

CptS 583 Software Quality

Course Project Specification

Overview

The course project is not only an integral part of coursework which accounts for 60% of the final grade, but more importantly it provides an essential means through which students practice and thus deepen the understanding and mastery of the principles and methodologies learned from the course lectures.

The project is a group assignment, supposed to be worked by a team of students. For this group project, each student team will first decide on a project to develop and then collaborate to define, assure, and assess the quality of the developed software system. Students will be given a list of sample project topics each with an initial requirements specification document. The quality modeling and assessment will cover both the quality of the resulting software product as well as the quality of the software process adopted (e.g., with respect to the analysis and design outputs).

Project Deliverables

The project will start with forming a team among the students taking this course in the same semester. Then the project progress will be monitored and evaluated through the following three key deliverables each corresponding to a major milestone. Each deliverable will be submitted as a single PDF (and, in particular for Deliverable II, code commits to the project repository).

Deliverable I

This deliverable marks the finish the tasks of setting up the team, nailing down a project topic, elaborating the initial requirements, defining the quality goals, and estimating the quality costs. Specifically, in this first project deliverable, each team will be asked to:

- List the **personnel and repository information** of the team, including WSU ID, name, and email; each team can have 3 to 4 members; a private Git repository should be created for the team project (e.g., on Github or Bitbucket).
- Specify the chosen project topic, and elaborate the **requirements** of the software based on the initial requirements for the chosen topic given by the instructor; a team can choose a topic out of the sample list, but must send the proposed topic

and requirements draft to the instructor at least a week before the due date of this deliverable.

- Clearly define the **quality goals** of the chosen project according to the requirements elaborated, outline the plan of quality assurance with respect to the defined goals, and describe the strategies for realizing the goals
- Clearly define the **quality metrics** and describe how the metrics will be used to measure the quality of the software and how they are related to the quality goals.
- Give the estimated **costs of the quality** targeted for the chosen project according to the economic assessment of the software quality development and maintenance processes.

This deliverable will be mainly be evaluated based on the level of depth of the quality assurance plan, the completeness and clarity of the defined goals, the choice of quality measures and their justification (with appropriate references), the strategies that will be used to assure that the quality goals are met as well as the justification for choosing those strategies. This deliverable will also be evaluated on the clarity of the scope definition of the quality assessment plan.

Deliverable II

This deliverable is the result of the second major milestone, which is the software product of the chosen project topic. Thus, for this second deliverable, each team will design and develop the software product and show that it meets the previously defined quality goals by following the strategies defined in the quality assessment plan. Specifically, for this second project deliverable, each team will be asked to:

- Conduct analysis and design according to the requirements specified (from the previous milestone).
- Implement **the software product** according to the design and develop necessary tests to validate whether the requirements are all met as expected; write software **documentation** clarifying the installation, configuration, and usage instructions; commit the code and tests along with the product documentation to the project repository.
- Verify the software product against the quality goals and metrics defined.
- Improve the quality using the initial verification and validation results
- Summarize the final verification and validation results as a **project report** (e.g., results of the quality metrics, which quality goals are satisfied and which are not, how quality has been improved using the initial results, etc.)

This deliverable will mainly be evaluated based on the completion of the software product, quality of the submitted code and tests, clarity of the project report, and the completeness of the report (e.g., objectively describing whether or not the quality goals are met). Further, the deliverable will be evaluated based on the quality improvements that the team performed and better quality (as per the metrics used) the improvements led to. The improvements include but are not limited to clarification and/or elaboration of the quality goals, the selection of alternative/additional quality measures, the selections of different techniques/tools to collect data, the selection of alternative/additional quality assurance strategies.

Deliverable III

This deliverable focuses on *independent* quality assessment --- that is, assessing the quality of software developed by others. For this final deliverable, each team will be given the software product along with its requirements and quality assurance plan, and then will be asked to:

- Conduct assessment of the product quality for the given software, using the project outcomes from a different team including the requirements and quality goals and metrics.
- Summarize the final verification and validation results as **an assessment report** (e.g., results of the quality metrics, which quality goals are satisfied and which are not, and other found issues); in addition, suggest in the report strategies for quality improvements with respect to the assessment results.

