real problems that a professional would encounter on a daily basis [8]. Additionally, a challenge for the course is finding an appropriate project for the term - “appropriate” meaning that projects need to be complex enough to require a team effort and application of skills learned in class. Krutz [6] suggests that students are interested in a real-world contribution. Buffardi, et al., [3] also note that “software engineering courses commonly incorporate semester-long team projects to approach emulating the real-world software development process and environment.” Beck [1] notes that these types of projects expose students to the “messy reality” of the types of situations and problems they can expect to encounter when they become employed. Students also need a clearly defined project in order to succeed, but students have issues with decisions made prior to their arrival because they want to understand why choices were made [4].

Buffardi, et al.,[3] observe that real software development projects “involve teamwork to design and produce more sophisticated products.” [1] suggests that students should be divided into teams; each team works on some portion of the system, and teams must work together to complete the project. Matthies, et al., [7] also note that agile may be a better route because the participants “adapt to the circumstances and find solutions that work in the given context instead of blindly adhering to a prescribed process.”
Another consideration is that “while students may enjoy the freedom of defining their own software products, students have recognized that such ‘toy projects’ lack the pressure of delivering well-designed and tested software to real users” [3]. Other major challenges for group work involve how to grade each individual, as well as ensuring work is divided fairly within the group [2], [1]. Francois, et al., [5] suggest the use of a “class-wide project,” which engages the efforts of all the students in the design and implementation, which is “out of the reach” of any individual student working independently in the timeframe of the course. Students will often express a significant amount of satisfaction in a project, which in turn contributes to their satisfaction with a course [6]. It helps if a project looks like something a student might encounter in the real world as a professional. Such a project also has a major benefit related to the process from the instructor’s point of view in that it makes the students critically eye the tasks in the project and then analyze them to make decisions about each task [1].

3. GATHERING REQUIREMENTS

The format of the two-semester sequence involved meetings on an as-needed basis to discuss questions and issues that the students had concerning the project. This time also provided feedback on students' progress for a certain week. The first meeting of the semester focused on students as they were beginning to think about prospective projects. The only restriction placed on the class by the supervising faculty was that the project had to provide some facet of service or community service. At the following meeting, the idea for the augmented reality (AR) campus map application was selected for further development.

The rationale for the AR campus map application was to provide visitors with a way to navigate the campus. The major functionality would be to provide directions from one building on campus to another building or location. Other functionality included a virtual bulletin board and a step counter. The virtual bulletin board would be utilized at a designated space in each building for a department, and each bulletin board might have announcements and student opportunities. The steps feature would allow the user to select the number of steps desired to walk, and the application would provide a route around campus that would satisfy the user’s step goal. The major functionality of going from one building to the next would best satisfy the needs of visitors on campus, as well as freshmen and transfer students. The virtual bulletin board would be utilized at the server and channel, where students could discuss issues and get help where needed. The project leads and the supervising faculty realized that communication was a problem. Students were not participating in the scheduled meetings due to conflicts or because students forgot about meeting times. During this first part of the semester, communication issues also occurred between teams and capstone students. One of the capstone students was out of town and unable to resolve issues. The other capstone student did the best he could to resolve the issues. However, not long after a quarter of the way through the semester, the project lost this second project lead. Communication between this person and the rest of those invested in the project ceased, even after attempts to reach out to this individual.

Project meetings were initially handled through a discord server and channel, where students could discuss issues and get help where needed. The project leads and the supervising faculty realized that communication was a problem. Students were not participating in the scheduled meetings due to conflicts or because students forgot about meeting times. During this first part of the semester, communication issues also occurred between teams and capstone students. One of the capstone students was out of town and unable to resolve issues. The other capstone student did the best he could to resolve the issues. However, not long after a quarter of the way through the semester, the project lost this second project lead. Communication between this person and the rest of those invested in the project ceased, even after attempts to reach out to this individual.

After receiving initial feedback from students which predominantly focused on communication, project meetings were held for the last 15-20 minutes of a class lecture. Students could resolve questions they had face-to-face with the re-
maining project lead. When the remaining project lead determined that students fell behind schedule, project work days were added during the second half of the semester. Everyone from the class would meet in a computer lab and have access to the project lead during these project days. Morale (measured through assigned grades in student feedback) and progress picked up in the second half of the semester. The major lesson learned was that while communicating from anywhere can be convenient, face-to-face communication is necessary to make sure the project continues to progress.

