The 401 Capstone Course at USC: A Case Study

Suhrid Karthik and Jeffrey Miller
University of Southern California

CSCI 401 is an undergraduate computer science capstone course offered at the University of Southern California. In this paper, we describe the design and organization of the course. We specifically discuss the various aspects of the course structure and address the choices that influenced their design. We document our experiences and learnings from designing and running an undergraduate capstone course at scale.

Keywords: case study, course design, course structure, experience report

Corresponding Author: Suhrid Karthik, skarthik@usc.edu

Introduction

CSCI 401: "The Design and Construction of Large Software Systems" is offered at USC[1] as a capstone course for undergraduate students in Computer Science. The course has been offered for several years, and the structure has undergone a steady evolution over time. The elements of the course structure were designed to satisfy learning outcomes for the students, delivery outcomes for their stakeholders, and help in evaluating student effort. In this paper, we aim to discuss these design choices in detail. The course regularly experiences large enrollment between 100-200 student each semester. We therefore also discuss our experiences in scaling the class based on rising enrollments. The remainder of the paper is structured as follows. First, we introduce course objectives and the broad course structure. In the subsequent sections, we discuss design of various course structure elements. In terms of evaluation, we discuss both student and stakeholder perspectives of the course. We summarize our learnings in the conclusions section.

Course Structure

Course Objectives

The course objective is for students to apply the skills that they have acquired throughout their undergraduate studies on a practical software project. This project is conducted with an outside stakeholder to develop real-world software solutions to large scale problems. The projects are planned to be a semester long (~ 4 months) activity. The students work in teams of 2-8 members under the supervision of the stakeholder for the duration of the project.

Stakeholders are invited to submit project proposals. We describe the process of project proposal, approval, and student assignment in the Project Assignment section. Once the teams are assigned to a project, they begin working on the project in consultation with the project's stakeholder(s). Throughout the duration of the project, students are required to complete various deliverables which are listed in Table 1. We describe the deliverables in detail in the Deliverables section. The deliverables are designed from a threefold perspective:

1. Maintaining project progress: To ensure that student teams make regular and consistent progress on the project.
2. Meeting stakeholder expectations: To ensure that stakeholders receive regular project artifacts and updates.
3. Enabling student evaluation: To assist in grading students' participation towards the success of their projects.

Deliverable	Grade
Project Schedule	4%
Intermediate Deliverables (Design/Functionality)	30%
Complete Documentation	6%
Peer Reviews	6%
Weekly Status Reports	30%
Final Presentation	20%
Peer Review of Presentation	4%

Table 1: Project Deliverables

Project Assignment

Project Proposal

We solicit project proposals from stakeholders who have expressed a desire to supervise such projects. We ask stakeholders to suggest suitable projects to which students can make meaningful contributions within the timespan of a semester. We ask stakeholders to submit the following information:

1. A description of the project with necessary background information.

2. A range for the number of student team members that are suitable for the project. (e.g. 3-5)
3. A list of the specific technologies that is required for the project (e.g. Android programming, Python). If the stakeholder has no preference, she can leave this choice up to the team.
4. In addition to the technologies, the stakeholder may wish to specify recommended background knowledge (e.g. security domain knowledge, front-end development).

We have found that this information helps us to match student interests with project requirements. Once the list of projects has been collected, the course instructor approves the projects. For the approval process, we consider the feasibility of projects with respect to timelines, expected skills, and responsiveness of the proposed stakeholder.

Project Assignment

Before the start of the course, we circulate the list of approved projects to the students enrolled in the course. This is done a few days prior to the first day of class so that students have a chance to read through the project descriptions and potentially form teams.

The actual project assignment process is carried out on the first day of class. We go through the list of projects linearly and ask student teams to indicate if they are interested. In case only a single team is interested, the project is assigned to them. In case multiple teams or more than the maximum number of students for the project are interested, we hold off on the particular project and go through the remainder of the list. At the end of the first round of assignment, we ask those teams that are interested in the same project to check if they would like to pick an alternative unassigned project. In case of conflicts, we ask the students about their backgrounds and interests and assign the project to the team that subjectively is a better fit. After the first round, we also assign students who are not yet in a team to groups.

Scaling the process

The above processes have worked relatively well so far without any major hitches. However, as student enrollment in the course grows (currently nearing 200 students in a single semester) along with the number of invited project proposals (over 50 projects per semester), we have been faced with issues of scale. For instance, it became challenging to track all the project proposals with the increased volume of email correspondence with the project stakeholders. We anticipate that stakeholders also find it hard to manage the project lifecycle through email and would appreciate accessing proposals through a single portal. The project assignment process has also become increasingly difficult. Manual facilitation to resolve all conflicts concerning the number of teams and projects was time consuming. For example, there can be multiple "ties" that need to be broken, and the unassigned students then need to be included in teams. We want to streamline this process and ensure that all students get one of their top choice projects as much as is possible.

