CptS322 Software Engineering Principles I

Course Project

1. General Description

The project is the central part of the grade for this course. You will apply and exercise what you learn from this course in the process of this project so that the concepts and abstract understandings on the software engineering principles can be transformed into practical skills.

The goals of the project include the following:

1. Provide an environment for software engineering practice in a smaller scale, enabling you to practice and experience the complete software process within a single semester, and to encounter and learn how to solve common software engineering problems.
2. Facilitate basic understanding of various concepts about software development, such as requirements modeling and software design methodologies.
3. Foster and/or enhance the ability to conduct self-motivated demand-driven research---you will need to pick up a few other things that are not to be covered in this course in order to get the project well done.
4. Help harness basic software development skills such as project planning and team collaboration as well as software version control.

2. 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 3 to 4 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 yourself (e.g., a WSU department, a local company or other external organization, a member of faculty or staff, etc.). Some potential projects and clients will be suggested but you are encouraged to identify your own. There should be a firm intention by the client to use the software in production.

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.

The following are some example project topics you could select from if you opt not to come up with your own topic. For these suggested topics, the clients are the instructor and the TA.

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: Automaton simulator

Project Description

You will be developing a simulation tool of Finite State Automaton. The tool should be able to construct a well-defined automaton, that is, with five elements in the formal definition: States set, Alphabet, Transition set, Starting set, and Acceptance states set.

The tool should also be able to save/load an automaton (specification), and to read user inputs and decide whether the automaton accepts the inputs or not. A possible means for storing an automaton is to use XML, with which you may utilize an XML parser, revisit concepts on Finite State Automaton, Regular Languages, and Regular Expressions. You are also encouraged to add additional features of the automaton tool. However, a command-line user interface should be included allowing users just to specify an automaton and let the automaton recognize an input.

Topic 2: 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 an either 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 3: 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 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 4: 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 5: 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 6: 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 costumers 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 computerizes 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.

Other Experimental Topics:

- *Learning Management System*. Have a team build a learning management system from the ground up. Start with basic features such as user management, course file management, and course communications, and build up to advanced features such as complex assignments, learning analytics and software scaffolds to encourage virtuous learning behaviors.
- *Classroom Response System*. Have a team build a classroom response system like iClicker or TopHat. Initial development could focus on the basic infrastructure, such as allowing instructors to design questions and allowing students to answer them. Subsequent development could greatly expand the features to include instructor and student dashboards, custom instruction types and even integration with LMSs.
- *Hacker Rank "mini assignment" submission platform*. Have a team construct an online tool that allows professors to assign interview-style questions (e.g. reverse a linked list) that are automatically graded based on instructor-provided test cases.
- *Course Schedule Search*. If your institution has a way of providing course information through an API, or via screen scraping (with permission), students can build a way of searching current and past course schedules that has an improved user experience over that provided by the campus. Some examples of searches often not provided by the default system include, being able to search to see what courses an instructor has taught over time, which terms particular courses have been offered and who has taught them, course enrollments over time, the ability to put together a "prospective schedule" before official *enrollment starts, ability to share ones's course schedule with friends, etc.*
- *Letter of Reference Request Management System.* Students frequently need to request letters of reference from instructors for scholarships, graduate school applications, etc. This system would provide students a way to log these requests, and to see when their request is fulfilled, where they stand in the queue, etc. Instructors have a way to track requests, see the queue of unfulfilled requests, and track the time spent on this activity.

3. Team

Each team may have 4-6 members. One member in your team should serve as the contact – not necessarily the leader or manager, but will be responsible for the regular communications of your team with your instructor and TA, such as scheduling meetings for milestones check. By the deadline of the Milestone 0, submit your team information in a single PDF to the TA and the instructor, including the following elements:

- Project topic (if you come up with your own project, you need to provide the project description and expected target client of the project)

- Team name
- Members: (ID, name, EECS server login account name, of each member)
  a. One of the members is marked as the main point of contact
- Project name (should reflect the project topic chosen) and *private* Git (e.g. via bitbucket or github) repository URL for this course project

You may use the ‘Team Finder’ forum on Canvas Discussions to build your team. Students who couldn’t find teammates by the deadline should see the instructor to work around this as early as possible.

