CSE/IT326 Software Engineering – Spring 2019

Department of Computer Science & Engineering, New Mexico Tech
Mon/Wed/Fri 10:00 - 10:50AM (Cramer 203)
Lecturer: Dongwan Shin

Final Project Guideline

The purpose of the course project is to provide the students with the knowledge of software engineering methodology and the skills to apply it. The particular project is not the goal in itself; rather, it serves as a vehicle to apply your knowledge and to develop the skills.

Projects also introduce students to teamwork, which is unavoidable for large-scale software development. Teamwork has positive and negative aspects, and familiarizing yourself with both helps you get ready for your future workplace. There are many challenges to good teamwork: time coordination; building on each other's strengths and compensating for weaknesses; reconciling different working styles, preferences, and levels of motivation; etc. A key challenge is to make possible group efforts that discern contributions of each individual. This is a challenge in the course, where each student needs to be assigned a grade, as well as in your future workplace where your raise and promotion will depend on your perceived contributions.

This project consists of two iterations of a software product. The first iteration is exploratory and represents the first attempt at developing the proposed software product. The deliverables for the first iteration are **demo 1** and an **interim report** with requirement and design specification. After these, the instructor will provide feedback and the students should reconsider and possibly revise the project goals before moving on to the second iteration. Keep in mind that it is perfectly acceptable to modify your objective in the middle of the semester, once you learn more about the project and have better understanding of what you can accomplish within the semester timeframe. (Of course, you cannot switch to a completely different project in the middle of the semester.) This is why we have two iterations, so that in the second iteration you can perform the necessary adjustments, based on what you learned in the first iteration. The deliverables for the second iteration are **class presentation - demo 2** and a **final report** .

1. Teamwork and Project Management

Teamwork is required since teamwork is an integral part of large-scale software development. Larger teams should in principle be able to develop more sophisticated products. However, larger teams are also more difficult to coordinate and manage. It is more difficult to split the work in larger teams and ensure that every team member plays a meaningful and important role.

Students will be working in team (typically 3-5 members in a team) for their project. Teams will be formed by **January 25th**. **All team members must take part in all project activities and none of the activities should be done exclusively by one student**. Each team member or a pair must be responsible for all aspects of development required for the features they are in charge of, although it is acceptable to seek help from other team members who are more knowledgeable about particular technical work (e.g., database integration, user interface design, etc.).

Project management deals with organizational matters that are needed for effective teamwork. Team members should elect one "team leader" who will take a lead in project coordination activities. Project management includes the following:

- Organizing group meetings and keeping track of deadlines
- Managing shared resources (such as database, software repository, etc.)

- Integrating individual contributions in a coherent manner, and resolving ambiguities and conflicting information, including
  - collating different sections into a project report representing the entire team
  - proofreading the collated report to ensure consistent layout, font styles, section numbering, and language style
  - integrating different parts of the program into one software system
  - running integration tests and ensuring that the whole system works as intended
  - fixing software bugs that result from incompatibilities between software modules contributed by different team members
- Anything else that affects everyone in the team

Team meetings should be structured to involve:

- Report on the progress since the last meeting
- Discussion of tasks that need to be done - come up with specific action items
- A period of time to assign responsibilities, draft the parts each person is responsible for, and ask questions to be sure you all are on the correct path

Tools to help organize the teamwork and project management:

- Doodle (doodle.com) - a tool for time management and meeting scheduling
- GroupMe (groupme.com) - a group messaging service that lets you stay in touch with groups of people via mobile phones.
- Asana (asana.com) - a tool for managing projects
- Versionone (versionone.com) - a tool for agile-based project management
- Mingle (https://www.thoughtworks.com/mingle/) - another tool agile-based project management
- GitHub (github.com) - a web-based version control system using Git

Saying that "nobody asked me to do this or that," or, "I did everything that I was asked to do" is an unacceptable excuse. Each team member should be proactive and not wait passively to be assigned responsibilities. Do not ask others what should be done; rather, take initiative and suggest what should be done to make your project successful. Take every opportunity to redistribute and/or rotate the responsibilities, make your personal suggestions be heard! Many times defining the problem and determining what needs to be done is more difficult than actually doing it. Hence, problem defining and task assignment must be contributed to by all team members, rather than by the team leader alone.