To address these issues, we have developed an online project management portal ("Capstone Project Platform") to handle these processes efficiently. The portal itself was developed by different student teams over the course of multiple semesters with the professor of the class as the stakeholder. The portal has the following features:

- Allow stakeholders to submit and revise project proposals.
- Allow course administrators to accept and reject project proposals.
- Allow students to view and rank the list of projects.
- Allow course administrators to run a project matching algorithm to automatically assign students to projects based on their preferences.

We anticipate deploying this portal for stakeholders and projects in the near future.

Deliverables

Deliverables constitute 40% of the grade and there are three categories of deliverables that we expect the students to produce and submit. We describe them in detail below. The grading methodology for these project deliverables is based ***almost completely*** on stakeholder approval. This process involves the students submitting the deliverable to the stakeholder, and the stakeholder subsequently emailing the course instructor to let him know if the deliverable was approved or not. This design is due to several reasons. One of the main objectives of the capstone course is for students to work with real-world stakeholders which involves managing their expectations. Secondly, a stakeholder is naturally the right person to judge the accuracy and quality of a deliverable since every project implemented in the course is different. And last, it allows us to effectively scale the process to many student teams since we rely on individual project stakeholders' judgements as opposed to any centralized approval mechanism or authority.

Project Schedule

After the project is assigned, we ask that student teams schedule a meeting with the stakeholder. This meeting is an opportunity for the teams to learn more about the project and its requirements directly from the stakeholder. We give the students almost two weeks before they are asked to submit the Project Schedule.

This is a document that broadly lists the agreed-upon project deliverables along with their milestone dates. The purpose of this document is that it serves as a loose contract between the students and the stakeholder regarding the project requirements. Additionally, we find that this document motivates students at the outset to consider the end goals of the project. This helps them to think critically about project planning and consider necessary system integration tasks. Apart from this initial meeting, the students are required to meet weekly with the stakeholder over the course of the project.

Project Deliverables

As the project progresses, we require that students submit deliverables once every two weeks. The definition of what constitutes a deliverable is flexible. For example, this can be project code, documentation, or a UI prototype. Students typically discuss an upcoming deliverable with the stakeholder during weekly meetings. We find a bi-weekly delivery schedule is effective in ensuring timely project progress. This time window allows for meaningful chunks of work while at the same time accommodating for resolving unforeseen difficulties and other demands of student time.

Final Project Documentation

The project documentation is the final deliverable expected of the students. This is intended to serve as the project handover document. We expect it to act as both a user and an administrator manual. We encourage students to include end-user guides as well as information on deploying and maintaining the application. We also expect details such as building and navigating the code-base. This information is valuable to the stakeholder based on the current project but is also useful for them to evolve the project after the involvement of student teams end (or to continue the project in a subsequent semester).

Status Reports

Apart from project deliverables, we require students to provide weekly individual status updates. These are separated into two formats. The first format is an online form that is required before the start of every week of class. In this form, students are asked to provide a description of their contributions to their project in the preceding week and their planned activity for the upcoming week. We ask students to enter a brief description and the number of hours for every task in the past week and for future tasks in the upcoming week. The second format of the status updates involves in-person meetings with the course instructor or the teaching assistant every week. During this face-to-face meeting, the course instructor/TA meets with every project team and typically asks the group members to briefly talk about their project progress. Additionally, individual students are asked to verbally discuss their tasks accomplished for last week and upcoming tasks. This in-person status meeting typically lasts for ten minutes. Apart from gathering status updates, this meeting provides an opportunity for us to gauge the progress of the project and offer constructive feedback. This may be suggestions on functionality, technology choices, or resolving teamwork issues.

Requiring students to provide regular status updates certainly helps the project to stay on track. However, we find that a dual format status update is more effective compared to a single update. Firstly, the online status update helps students to not only reflect on their past contribution but also compels them to think critically about their activities for the next week. This helps them to not only be better prepared for the in-person meeting but also helps in their group planning. Secondly, asking students for individual updates when their entire group is present also acts as an effective commitment device.

Remote Status Updates

Each individual project team is required to come into class only once a week in order to provide their in-person status update. Since this is often a short ten minute meeting, an increasing number of students requested if they would be able to attend the status meeting online. We therefore decided to trial this. After being satisfied with the logistics during the trial period with a few teams, we decided to extend it to all project teams. We chose to use the Skype[2] platform for online meetings. We ask the teams to create Skype groups with all their team members. We also specified a specific group name which includes the project number and a time so that we can sort the groups by name which helps in placing calls. We found online meetings to be an effective substitute for in-person meetings. This allowed the students to better plan their day without having to necessarily come to class only for a short meeting. We also find that conducting such meetings online helps with scaling the process as the number of teams increase. We did reserve the right to ask a team to attend in-person if necessary.

