

Evaluation of Project 1

CS 422 Software Methodologies

By Anthony Hornof - May 15, 2024

Criteria posted at: https://classes.cs.uoregon.edu/24S/cs422/Project_Evaluation.html

“Points: x / y ” indicates that x points were earned out of a maximum possible y points.

Group 7 - [Ryan Helms](#), [William Qiu](#), [Nikhar Ramlakhan](#), [Abie Safdie](#), [Caleb Sutherland](#)

Criteria for Evaluation, Points, and Comments

Initial Project Plan / SRS / SDS

Points: [13](#) / 15

The **Final Project Delivery** must be **complete** and **consistent**, must provide **instructions** on how to setup and and run your code. A README or equivalent must describe the code structure, and code must be documented. Code that you have used from other sources (not written entirely yourself) must be clearly indicated.

Application Quality

Application quality includes features used by all categories of user. Often this includes administrative users (sysadmins, etc.) in addition to end users. Functionality also includes performance and scalability.

Robustness. This evaluates the extent to which the system can be run with no bugs or errors, such as by failing to install and compile, by behaving differently than specified, or by crashing.

Points: [10](#) / 10

Feature Set. This evaluates the extent to which the system exceeded, met, or fell short of requirements. This considers requirements that were assigned and specified by the team. This evaluates whether essential features were missing and whether extended features were provided.

Points: [8](#) / 10

- [Good basic features.](#)
 - [Able to open PDFs, take notes, save the notes. Good.](#)
 - [Good example PDF files are loaded with the installation.](#)
-

Ease of Use. This category includes setup (if any) as well as usability for end users and for administrators. It is typically distinct from quality of documentation, but may include presence and usefulness of online help. Usability considerations for administrative users may be quite different than usability considerations for end users; both are considered here. For a prototype, this addresses both existing and planned features.

Points: [8](#) / 10

- [Does a good job supporting the basic required tasks.](#)
 - [Adequate for a prototype.](#)
 - [Some obvious issues, such as opening a window that fills the screen, and having the scroll bar far from the document being scrolled. But understandable for an initial version.](#)
-

User Documentation Including Installation and Setup. This evaluates the extent to which documentation clearly and accurately describes, for an end-user, how to accomplish real-world tasks using the software. This

includes how to install and set up the system. The documentation should be clear, accurate, well-written, well-organized, and complete.

Points: 10 / 10

- Good user documentation.
- Installation was straightforward. Good job.
- “Installation_Instructions.pdf” and “Server_Installation_Instructions.pdf” were both very good.
- “Server_Installation_Instructions.pdf” described the steps for setting up the DB very clearly. Good job.
- Perhaps the “Installation_Instructions.pdf” should direct the person installing to “Server_Installation_Instructions.pdf”.
- A small thing: “Installation_Instructions.pdf” directed to “Final Version” instead of “ARA Source Code Files”.

Organization, Planning and Technical Documentation

Technical documentation and system organization are evaluated together. Good documentation can make a good design evident, and poor documentation can doom a good design to degradation over time, but good documentation cannot compensate for poor design.

Project Plan. This section is evaluated on how effectively the team preforms project planning, how well it followed the plan, how well the team adapts the plan to inevitable hiccups, and whether risks are effectively considered, mitigated, and re-evaluated when plans change.

For full points, a report should clearly indicates who did what, when they did it, and how long everyone spent on each of their tasks (assigned date and completed date, as well as time on task). There should be a clear record of when meetings were held, and what was accomplished and agreed upon at each meeting. A series of continually-updated project plans should show the status of major project milestones and deliverables at regular intervals during the project lifecycle.

Points: 10 / 10

Systems Requirements (SRS). This should include all elements described in the initial SRS submission, updated to reflect what was actually built. This should include a clear, complete, and well-organized description of requirements, including a rationale for what is included, what has been deferred to the future, and (as appropriate) what has been excluded. The document should provide a clear user-centric specification as well as a well-defined, precise technical specification. Future developers should be able to use the document for system creation or maintenance.

Points: 10 / 10

System Architecture and Design Documentation. This should provide an easily understandable specification of the system architecture. Key architectural design decisions should be communicated, along with the rationale for each. The design should satisfy stated design goals. The software design is communicated as follows:

- Software Architecture: This section should describe the software structure by answering the following questions for a reader:
 - How is the software decomposed into components?
 - How do the components work together to implement the most important application features?
- Why was this particular design chosen? What is the rationale for any key design decisions?
- Module Interface Specifications: defines the interface of each component in the design including the services provided, parameters, and the effects of calling each service.

Points: 6 / 10

- Moderately developed and improved from the initial submission.
- Good job using more-descriptive names.
- A top-level Software Architecture and model diagrams based on what we learn about those topics in class.
- The figures show good thinking. They either need Keys or should be converted to standard diagrams such as UML.
- Database tables should have been described.
- The SDS was loosely similar to, but was not updated to clearly describe, what was actually built.

Technical Documentation. This evaluates the technical documentation; that is, the human-readable text that a programmer or human-installer would use to understand (a) how to install the software, (b) software dependencies, (c) required versions of components, (d) how source code files relate to each other, (e) the purpose and inner workings of each source code file, function, if-then statement, and loop, (f) each line of source code, and (g) the purpose, scope, and lifetime of each variable. This includes documents that are separate from the code as well as comments within the code.

Points: 8 / 10

- Good job commenting your code.
- Decent programmer's doc.
- Some technical documents could do a little better with "What is this thing?" at https://classes.cs.uoregon.edu/24S/cs422/good_writing.html
- Database tables should have been described.