This deliverable will be mainly be evaluated based on the clarity of the assessment report, and the completeness of the report (e.g., objectively describing whether or not the quality goals are met). Further, the deliverable will be evaluated based on the quality improvements suggested. The improvements include but are not limited to clarification and/or elaboration of the quality goals, the selection of alternative/additional quality measures, the selections of different techniques/tools to collect data, the selection of alternative/additional quality assurance strategies.

Project Grading Breakdown

The deliverables will be graded using the evaluation criteria listed for each deliverable above. The following table gives the point breakdown using the percentage of final course grade/points.

Deliverables	Components	Percentage of the final grade
Deliverable 1		15%
	Quality goals and cost estimation	6%
	Quality metrics and assurance strategies	6%
	The delivered document itself	3%
Deliverable 2		25%
	Delivered software code	10%
	Software documentation	5%
	Project report itself	10%
Deliverable 3		20%
	Evaluation of the process and product quality	10%
	Suggestions for quality improvement	5%
	Assessment report itself	5%

Team Collaboration and Initial Grade

All team members are expected to contribute equally to each of the project deliverables and to all components of each project deliverable. **As part of each deliverable, the team will list the tasks assigned to each team member, the tasks completion percentages, and the tasks completion dates.** The instructor will refer to the commit history of artifacts (e.g., code and documentation) on the version control system (e.g., github) to check the contributing activities and time of each team member.

For each deliverable, every team member will initially receive an equal grade, which is given based on the submission from the team as a whole. This initial grade will be adjusted as described below.

Peer Evaluation and Grade Finalization

By the end of the project, each team member will fill a confidential peer evaluation questionnaire evaluating her/his own performance and the performance of the rest of the team. The questionnaire will include a form, where each student will give the overall contribution percentage by each team member in his/her view. For each form, these percentages must add up to 100%. The peer evaluation results will be used as guidance and will help the instructor to understand the team dynamics and the contribution of individual team members, and to eventually adjust each team member's project points.

The task assignment, the version control history, and the peer evaluation questionnaire will be used to adjust the project grade for individual team members. Based on the initial points of X of a student (i.e., which is the same for each member, equal to the group grade), the final project grade for the student will be $X \cdot P \cdot Q$, where P is the average of the completion percentages of all tasks assigned to the student by the team throughout all milestones/deliverables, and Q is the sum of the contribution percentages given by the student him/herself and all teammates of the student.

For example, suppose a student *Joy* worked on a team of four students including himself/herself, and his team eventually received a score of 45 points (i.e., $X=45$) for the team project. Also suppose that Joy was assigned three tasks, each for one of the three deliverables. For these tasks, by the time of delivery, *Joy* completed 100%, 70%, and 100%. Respectively. Next, suppose on the four questionnaires, the contribution percentages for *Joy* were 25%, 25%, 20%, and 20%. The final grade *Joy* will receive from the project will be $45 \cdot (100\% + 70\% + 100\%) / 3 \cdot (25\% + 25\% + 20\% + 20\%) = 45 \cdot 90\% \cdot 90\% = 36.45$.

Sample Project Topics

The objective is to develop a software product for *an actual client* who intends to use it in regular production. At the beginning of the course, you will form project teams with 4 to 6 members. During the semester, the project team will work together through the full development cycle, from understanding the requirements to delivering a functioning product. A client can be any person or organization except you yourself. Typical examples include a particular person, a university (e.g., WSU) department, a local company, or other external organization. A client from your local community (e.g., a nursing house, a pharmacy store, an auto shop, and a youth club, etc.) would be

particularly great to serve through this course project. Some potential projects are suggested below but you are encouraged to identify your own different topics (please contact the instructor and get the approval of your own topic in advance). In any case, the rule of thumb is that the software product will be practically useful, and there should be a firm intention by the client to use the software in production/practice.

In selecting a project, think broadly. Your project can be an application, system software, or even a toolkit. Software engineering covers everything from cell phones to supercomputers. The only conditions are that there must be a real client and real users.