Peer Reviews

Another measure of student accountability that we use is peer reviews. We require students to submit three peer reviews throughout the semester. For each team member, students are required to state what they have done well and potential areas for improvement. We find peer reviews to be a valuable tool to ensure cohesive teamwork. Peer review responses provide valuable clues on team dynamics. In cases where team members are critical of a certain student, we reach out directly to the student to check. Based on our experiences, we recommend 2-3 peer reviews to be conducted over the course duration.

Final Presentations

At the end of the semester, students are required to prepare a presentation to showcase the result of their efforts. The presentation is graded on an 11-point rubric. These include parameters such as use of visual aids, architecture employed, and skills and tools used. An important part of the presentation is a live demonstration of the software. We invite stakeholders to attend the presentations as well. The presentation is assigned 20% of the grade. We find that the relative importance of the presentation and the requirement of a demonstration with the prospect of the stakeholder in attendance all combine to motivate the teams to deliver a polished product.

We also require students to grade presentations of other teams. While we do not use their grade, we find that such a peer review encourages student participation in the process. The presentation scores from the professor and students are also used in ABET[3] assessments.

Evaluation

To measure the effectiveness of the course and its design, we asked the students and stakeholders to respond to a brief survey after the most recent iteration of the course.

Student Responses

We had a total of 21 student responses. In order to measure student learning outcomes, we asked how much the class helped them in advancing different skills. Students were asked to respond on a 5-point scale (Poor, Fair, Satisfactory, Very Good, Excellent). In terms of teamwork, **90%** of students rated the advancement of their skills as Very Good or Excellent. In terms of technical skills (e.g. technologies, tools, etc.), **2/3** of the responses were Very Good or Excellent and the remaining 1/3 were Satisfactory. We anticipate that this is due to the natural distribution in the level of technicality of projects. The real-world (e.g. requirement scoping/stakeholder negotiation) skills had **2/3** of the students responding as Very Good or Excellent, whereas the remaining 1/3 were either Fair or Satisfactory. We also asked students to rate elements of the course structure on a five point scale (Very Ineffective to Very Effective). Results are in Figure 1.

From these results we can see that students found weekly stakeholder meetings/biweekly deliverables to be quite effective. Weekly in-person status meetings again were rated to be more effective than online status updates. This is reasonable to understand as the online updates are more of a mechanism to enforce student commitment. A sizable minority of the students found peer reviews to be ineffective. This could be due to the reason completing the peer review did not have a direct bearing on grading for most students except for the poorly performing, and therefore the process could be seen as busywork.

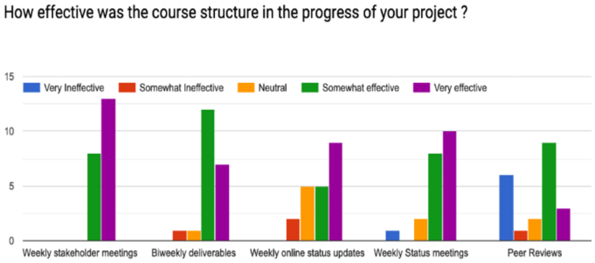

Figure 1

Stakeholder responses

We had a total of 13 stakeholder responses where we had a different set of questions. When asked to rate outcomes of the projects, **all** the responses indicated that the level to which functional requirements were met to be Very Good or Excellent. **77%** of the respondents felt the same way for non-functional aspects (e.g. usability). Regarding the course structure, **85%** and **77%** of stakeholders felt biweekly deliverables and weekly meetings to be "Very Effective" respectively. For approving deliverables, we had a few (4) responses rating it as Somewhat Effective. We anticipate improvement here when we reduce the need for email communication (e.g. through the Capstone Project Platform). Additionally, more stakeholders rated the documentation to be Very Effective compared to the presentation (**77% vs 54%)**. This is understandable, since the presentation is geared more towards students, but we will use this feedback to tailor the presentation structure.

Conclusions

We hope our case study of this capstone course has been informative. We highlight our key findings:

- Distributing the project delivery into regular graded deliverables is effective from both the student and the stakeholder perspectives. Mandating stakeholder approval for grading scales the evaluation process efficiently while ensuring stakeholder satisfaction.
- Using multiple measures to enforce student status reporting results in consistent student involvement and project progress.
- Developing customized tooling (e.g. our Capstone Project Platform) is increasingly important to scale to a large number of student teams.

References

[1] USC, University of Southern California, https://www.usc.edu
[2] Skype, https://www.skype.com
[3] ABET, https://www.abet.org/