The basic functionality of the project was completed at the end of the semester, enabling someone to navigate from one spot on campus to another location. The suggested features of a virtual bulletin board and routes to achieve step goals were postponed. The bulletin board was also delayed to consider the time commitments needed from other departments to maintain the virtual space. While the prototype works, there is room for enhancements, including the aforementioned postponed features, as well as improving battery performance.

5. STUDENT FEEDBACK

The following subsections describe feedback received from students at different times in Spring 2018. Subsection 5.1 addresses responses to the same question asked on the first two exams - grading each student in his group and justification of grades assigned. Subsections 5.2-5.4 address questions concerning hindsight, suggestions for faculty, and likes/dislikes/did the student feel he benefitted from the project. These questions were asked on the final exam for the survey software engineering course. Subsection 5.5 shows feedback received from the one capstone student who persisted throughout the semester.

5.1 Feedback Via Exam Questions

When the first two exams for the course were administered, students were required to answer a question about their immediate team, as well as the state of the project. The students needed to assign grades to each of their own group members and provide supporting reasons; specific comments were not shared with any students who received negative feedback. Feedback was shared with students in the form of comments written on their graded exams. These students were encouraged to continue participation because their grades depended upon the outcome of the project’s successful launch. The grades assigned by students were used as a guide as to how to grade an individual student’s work on the project at the end of the semester.

One of the major themes which popped up on the first exam was communication. According to one student, “So far (our) group (has) lacked communication. We have not discussed any ideas and I am not sure if anyone else has started making paper prototypes.” Students were not afraid to grade each other down, due to a lack of communication. Another student’s response: “I would give (this) member 30 points (out of 100) as he appears to know what is going on in the project thus far but has not actually said a word to me as of yet.” Another student in another group also commented (out of 100) as he appears to know what is going on in the project thus far but has not actually said a word to me as of yet.” Another student in another group also commented on communication issues, as well as a concern about organization: “To be honest, my team’s project has been stale. The one time we met together as a group, it was just me (a project lead) and (another team member) in the discord channel. Our other member wasn’t there because of miscommunication. We have the will/desire to work, but no organizational skills.”

With the first exam, some students were also marked higher by their teammates for their willingness to take on the leadership role for the project. Other students working on the project were graded higher based on their understanding of concepts that were necessary for the project. One other reason for receiving high marks from teammates came from regular attendance at meetings.

The students in the survey class also graded their project leads; at this point, the project had two capstone students as project leaders. On the first exam, students were not receptive to one of the team leads because he was unavailable for a period of time while he was traveling. The second project lead received higher marks because this student was the active leader on the project. Issues with this project lead included his occasional “condescending tone,” as well as “miscommunication leading to confusion.” Those negatives were outweighed by students who thought that he “work(ed) so closely and intelligently with our group to make sure we understand what it is we need to do and how we may be able to do it. He also seems to be willing to assist as needed with the work our group does reliably.”

A student’s average score earned from his teammates in the first phase was a 72.63 (out of 100). The average for the project lead whom the students felt was available to answer questions was 80.33; the second team lead earned a score of 74.42, with three students abstaining from assigning a grade, due to lack of involvement.

After receiving feedback from students, the supervising faculty attempted to get the project back on track. Part of the reasoning behind asking for feedback about the project leads was to gauge how well the two leads were performing from a leadership standpoint. With one student committed at that point, while the other was less certain, the instructor used the student feedback to address the issues which the survey class had in a meeting that took up part of a lecture day when the first exams were handed back. This meeting involved all of the students in the survey Software Engineering course, the two capstone students, and the instructor.

The communication issues continued with responses from the second exam. Students gave each other higher marks than on the first exam, but some noted that others did not speak until directly spoken to. Scheduling conflicts due to other priorities, such as work and personal life, also interfered with being present for meetings and reaching milestones. Some students were reported to have missed meetings, but they remained very active and constructive teammates as they completed project milestones.