For each suggested project, the initial requirements are provided alongside (unofficially). Part of your first deliverable will be to elaborate these requirements.

Credits: some project topics are suggested on the basis of the past [software engineering course projects for Cornell CS5150](#) and [Prof. Marsic's software engineering book](#).

Topic 1: Electronic Voting System

Project Description

You will be developing an electronic voting system that enables voters to vote securely from remote sites (rather than centralized polling sites). The system should be configurable allowing users to specify candidates, total votes available, date range for voting, and the winning number of votes reaching which the winning candidate can be determined without waiting until all votes are casted (think about the presidential election process).

You are expected to build either a desktop application or web-based implementation that is robust, user-friendly, and deployable. The system should include a user interface showing the voting results (e.g., the final votes each candidate received and the winning candidate, and possibly the vote statistics by geographic locations or political regions) for which visualizations can be employed to enhance the understanding of results.

Topic 2: Librarian Assistant

Project Description

You will be developing a librarian assistant that helps a librarian manage common library-management tasks, including user registration, book search, placing a hold on a book, checking out a book, and returning a book, etc. You may need to use a database of book information to support this application. Users can also browse the books in the library and look into the details about each book selected. The user interface could be a desktop application or a website portal.

You may use your creativity to determine which properties/attributes each book should have. You are encouraged to sit with a WSU librarian to understand details requirements of the application assuming that the librarian would end up with using this software for their routine tasks.

Topic 3: Mobile Type-II Diabetes Lifestyle Advisor

Project Description

You will be developing a mobile app (on iPhones or Android phones) that helps users suffering from diabetes risks with healthy eating. Type II Diabetes is fast becoming one of the most pervasive and costly diseases to treat in the United States. Emerging data shows that lifestyle changes in the early-diabetic stage can potentially reverse the condition or at least postpone dramatically the full onset of the disease. For example, better glucose control will have remarkable quality of life improvements for patients and remarkable cost savings possibilities for the medical care system. The app will enable patients to learn to control their blood glucose levels via a smartphone-assisted data logging process that tracks reactions to food (type, quantity, and combinations) and monitors the effects of regular walking, jogging, or running. Long term trend analysis will demonstrate to the patient the value of continuous attention to their disease and the mitigation of adverse side-effects.

Gait data along with body temperature will be sampled by sensors and comprise the exercise data-set. A detailed food journal must be made available to the patient users and will form the basis of dietary input on the hardware via an appropriate GUI. This food information, together with patient-user entered blood glucose values will be used for blood glucose analysis. Pre, post and 2-hour post eating blood glucose data as well as daily weight and blood pressure values will be entered by the user. After collecting this physiologic & user lifestyle data, the app will perform data processing, report generation, and user interface. It also performs both long-term and short-term data visualization and performs trend and event analysis on the entire set of stored data.

Topic 4: Patient-Doctor Pairing Web Portal

Project Description

You will be developing a website that helps patients seek doctors needed and also help doctors identify patients in need. A large number of individuals in parts of the developing world do not have access to proper health care. There are an inadequate number of healthcare professionals, limited access to information, and drugs are scarce.

This project aims to address the first and second challenges by linking the individuals with healthcare professionals. Your client would like to create a web app which attempts to link

patients with healthcare professionals through short message service (SMS). Willing healthcare professionals would register their contact details and their location and would receive messages from individuals with health problems who require information. The application would allow individuals with health issues send an SMS with a summary of their symptoms to the application which forwards the message to a healthcare professional who helps diagnose and suggests a line of treatment. It is a very practical project which would help a large number of individuals and we believe would be a very fun build. For simplicity, you don't have to implement the real SMS service.