2. Individual Contributions Breakdown

Many students find team work difficult due to different personal interests and working habits. Therefore, each student should keep track of his or her contributions to the project. The exact breakdown of individual contributions must be provided for every project deliverable, so that individual grades can be fairly assigned.

The **breakdown of individual contributions** should be submitted:

- In case of written reports, as an appendix of the report, or
- In case of project demos, by email to the instructor immediately after the project deliverable. Either of these options is acceptable:
  - One team member emails a single contributions breakdown for their team (this is preferable, but please copy your colleagues on this email), or
  - Each student emails individually an itemized list of their own contributions only.

Each student should provide an itemized list of his or her own contributions to the components of the particular deliverable, such as:

- requirements specification (use cases and non-functional requirements),
- software design (whole system or list the specific modules),
- coding (whole system or list the specific modules),
- debugging (whole system or list the specific modules),
- report preparation (whole report or list the specific sections/diagrams),
- Other: any other relevant contribution.

If several students contributed to a particular component, **quantify, as a percentage**, each students contribution to this component. Also provide a short description of your own contribution.

Team members who are not listed in the joint breakdown or do not email their contribution list will be considered as not contributed to the deliverable in question and will be assigned zero credit for this deliverable.

A potential problem has been observed in the past years. These projects are sufficiently complex that it is impossible for a single student to do everything. Good teamwork requires division of labor and building on each others strengths. Some team members may contribute more to design and documentation (report writing); others may focus on coding and testing the software (implementation for the demo). Let us say you have an ideal case where all team members are contributing to their best and everyone is happy with each others contributions. You may feel that just because some team members worked mostly on report writing, it is unfair to say in the contributions breakdown that others barely contributed. So you put "equal contribution" although you know this is not true. A better way may be to report the report contributions accurately and then give more weight to those team members who worked on the implementation and demo when reporting the demo contributions. This is why it is always best to be honest and list exactly who did what (instead of just saying "equal contribution"), item by item. If each team member genuinely contributed, their contribution will be appropriately rewarded.

3. Project Proposal

Each team must submit a written proposal for their project. A template will be provided, and the submission deadline is **February 1st**. Proposals will be reviewed and must be approved by the instructor.

It does not really matter if the project was done before or it is being done by another team as long as you will not copy their designs and/or the code. Every problem can be solved in different ways. If you will come up with an original design and implement your own code, it is perfectly fine that you develop a product somebody else did earlier or is doing now.

4. Project Report

Two reports are required for your project: interim report and final report. Templates will be provided, and the submission deadlines are **March 8th** and **May 1st**, respectively.

Reports should be prepared using a word processor. Handwritten reports or reports which contain handwritten material (e.g., figures, tables, etc.) will not be accepted. It is easier for us if you turn in your documents electronically in PDF format. Students are highly encouraged to use a software tool to prepare the UML diagrams for the reports. Any tool that supports creating UML diagrams is acceptable.

5. Project Presentation and Demos

Each team will give two demo presentations, each lasting about 15 min. If you are using your own laptop/notebook to project the slides and run your software, please ensure that the video connector is compatible with the cables in the lab before the actual demo.

After the demo, each student should provide an itemized list of his or her contributions to components of the demo, such as:

1. software coding (based on, but not including, the designs specified in project reports) list the specific modules,
2. system integration,
3. testing and debugging (whole system or list the specific modules),
4. slide preparation and product brochure preparation,
5. demo presentation,
6. Other: any other relevant contribution.

