

CIS 467 Project Requirements

Introduction

This document describes the Computer Science Project course. It is meant to provide you with everything you need to know about the project in order to accomplish it successfully. If, after reading this, you still have questions, please don't hesitate to ask.

Objectives

The CIS project course is a “capstone” course and is designed to allow you, the student, an opportunity to demonstrate the skills you have learned in prior courses. It is a “capstone” in the sense that it sits above, and spans your prior coursework, focusing all of them together into a single course. In addition to utilizing all that you’ve learned to-date in the CS program, the course is also designed to approximate the typical real world setting many of you will find yourselves in as professional software developers in the near future. You will work with a development team on a non-trivial software development project over the course of the semester. Your capstone experience derives its fidelity to the “real world” from the fact that you will be creating software for a customer – a carefully selected capstone project sponsor.

The course is also designed to give you the opportunity to add a significant “real” project to your career portfolio. Many students use the project in CIS 467 as a highlight in their programming portfolio that they show to prospective employers.

Finally, this course will provide you the opportunity to increase your team-work skills. We expect teams of approximately 4 or less to work together to design and implement a non-trivial software system.

Course Goals

The course has three overall goals: product, process, and professionalism. Please keep all of these in mind when designing and developing your project. All three goals affect the grading of the various deliverables.

Product emphasizes the fact that the project you are assigned is a real product that is being requested to serve a real need. In building this product you will bring together the skills and knowledge acquired in many/most of the courses that you've had previously. The end deliverable will be useful, and bug-free.

Process measures the software development processes that you use in order to reach your project goals. Process addresses issues such as: What methodologies are used to ensure successful project completion --- agile, extreme, waterfall, etc.? How good is your time-estimation? How did you adjust your deliverables schedule to compensate for advances or setbacks in project development? How well did your team members work as a team? Did they support each other to produce a successful project?

Professionalism looks at things like the quality of the deliverables. Are they complete, well-written, and error-free? How good is the final presentation? How well did the team communicate during the milestone meetings? Did the team members exhibit professional behavior during the semester?

Evaluation/Deliverables

You must read the paper by D. Gotterbarn, K. Miller, and S. Rogerson, Software Engineering Code of Ethics is Approved, *Communications of the ACM*, Vol. 42, No. 10 (October 1999), 102 – 107. It is available from the ACM digital library. It explains SECEPP (the Software Engineering Code of Ethics and Professional Practice). It outlines professional responsibility towards the profession, the public, teams, etc. Your knowledge of the SECEPP will be demonstrated in the Final Report.

The Project & Project Customer

Students will be divided into teams. Each team will be assigned a project that has been defined and requested by a Capstone Sponsor. Capstone Sponsors consist of external corporations and internal GVSU organizations that will be assuming the role of the customer. Just like the real world, the quality and completeness of the submitted project proposals varies by customer. In some cases, the customer has a fairly well defined idea in mind, and in other situations the customer has a more open-ended idea that requires more fleshing out. In either situation, your goal is to engage with your assigned customer, understand (or help them understand) their needs, and in the end deliver a functional software product that delights them. Your grade in part depends on how well you satisfy your customer.

Team Assignment / Size

Teams will be defined by the instructor and consist of approximately 3-4 students each.

Project Portal

Each team must establish an online project portal by end of the day January 11. Your team's project portal is where all of the documentation and code your team produces will be stored and organized. This semester you will be using github (<http://github.com>) for your project portal. If your project sponsor is ok with an open code repo on github then you can use the default "free" plan. If your project sponsor would like the code repo to be private one that can be accomplished for ~ \$22/month. You should have a conversation with your customer about this before you setup your project on github so there are no surprises later.

If your github project ends up being private, you must add me to the project as a collaborator so I can assess your work over the course of the semester. In any case, you do want to get your project portal established quickly (this week yet if possible) and get up the github learning curve quickly, if you haven't already.

Email the instructor a link to your project as soon as it is established. If your project ends up being private, than add me as a collaborator (my userid is jengelsma).

Failure to have your portal created and initial journal entries (see next step) posted by the above deadline will impact your grade.

Journal

Professionals such as plumbers, electricians, lawyers, and accountants keep a log and charge accordingly. We software engineers must do the same. As an exercise for your future endeavors after graduation, you must develop the habit of tracking your time since it will be the basis for planning and cost estimation in your future projects.

Everyone must maintain a journal showing: "**date, start time, end time, total time spent, task description**" for all times spent on this project. Remember to include times that we take for granted such as responding to emails and meeting times. Finally, keep track of the total accumulated time spent on the project.