Topic 5: Restaurant Automation

Project Description

This project develops a computerized system to help restaurant personnel coordinate their activities and improve their services, and for the management to track business growth and create future plans. The goal for this project is to introduce automation in privately-owned restaurants, that is, small to medium-sized establishments. Typical problems restaurant personnel are facing include: coordination of their work activities, anticipating and handling periods of low/high patron traffic, recognizing trends early enough to take advantage of bestsellers or abandon the flops, lowering operating costs, and increasing efficiency/productivity and profits. Many restaurants are still operated using pen and paper methods, with little or no automation. Patrons enter the facility to be greeted by a host, who often times has a "dry erase" diagram of the tables, maintained on a blackboard. The host can see the status of the tables based on whether or not they or someone else physically updates the diagram. Once seated a waiter tends to the customers by jotting down the orders onto a piece of carbon paper and delivers it to the kitchen for proper food preparation. The waiter then has to periodically check back to find out when the meal is ready. When the food is done, the piece of carbon paper is saved for proper record keeping by the management. This "old fashion" system works but yields a large amount of tab receipts, wastes a lot of time and is simply out-of-date. In old fashion systems, waiters have to carry pads around to take orders, always have a working pen and be sure to keep each bill organized and "synchronized" with the proper table.

This project automates restaurant operation so that all information pertaining to patron's orders and staff activity will be conveniently shared and stored over the restaurant's intranet. Hosts will be able to view table status with a click of a button. The wait staff will be able to enter the patron's orders quickly and efficiently and then have it electronically delivered to the kitchen. The kitchen staff will be able to view the incoming orders and notify the proper wait staff when the food is ready. Bus boys will be able to view real-time floor status allowing them to know which tables are clean, dirty, or occupied. Most importantly, all of the restaurant

information is organized and saved in the system database for the management viewing and archival. The analysis will consist of by-the-day and by-the-hour breakdowns of: revenue and revenue percentage per menu item, menu item popularity, Personnel efficiency, average turnaround time (how long patrons spend in the restaurant), average preparation time (time from when the order is placed to when it is ready). All data is automatically collected and processed allowing management to focus on analyzing the data rather than calculating it. (A more detailed requirements description can be found [here](#).)

Topic 6: Mobile Textbook Company

Project Description

This project aims to replace traditional textbooks with apps on slate devices like the iPad or Samsung Galaxy S5 (or any other mobile devices). The basic framework for displaying the material in the apps is a web page within the app. Because of this, most of the interactive elements of the app must be done in web-friendly technologies such as JavaScript. However, content creators (math teachers and professors who write the material and design the animations and interactive features) are generally not fluent in JavaScript themselves; they are simply good at explaining math and designing pedagogically useful interactive features and animations. The goal of this project will be to make software, either a web-based solution or a downloadable application, to facilitate the distributed generation of the interactive features and animations. The client needs the team of content creators, teachers and professors who are dispersed around the state and country, to be able to design interactive features and animations from home while writing the accompanying text etc.

The ultimate goal is to make software that allows users, the team of content creators, to design and build interactive features and mathematical animations, through a GUI. For the purposes of this course, however, we will start with a simple class of interactive features like simple 2D coordinate geometry interactive diagrams and go from there.

Topic 7: El Farol Bar Problem and the Minority Game

Project Description

This project aims to develop a software simulator that models a population of simple brains engaged in decision-making and learning. Detailed requirements can be found [here](#).

Topic 8: Parking Garage Automation

Project Description

This project develops a computerized system to manage parking usage and online reservations in a parking garage. Detailed requirements can be found [here](#).

Topic 9: Traffic Monitoring

Project Description

This project develops a system that will inform users about road and traffic conditions in the area of interest, by taking into account historic traffic and weather information. Detailed requirements can be found [here](#).

Topic 10: Virtual Biology Lab Experiments

Project Description

This project aims to create a virtual laboratory in which students learn the mechanics of mitosis, or cell division, through computer simulation. This virtual lab simulates lab exercises that are performed by students who are taking general or upper level biology courses. Mitosis is the continuous process that a parent cell undergoes to divide and create two identical cells, which are often called daughter cells. Scientists have divided the process into several discrete phases (usually four to six), each characterized by important events. Although this process captures many cells in different phases of the cell cycle, keep in mind that the cell cycle is a continuous process. Detailed requirements can be found [here](#).