For the project leads, the second capstone student simply disappeared. The other team lead stepped up his role, and the students took notice. One student responded that the first project lead was “effectively the polar opposite of (the second project lead) as he responds to questions in a timed manner and actively seems to be part of the project it-
self, attempting to help anyone who may be struggling." A second student noted, “He’s always available to talk to and pretty much always has an answer when asked a question.” A third response, also positive, was: “Since he jumped (back) into the project, (the first lead) has provided everybody with very good communication and documentation of team goals.” Another student also kept in mind how well things changed after the first exam - “strong improvement, specific tasks laid out, and helps with development often when able. Meetings went from never to weekly (just prior to [instructor] involvement).”

Students earned an average score of 86.43 (out of 100), with comments providing proof of improvements in morale and work ethic. The first team lead, the one who eventually stopped communicating and working, received a score of 8; the second project lead rebounded, and earned a score of 96.9 from his subordinates.

With some improvement and still some work to do in other areas, the supervising faculty revised what was done in the survey class. Weekly, in-class meetings were allotted after a lecture. The students communicated with the remaining project lead to clear up or address issues they were having about specific tasks assigned. Additionally, project work days were utilized in lieu of a traditional lecture, which allowed students class time to work and ask any further questions of the remaining capstone student. While a priority, grade-wise, the project was given a higher priority in the classroom as well, which drove the project’s continuing progression to a working prototype by the end of the semester.

5.2 Hindsight
Student answers were varied. One student responded, “I would (have) put more emphasis on reaching deadlines on time. I also would try to help...change communication to become more effective. The software process model wasn’t set in stone and we did have to go back to change things but I would have put emphasis on structure.” Another response was, “I would make sure communication was established earlier on...This would have saved more time for proper documentation and testing due to us avoiding a major problem we ran into that made us have to redesign the implementation of (a necessary) algorithm.” A technical concern from another student was, “I would have asked...to use a server-client model and do all processing off device. I’d also use an object-oriented design...(and actually have used) instead of a massive combined spaghetti pile.”

The third student responded that he would have begun work on the project sooner. “The last 2 weeks felt rushed compared to the first 3 weeks.” Another student was also concerned about structure: “I would have made sure that there was a more stable plan moving forward. Make a more effective schedule for tasks. Learned more about the project, Swift, and GitHub in order to be a more effective team member.” Two other students had similar feelings about GitHub and that some instruction might have helped (either by the project lead or the instructor).

Other students were concerned about process. As a student remarked, “I would address the issue of communication and implement the task deadline for every two weeks earlier...Also fix some of our current project version’s flaws like battery usage and node placement.” Another response stated, “I would start with the explanation of agile. I think we had potential to ‘hit the ground running’ on the project if we knew more about agile. We lacked communication and timely meetings.” The last response was also concerned with communication: “I would have communicated better and pushed through the tasks at a more driven pace. Further I might have recommended to (the project lead) that he make the initial tasks smaller and more digestible to get everyone more motivated.”

In sum, tools, process, and communication appeared to have been the major issues with this project.

5.3 Feedback on Faculty
The students’ responses indicated they felt that the instructor’s approach was adequate without any major changes. One student’s suggestion: “I would change the leadership roles by getting to know each individual and basing the position off their experience, knowledge, and understanding of the task at hand. I would not have the professor do anything differently.” A second response said that “the project days allowed for my group to catch up on our task.” To build off that response, another student replied, “Toward the end of the project, we started having weekly meetings with (the project lead) during class time. It would have been nice to have had that earlier in the semester because it helped us all to discuss the project as a whole rather than separate teams.” One student concurred with the project days - “maybe have the project work days earlier on in the semester to get people involved and it makes sure there’s a time where everyone can work together and focus.”

Another student responded that placing more emphasis on how important documentation is to the “final phases of the project and assign someone to make sure documentation stays relevant and up to date.” A student suggested enforcing some kind of punishment for teams not completing assigned tasks. “The lack of a real incentive killed most motivation unless personally prodded.” Another responder was concerned about grades and the focus of the course: “I would suggest to make the primary focus of the class the project. Another thing I’d suggest is for us to see (how) the project is (affecting) our grade; I know I worked hard, but if (the project lead) or my teammates don’t, then I need to tell you.”