Each of you will establish your own personal journal page Always keep your journal up to date and post a link to everyone's journal on the team website. You may need this information for the team evaluation reports. You should keep your work journals on your project's wiki on github where all of your teammates as well as instructor can view them and understand what you are working on. You must journal consistently the entire semester – an entry for every day that you work on the project. **Failure to journal regularly will impact your final grade.**

Git Repository

When working on a team-based software project, modern software configuration management (SCM) tools are absolutely essential. Your team will be using git this semester to manage your

code base. Git will help you keep track of all revisions of your code, and allow you to effectively manage the complexities of a larger shared code base being updated by multiple people (possibly geographically dispersed) simultaneously.

Some of you have already have had some experience using git or a similar distributed SCM. Some of you may not have. Whatever your level of experience, you are expected to become proficient with git as quickly as possible. We will not be teaching you how to use this tool, though we can help you along the way as needed. The optional textbook for the course (see syllabus – freely available online via GVSU library's access to the Safari Collection) is a good place to start, and there are plenty of other online resources as well, including this free e-book entitled "Version Control by Example", <http://www.ericcsink.com/vcbe/>. The good news is that git is fairly simple to work with, and it's a good tool to have in your toolbox moving forward.

In addition to helping you manage the complexities of team-based software development, your code repo is also important for another reason in this course. While the journal and written peer evaluation and final report represent a somewhat qualitative log of your activity on the project, your code repo provides a more quantitative picture of what is happening on the team day by day. Your instructor will be lurking behind the scenes in your project repos and assessing your progress as both a team and as individuals over the course of the semester.

Prospectus

Your first deliverable requirement is to produce a prospectus for your project. The prospectus should fully describe who is on your team, the technical strengths of each team member, and a brief but complete description of the system you have been assigned to build. When complete, post your prospectus on your project portal. (I will assume you do this for all documents from now on).

Completely describe the project you have been assigned. In order to accomplish this, you will need to establish contact with your customer as soon as possible, and have an initial meeting in which you can acquire a more solid understanding of what you have been asked to build. You should carefully review the preliminary project description you have been given on the first day of class, and start laying whatever groundwork you can accomplish before your initial meeting with the customer. e.g. start reading up on any dev tools/platforms they might reference, google other domain specific terms they might be using, etc.

Your prospectus should give the instructor a very clear picture of the application you are going to produce. Don't worry too much about how you're going to do it. Provide a list of all intended features. Your prospectus should describe the vapor-ware that you will create. The prospectus is not a contract. It is a road-map that outlines what you will try to create. It is not necessary for you to implement every feature.

You should also describe areas of **technical growth** -- technical areas in which team members are not necessarily well versed in at the outset of the project that must be mastered in order to

complete the project. In the course of the semester you will demonstrate your expert technical knowledge, and your ability to learn new things on your own. These technical growth areas should be ones that you will spend a great deal of time researching new techniques, implementing algorithms, etc. Later on, you will be evaluated on your progress in these technical growth areas (see section on grading).

Design Document

The next requirement is a design document. You should produce a document that describes *how* you are going to implement your application. You should describe things such as...

- What language(s) will be used?
- What frameworks/libraries/API's will be used?
- How will your code repository be organized?
- Will you be using a database to store information? If so, what database system? Sketch out a possible schema.
- What is the basic organization of the system (client/server). Include figures showing your product's software architecture.
- How will the user interface work?
- The design document should discuss and provide rationale for what you are taking the approach you are taking. This should include some discussion of alternate approaches that you've considered, but decided not to pursue.
- What protocol(s) will be used between the system components? Are they standard protocols? If not, you should document them in your design document.
- What kind of testing will you do to ensure the software you deliver does what it is intended to do?
- Where will you be doing the development? What environment(s) will you use?
- What is the division of labor -- who is going to do what.
- What design methodologies will you use? Agile, waterfall, hybrid, etc.

Use figures and technical drawings liberally to describe the organization of your system. If your software system has a non-trivial user interface, you should provide wireframes or storyboards. There are a lot of free tools available that you can utilize for this. Provide a Gantt chart that shows when you will have the various parts of your application finished. Estimate as best you can. You will be graded on how well you meet your own schedule.

The design document involves a significant amount of work; so do not wait until the last minute to put it together. A good design document should clearly convey how you are going to implement your software product. It demonstrates that you've got a solid grasp of what it is you are implementing and that you've put some serious effort into thinking about how you will approach your task for the rest of the semester. While you will be turning the design document in to your instructor as a deliverable, you must also share a copy of it with your customer and ask them for feedback.