It is highly recommended that each team develop some sort of organizational structure (division of responsibilities, mechanism for making decisions, means for communication) in order to achieve both coherence and effective parallel effort. You may want to go Agile (e.g., Scrum). You may want a team leader; you may want to have a managerial lead and a design lead (cf. the Surgical Team concept from Brooks's Mythical Man Month). You may want to have functional specialization (e.g., design versus testing); you may want to assign one or two people per component; you may want to do Agile software development. You might want to rotate the lead for each milestone. You really, should have regular meetings; you should have a mailing list and chat set up, as well as a repository and issue tracker for all project information. Communication is your biggest risk, since you probably have disparate schedules, which mutates into a coherence/consistency risk, amongst others.

4. Milestones with deliverables

Below are the milestones that denote the major cycles of product development along with expected deliverable work products. The work products are repeatedly (and incrementally) defined and evaluated at increasing levels of detail. You should augment these major cycles with minor cycles as-needed to resolve individual risks, etc. Also, these milestones are dates for you to deliver documents to me and possibly code also to your project repository (I will not read your code although the TA might). *It is expected that your actual work precedes the milestones by quite a bit.* My grading of these documents will constitute the ``review and commitment'' parts of the cycle (although normally this would happen in a highly interactive and probing meeting); you are responsible for validation, which is a much more detailed activity than the review for commitment. More details on requirements for each milestone submission will be given to you when the milestone is released.

Milestone 0 – team formation and project repository creation
This is not a technical milestone to be graded, but it must be completed before other milestones are to be accomplished. Your team will be created and the GitHub repository will be set up. Your team will deliver a PDF for the team info as described above.

Milestone 1 – Tools, technologies, and process plan

Identify key tools, technologies, technical skills, and processes that you will be depending on throughout the rest of the project. Process should not only include the traditional aspects of a software process, but any kind of process you feel you need to adopt to be successful. Your team will deliver a PDF describing all these information, including the justification of why the software process is chosen for your project.

Milestone 2 – Requirements modeling: Scenario-based elements

Discover the needs of your customers (clients) based on the project description, and describe the requirements in terms of scenario-based elements of the requirements model. You will identify the preliminary use cases, elaborate them and finally document the work products. Your team will deliver a PDF including elaborated use case diagrams along with formal description of each use case.

Milestone 3 – Requirements modeling: Class-based elements

Further clarify customer requirements by identifying the class-based elements of the requirements model. This milestone focuses on data modeling instead of high-level use scenarios. Your team will deliver a PDF including all class diagrams, package diagrams, CRC index cards, and possibly other diagrams and documents. Note that for clarifying requirements, you could develop a prototype of the software at this point.

Milestone 4 – Design modeling: software architecture

Propose a high-level architecture for your project, identifying key components and the connections among them. Your team will deliver a PDF including the architecture context diagram and component diagrams that represent the architecture design.

Milestone 5 – Design modeling: software design

Perform data/class design, user interface design, and component-level design. Your team will deliver a PDF including corresponding design diagrams that form the design specification for your project.

Milestone 6 – Construction

Based on the design, now implement the software towards a functioning product. You will deliver a PDF describing the usage instructions for your software. Also, you need to submit code to the project repository.

Milestone 7 – Testing

To make sure your project is designed and implemented as expected, develop test cases to verify and validate the conformance of your implementation with the design as well as the conformance of your design with the requirements. Your team will deliver a PDF describing the test cases developed.

Milestone 8 – Presentation

Demonstrate that your project works as a functioning product. Your team will deliver a PDF including the screenshots showing your software running against test cases, and a retrospective summary of the project experience (e.g., lessons learned, problems struggled with, and afterthoughts). Each team will present the project demo and summary to the entire class, during which each team member will talk about the part of the project with respect to his/her contribution. The project presentation itself will account for a primary part of the grade for this final milestone.

5. Project Evaluation

The grade of this project consists of the grade per milestone received as a team and the peer evaluation score per team member. First, each member will receive the same grade as the team grade. Second, the project grade will be adjusted for each team member based on how teammates peer-evaluate the member. Details on peer evaluation will be given later.