A last concern was both with the project, and something that could have been included - a code review: “I wish I had tried a bit more at the beginning as in the end our project isn’t (as) fully complete as I would like...Otherwise the project just needs better documentation which you could have possibly asked for throughout development (code walkthrough).”

The students found that the project days they were given toward the end of the semester were invaluable as well as an increased amount of time to communicate with the project lead. The face-to-face communication seems to have been better received by all than having students communicate via discord server whenever they had the opportunity. The project days were added after midterm, based on student
concerns about completing the term project. Because our Introduction to Software Engineering course is a survey course, critical parts of the theory were selected, instead of continuing to follow the given textbook.

5.4 Likes, Dislikes, Did Students Benefit?
The last question on the final asked students about their likes and dislikes on the project, as well as if they benefited from time spent on the project, and if other students taking the course in the future would also benefit. Nine of ten students responded positively, in that they felt like they benefited from working on the project and could also see it as a benefit for capstone students needing help in completing a project. Some of the other benefits the students saw included real-life communication issues, experiencing chaos, working on a large-scale project for (probably) a student’s first time, and getting a taste of what they might see once gainfully employed in the profession. The one student who didn’t feel they benefited had issues involving the Swift programming language. This was the only positive for this student, in the end - picking up knowledge of an additional programming language.

One student opined, “I liked the challenge it presented and the experience I gained through working on it. I didn’t like that it was only targeted towards Apple products only because it made it more difficult to build and test with limited resources...I think other students would benefit by testing and reviewing code.” A second student said that a positive was “being able to be a team leader and coordinate people based on their skill level,” while a negative was “people not understanding what (they) can/can’t help.” Another student replied, “Getting to work on an actual, applicable piece of software rather than learning about one that someone else done. As we began making progress with the application, the morale and initiative picked up quite a bit and students felt like their contributions were needed and had some fun on a collaborative project [unheard of].”

Another response: “I liked how unique and beneficial the project was to bringing together all of our knowledge from both previous and current courses. I disliked how, at first, there could be major communication issues delaying the project. This was by far the most effective learning experience for my major I had and I think it could be for others as well.” Another response: “I greatly enjoyed the project, it was challenging, yet rewarding...I got to see what’s in my future here, put something extremely cool on my resume, and had [some] fun on a collaborative project [unheard of].”

The next student response: “I enjoyed how it was an actual tangible thing that you can say you built and not just another program that’s been done for years. My least favorite part was definitely the Gitflow, it’s overly complex.” The next student said “What I liked most was that I learned how to work in a team and how important it is to communicate with each other. What I liked least was that we can’t put the app on the App Store. It’s like our efforts meant nothing. I’m not sure about other students but I feel like they feel (the) same way as me. This was a cool project.” Another student was mostly positive, saying, “(What) I liked most was learning how to work on a large team. I have only worked in pairs until this class. I least liked working with Apple products. I wanted more say in the platform because I have never used (A)pple products...I learned so much about software development.” The last response: “The project showed the level of effort necessary which I enjoyed as it helped prepare for real-life applications better. The chaos at the beginning with the project leadership as it caused a decent amount of confusion.”

5.5 Project Lead’s Thoughts
The project lead was asked to write a journal of thoughts over the course of the semester about working with the survey course students. His initial thoughts included that the “capstone project is shaping up to be the most memorable, stressful, and rewarding experience of my college years. Although at times it’s been overwhelming; watching the individual software engineering phases merge together has been a learning opportunity that I wasn’t initially expecting. I believe both myself and the SWE class have grown as software engineers and computer scientists as a direct result of this project.”

The project lead’s thoughts about the overall scope of the project: “In my opinion, this is the most real-world project I’ve been a part of. Although they usually result in a working application, the different aspects of software development are usually taught over separate courses and less emphasis is placed on anything outside of that course’s domain. The capstone project demands that equal attention is paid to each individual part or things will begin breaking down quickly. As a result, the benefits of a disciplined, systematic approach are more apparent.”