If several students contributed to a particular component, quantify, as a percentage, each students contribution this component. If all members of the team feel that everyone contributed equally, you can just write "All team members contributed equally" instead of a detailed breakdown. It is essential that you email the specific contributions for each team member. Some people are more comfortable with public presentation and quicker to answer questions, so the instructor may get impression that these students know the best and contributed the most. Hence, they will receive more grade points for the demo.

6. Project Grading Policy

Given that this is a design project, it is impossible to define precise grading criteria. The grades will inevitably depend on what student teams have to offer - so, grading is "market-based." Those that offer the most-impressive product, receive the highest grade. Which means that, if you receive a grade you are not happy with, it is not necessarily that you did bad work; rather, it is that others did better. Of course, there may be several teams with an impressive product, so several teams may receive the maximum grade.The grading policy for individual deliverables will be posted in the description of that deliverable. The specific requirements to be met will be listed. The reports will be graded based upon the technical content and clarity of exposition. However, it is **not** enough to meet all the listed requirements to receive the maximum grade. For example, having a perfect report for a mediocre project will result in a low overall grade. Thus, the overall perception of the project (relative to others) is the key scaling factor for all other aspects of the grade.

Grading the project-related deliverables works as follows. Say there are three teams, the first team has three students, the second team has two students, and the third team has four students. Then

1. The project deliverable for each team is graded independently of each other, and the maximum possible score that a team member can earn depends on his or her reported contribution breakdown. The actual earned points depend on the judged quality of these contributions. Suppose that the team members earned points as follows: (Team 1 reported "equal contributions", but lost somewhere 10 points, so the remaining 90 points are split equally:)
   - Team 1: member M1,1=30, member M1,2=30, member M1,3=30
   - Team 2: member M2,1=21, member M2,2=50
   - Team 3: member M3,1=21, member M3,2=12, member M3,3=35, member M3,4=26
2. Next, all team deliverables are compared as a whole relative to one another, and a holistic project grade is assigned. Let us say that Team 1 and Team 3 appear to have done about the same size project, and Team 2 worked on a smaller project, so it receives 70%. The teams project grades are estimated as Team 1 = Team 3= 100%, and Team 2 = 70% (relative to the other two teams).

The above example scores are shown in again in the following table. *Project grade* column shows the relative overall quality of the deliverable, compared across all teams. The team members individual earned points are shown in the column *Earned points*. The next column shows the *Adjusted points* of all students, obtained by scaling their individual score with their project grade. The maximum score in this column is 35. Finally, the last column shows the *Normalized points* for all students. This method compares each students contribution within the team and well as across different teams.

Late reports will be levied a late penalty of 10% per day, up to 3 days late. After that, no credit will be given, unless more than 40% of team members (e.g., at least 3 out of 5) provide a written excuse

Team	Project grade	ID	Earned points	Adjusted points	Normalized points
1	100%	M1.1	30	30	(30/35*100) = **86**
		M1.2	30	30	(30/35*100) = **86**
		M1.3	30	30	(30/35*100) = **86**
2	70%	M2.1	21	(21*0.7) = 14.7	(14.7/35*100) = **42**
		M2.2	50	(50*0.7) = 35	**100**
3	100%	M3.1	21	21	(19/35*100) = **60**
		M3.2	12	12	(12/35*100) = **34**
		M3.3	35	35	**100**
		M3.4	26	26	(26/35*100) = **74**

from a physician. Since the deadlines are known well ahead, there will be no extensions for any of the deadlines.

The weight for each deliverable for the final project is as follows:

Deliverable	Weight
Project proposal	**10%**
Interim report	**20%**
Demo 1	**20%**
Final report	**25%**
Demo 2	**25%**

7. Important Dates

Plan your time wisely for your project.

- **Proposal Due:** 2/1/2019
- **Interim Report Due:** 3/8/2019
- **Project Presentation:** 4/29 & 5/1/2019
- **Final Report Due:** 5/1/2019