Customer Meetings

You will want to meet regularly with your assigned customer. It is suggested you meet no less than once every two weeks. Weekly meetings would be ideal if possible. These meetings should be arranged in a way that is convenient to all involved. They could be held on campus, at the sponsor's site (if external to GVSU), or via conference call (Google Hangouts, Skype or freeconference.com). It is suggested that your initial meeting should be conducted face to face. I will provide an email introduction for each team, at that point it is up to the team to organize itself, and make sure these meetings take place. Here are some suggested guidelines on meeting with your customer.

- Your customer will be a very busy professional person, so make sure you are well prepared for your meetings with a written agenda, and show up on time.
- Assign one team member to take careful notes during the meeting, and clearly identify any action items, who it is assigned to etc. These notes should be reviewed and agreed to by all after the meeting (once again, a wiki is a great way to do collaborative documentation.)
- In order to make sure customer meetings happen regularly, schedule them at a regular and consistent time. For example, agree to meet every other Friday morning at 10am with your customer, no matter what.
- Don't annoy your customer with frequent detailed email on technical issues that you could easily resolve in a few Google searches.
- Consider assigning a team member to act as the team's customer liaison for any interactions between your regular meetings. The liaison for example, might collect questions from various team members and compiled them into a single email, so the customer doesn't feel constantly inundated. At the same time, if the customer makes himself/herself readily available and encourages frequent and detailed interaction, don't be afraid to take advantage of that.
- A large part of your success will be dependent on how well you engage and satisfy your customer. If you show a high level of professionalism in the course of the project, and make significant and consistent progress from the get go, your customer will reciprocate and invest more and more of their own time and effort into the initiative.

Milestone Presentations

Over the course of the semester, each team will make three public milestone presentations during the regularly schedule class meeting time. Before the scheduled presentation, each team will produce a progress report. The milestone report should include a copy of the project's Gantt chart, and a summary of what tasks/deliverables were to be completed to-date, and also a summary of what has actually been achieved to-date. Teams should describe any setbacks that caused schedule slippage, and a summary of plans to address this moving forward. Finally, each team will also summarize what they hope to accomplish by the next milestone presentation. Any revisions to the schedule and/or deliverables should be clearly documented and explained. Subsequent milestone reports must have previous milestone reports attached

as appendices for reference. Copies of this report should be posted to your project portals by noon on the Monday of the week the presentation is scheduled.

Each team will be given 15 minutes to present their progress. (10 minutes to present, 5 minutes for Q&A). Project teams should prepare professional quality visual presentation materials summarizing their milestone reports using software of their choice (PowerPoint, Keynote, Google Presentation, Prezi, etc.) A copy of this presentation should also be posted to your project portal in advance of your presentation.

Milestone Presentation Attendance: All students enrolled in the class must attend all scheduled milestone meetings. Each student will be required to provide individual feedback for each team. This information will be collected and provided back to the team in anonymous form. Failure to attend and participate at milestone meetings will impact your individual grades. In addition to student's enrolled in the class, other CIS faculty will also be invited to attend and provide feedback.

The batting order for presentations during milestone presentation weeks will be announced the week before.

Post Milestone Presentation: The instructor will provide you a written summary of his assessment of the team. Included will be the aggregate (anonymous) feedback from the class. Teams are encouraged to meet with the instructor to go over this feedback in person if possible. The instructor will be available for consultation during the schedule class period, office hours, or by appointment.

Milestone Report Content

For each milestone your team is required to produce a milestone report that outlines the progress made to date. The milestone report is due on your website *before* each scheduled presentation. The written progress report should follow this skeleton outline...

Milestone X Report (Prepared by: *Names of team members*)

Intended Progress: Describe the features/components you planned to complete by this milestone (per Gantt chart).

Progress To Date: Describe what the team have been doing towards completing the tasks involved in realizing the current milestone objectives. Report on any significant problems with meeting goals, and identify the team's plan to address.

Projected Progress: What will you do in the next project phase in terms of code development, testing, integration and debugging. How will the team adjust its workload in order to compensate for current rates of progress in the chosen development areas, and how will work towards the next milestone be organized.

Conclusion: Brief overview of the status of the group and project overall.

Team Evaluation Reports

The team evaluation report is your chance to rate your teammates. You are each to complete an evaluation of your teammates individually on the designated days (see timeline at the end of this document). I will hold these reports in confidence. The format of these evaluations is illustrated below.

Team Evaluation Report

To be completed individually (confidentially) by each team member and uploaded to Blackboard. Assess each team member's performance since the last team evaluation report.