Even though everyone seemed to feel great about this project as a learning experience at the end, there were struggles: “As a learning experience, this project has been amazing. However, as a course that I’m ultimately graded on and responsible for, it’s been extremely stressful. I think any group projects developed in college will have to contend with not everyone doing their fair share, but this one has been exceptionally poor in that regard. For the first (two-thirds) of the project, I’ve felt like my only options are to either do something myself or accept that it probably won’t get done. As we began making progress with the application, the morale and initiative picked up quite a bit and students were eager to solve the next problem or implement the next feature.”

The project lead also noticed that the tools utilized for the project were also a struggle: “One of the biggest technical issues that held us back during development was familiarity and knowledge of working with Git. Ensuring we were developing on the proper branch, with the most recent code base, was something that didn’t seem to be understood. I attempted to write up a detailed explanation of Git, the GitFlow workflow, and how it all fits together, but it still didn’t lead to widespread understanding. Ultimately, a few individuals in the development team learned how to use Git, and the individual teams always use one of those accounts to initiate pull requests or commit any code. As I believe that Git is absolutely crucial to developing real-world applications with non-centralized team members, it would be great if more emphasis was placed on teaching students how to properly use source control throughout their tech-related college courses.” Swift was also cited as a major issue, from students in the survey courses. To learn a new programming language together with the topics covered in the survey course.
Suggestions for improvement included more guidance from the supervising faculty. The lead notes, “It would’ve been helpful to have someone in a senior leadership role saying, ‘I need the design documentation on that specific class by next Friday’ or ‘be prepared for a technical code review by the end of the month.’ It may not even be an issue with a capstone class consisting of more members (each able to focus on something different) but being by myself I think a little more guidance would’ve helped nudge me along in the right direction.’

The student does note that this project was a valuable learning experience: “All in all, this experience will stick with me throughout my professional career. Just working with a larger development group helped me realize what real-world software engineering must be like. I would like to thank (the supervising faculty) for trying something new with the capstone project, as there’s no way I could’ve done any of this without the help of the SWE class.”

6. FUTURE WORK

Supervising faculty hope to return to this project, either in the form of another capstone project or in some other software engineering-related course. The only functionality completed for the capstone was getting users from one point on campus to some other desired point. The suggestions of a digital bulletin board for each department, as well as achieving step goals through suggested routes could be added in a future capstone iteration.

The continuation of this project also introduces a concept that students are rarely exposed to, beyond theoretical discussion: software maintenance. Students do not realize that the “glory job,” so to speak, of creating a new product from scratch rarely happens; when students start work at their first job, typically they will maintain a piece of software. Other courses’ content could be exercised as well, such as testing, and the many different types of testing to ensure the product’s correctness.

When other students inherit this project’s code, much of what is taught to them early on in programming about readability would also be addressed. Students may not realize that having meaningful comments, if any comments, is crucial because someone else might pick up work on a piece of software after them. In turn, other items such as writing clear code and adopting meaningful names would be reinforced at this point. A capstone project of maintaining this software after them. In turn, other items such as writing clear code and adopting meaningful names would be reinforced at this point. A capstone project of maintaining this product could be beneficial to students because all of the prerequisite courses could still be hit, but perhaps with a different emphasis (for example, requirements specification for new functionality and testing strategies for exercising the new functionality, as well as any corrective maintenance). A process would need to be employed, and one better than the semi-agile, Scrum-route that was used for this particular project.

Additionally, supervising faculty will need to take a more proactive role, suggesting tasks to complete, such as some type of walkthrough or inspection, and maybe even acting as a de facto change review board. One of the other points supervising faculty should consider if they are utilizing and picking up this same project is an expansion into Android development. Android users also maintain a significant user base for mobile devices. Students might even see the challenges of making the same app look the same on two, totally different platforms.

7. CONCLUSIONS

This paper has documented a year-long project involving capstone students. The focus of this paper is to discuss the results of taking a senior project and providing students with their own shop, consisting of those who are enrolled in the survey Software Engineering course. The major caveat in the second semester was communication, and resolution was attempted through more face-to-face communication with the project leads. The students were able to successfully deliver a working prototype, and most of the survey course students felt like this project benefited them.

8. REFERENCES