Class Presentation. The presentation should clearly and accurately convey key lessons learned and the application of lessons from class.

Points: 5 / 5

- Thank you for following the advice at "How to Present". I hope you found the suggestions useful.

Numerical score before applying criterion: **88**

Class criterion for this assignment: **88** (The "criterion" policy is discussed in the course syllabus.)

Numerical score after applying criterion: 100 <== This is your score for this project.

11:05 - 11:13

(ix down)

CIS 422/522

How to Present Your Project in Class

A. Hornof - February 1, 2023

[The only updates on 2-1-2023 were (a) to add the section "LIVE DEMO" and (b) to change "REAL DEMO" to "REAL TASK".]

This document summarizes how to do group presentations for CIS 422 Software Methodologies. The document starts with the grading worksheet that will be used to evaluate your presentations, and is followed by a discussion of what and how to present.

Presentation Grading Worksheet

Group: 7 Date: 4-30-24

It is the responsibility of *all* group members to ensure that all of these criteria are met. You should practice your presentation, ideally in the actual classroom with the exact technology that you will use for your presentation, with all group members monitoring all of these criteria. You may help each other during the presentation (such as to ask a group member to move closer to the slide if they are pointing at something from far away).

Evaluation of Participation

- ☒ Y ☐ N - **SHOW UP.** All group members should be present.
- ☒ Y ☐ N - **EVERYBODY SPEAKS.** All group members should speak for roughly the same amount of time.
- ☒ Y ☐ N - **FULL NAMES.** The first slide should list all group members, including their first and last names. At the start of the presentation, all group members should introduce themselves with their first and last names.

Evaluation of Intellectual Content

- ☒ Y ☐ N - **WHAT.** Explain what you did, as described below under "What and How to Present".
- ☒ Y ☐ N - **HOW.** Explain how you did it, as described below under "What and How to Present".
- ☒ Y ☐ N - **DO A DEMO.** There should be a demo of the software. The demo should quickly and clearly convey how the software works.
- ☒ Y ☐ N - **LIVE DEMO.** The demo should be live, with the presenter interacting with the software in real time, not a video of the software running.
- ☒ Y ☐ N - **REAL TASK.** The demo should show a user doing a real task motivated by a real human need, with real data, not just clicking through the software with random meaningless button presses and keystrokes.
- ☒ Y ☐ N - **NO LOGIN.** The demo should *not* show a user logging in with a username and password. Write software that does not require a login, or log in before starting your presentation.
- ☒ Y ☐ N - **UNIQUE TECHNOLOGY.** The presentation should discuss technology as it is applied to this specific project, not how the technology is used to build a generic software program.
- ☒ Y ☐ N - **UNIQUE WORK.** Present information that is unique to your group's work, what you did that was different. For example, do not spend more than 10 seconds describing the assignment that everyone did, or the technology that everyone used.

Evaluation of Visual Materials

(slide & text a little small)
Y / N - READABLE TEXT. All text used in the presentation should be large enough to be readable from the rear of the classroom by a student with typical (20/20) vision. This includes text in diagrams and demos.
Y / N - READABLE DEMO. In the demo, the screen should be resized in so that all text is readable from the rear of the classroom.
diagram text was small

Y / N - SLIDES MAKE POINTS. Every slide should make a main point, with substantial information on the slide supporting the main point.

Y / N - USEFUL SLIDES. Visual material should convey the major points of your talk, and also provide substantial supporting material. Viewers should be able to just read the slides, with no audio, and understand the main points as well as the support for the main points. Do not use the Powerpoint "build" feature to reveal one bullet point at a time.

Y / N - NO CHARTJUNK. All of the visual images should be directly relevant to the project, and should help to explain your main points. Every pixel on the slide should either help to explain the points you are making, or should structure the talk (such as with slide numbers). If you show a photograph, the photo should directly support and help to explain a point you are trying to make. Do not use clip art unless it helps to make a point.

Evaluation of Quality of Presentation

Y / N - FINISH ON TIME. Finish within the allocated time.

Y / N - POSITIVE ATTITUDE. Present with a positive attitude. For example, do not sarcastically call attention to how you are fulfilling all of the presentation requirements listed here. Just focus on doing a good presentation.

Y / N - POINT WITH PRECISION. Direct the audience's attention to the things you are talking about. If you are talking about a module in an architectural diagram, don't just wave your hand in the direction of the slide from six feet away. Instead, move your pointing hand to within 12" of what you are pointing at. Hold your "upstage" arm steady in the air, pointing at the thing, while your body faces the audience. (If you are giving a remote presentation and you are referring to something on the screen, use your mouse cursor to point at that thing.)

Y / N - DON'T BLOCK. Stand to the side of your slide so that you are not blocking the slide for any of your viewers.

Y / N - PROJECT YOUR VOICE. Speak loudly and clearly. Project your voice. Fill the room with your thoughts and ideas.

Y / N - NO TECHNICAL DELAYS. There should be no preventable technical difficulties that cause more than one minute of delay.

What and How to Present

Please prepare visuals, which can be in the form of presentation slides or a word processing document that you just scroll through, as I do with my lecture notes. Please make sure the text is big enough to be seen from the back of the room. I am interested in relevant, easy-to-read content, especially content that is unique to your project, and that does more than restate general software engineering knowledge. I am not interested in irrelevant aesthetics such as interesting fonts and colors. For the last four topic areas below, please try to identify at least one interesting, important, big thing that your group learned about that topic.