Name: *You* **From:** *Date* **To:** *Date*

Member 1: *FirstName LastName*

	Strongly Agree	Agree	Disagree	Strongly Disagree
1. Made good technical contributions				
2. Has been responsive and dependable				
3. Showed initiative				
4. I am glad he/she is in my team				

Enter below, as a numbered list, brief and precise comments to justify your assessment.

A Word document is available on BlackBoard with 3 copies of the above table – one table for each of your team members. Please fill out this document and submit via Blackboard before or on the evaluation due date. There is also space to provide additional comments if you feel inclined.

Final Report

The final report is precisely what it says. It should be a document that you will be proud to send to potential employers. It should be readable and fully understandable by a person who has the level of knowledge of a CS senior. The report must contain at least the following sections:

Abstract: A 200-300 word statement highlighting the functionality and major components of your system.

Introduction: A 3-4 page description expanding on what you said in the abstract. Remember that a lot of readers will not continue reading if either your abstract or your introduction do not entice them to spend more time reading the rest of your report.

Body: This section will contain the bulk of your report. The body of the report should have subsections detailing each major component of your system. Use drawings and illustrations where appropriate to improve the readability of your report. You will likely be utilizing some of the material from your original design document in this final report, though you may need to revise it to reflect the actual implementation. Your report body should also discuss non-functional requirements such as performance, security, etc. If a deployment involving actual users was achieved, you will want to include details on that (e.g. analytics gathered, user response).

SECEPP Issues: Study the SECEPP paper (see above for the reference). Identify all items under principle 3 (Product) that are relevant to your work. For each such item, state it and then give a brief and clear comment on how your team addressed that item during the development of the project.

Conclusions: State here an objective assessment of lessons learned, major difficulties encountered, things you wish you had done differently, and any opportunities you see for further work to enhance and/or extend your system.

In the abstract, introduction, and body be sure to highlight your two areas of technical growth.

A bound hard copy of your final report must be delivered. The inside of the back cover must have a DVD/CD sleeve containing a README file, an electronic copy of your report, and all the code that is part of your project.

Final Demo Video

Each project team must create a 2-5 minute final demo video of your product. The exact information and organization of this demo is left up to each team. The intent of the video is to give anybody unfamiliar with your product a quick overview of what it does. It should be well organized and compelling. A copy of your video (in decent resolution and in a standard format) must be included on the DVD mentioned in the previous section.

Here are a couple of example demo videos both of which take a different approach. You can use these as models for your own product video.

You can find a number of final video's (of varying quality) from previous semesters out on the GVSU CIS YouTube Channel: <http://www.youtube.com/gvsucis>.

Final Presentation and Live Demo

At the end of the course, you are required to produce and deliver a final 25-30 minute presentation. The format of your presentation should generally follow the format of the written final report.

Following the presentation, you should be prepared to demonstrate your application. Your peers will also have a chance to rate your application. During the final presentation, the other members of the class will assess your project.

Your customers and their colleagues from the sponsor organization will be invited to attend the final presentations.

Each team will also be demoing their final project at the Padnos College Project Day event.

Grading

Your final grade is broken down into several components.

Prospectus: 5%

Design Document / Design Critiques: 20%

Milestones: 25% (M1 5%, M2 10%, M3 10%)

Final Presentation / Demo: 20%

Final Report / Demo Video: 20%

Journal: 10%

Your individual grade may be higher or lower than the team grade, based on the instructor's individual assessment of your performance.

If your individual performance in any area is unacceptable, then you will fail this course, regardless of how well the team does. That means if you happen to join a really great team and you do nothing, you will fail.

Weekly Timeline

1. (Jan 8/11) Teams and Project Assignments announced / Proj Portals Established.
2. (Jan 17) Prospectus and Team Evaluation #1 due
3. -
4. (Jan 29 / Jan 31) Design Crit Sessions (overall design)

5. (Feb 5) Design Document and Team Evaluation #2 due
6. -
7. (Feb 19/ Feb 21) Milestone 1 Presentations and Team Evaluation #3 due
8. -
9. Spring Break
10. (Mar 12/14) Milestone 2 Presentations and Team Evaluation #4 due
11. (Mar 19 / 21) Design Crit Sessions (user interface)
12. -
13. (Apr 2/4) Milestone 3 Presentations and Team Evaluation #5 due
14. -
15. (Apr 18) PCEC Project Day Presentation
16. (Apr 25) Final Presentations (during final exam period) and Team Evaluation #6 due