

Homework #1

Problem 1.1: What are the basic tasks that all software engineering projects must handle?

- All software engineering projects must involve gathering requirements, high and low level design, development, testing, deployment, and maintenance.

Problem 1.2: Give a one sentence description of each of the tasks you listed in Exercise 1.

- Gathering requirements: Finding out what the customers want and need and turning these into requirements documents.
- High and low level design: High-level design involves what data design to use, what platform to use, interfaces with other systems, and high-level information about the project architecture. Low-level design has information about how smaller pieces of the project should work.
- Development: This stage involves beginning to program and implementing low-level designs into code.
- Testing: This task includes running tests and fixing code when tests fail.
- Deployment: Ideally for deployment, a programmer is able to roll out their software, but this task may be difficult, time-consuming, and expensive.
- Maintenance: Making sure to fix any issues when finding bugs in the code.

Problem 2.4: ...Compare this process to what you can do with GitHub versions. How are the two tools different? How are they the same?

- After viewing the history between the two document versions, I noticed that I was able to see all of the changes that were made to my document. The original blank document was seen under Version 1 in the version history, and some of the changes I made were also seen but under the name Version 2. When comparing Google Docs to GitHub, some similarities I have observed are that both collaborative tools allow users to create and view past versions of their work. Naming versions in Google Docs is similar to commit names in GitHub. One difference I noticed includes that commit messages in GitHub are required, whereas naming versions of Google Docs is an optional task. GitHub also allows for branching where users work individually and then merge their code into one main repository; on the other hand, users utilizing Google Docs must all work on the same document. Finally, Google Docs is designed more for text documents, and GitHub is focused more for coding and software development.

Problem 2.5: What does JBGE stand for and what does it mean?

- JBGE stands for “just barely good enough.” JBGE means that much time is wasted when changing code and updating documentation if someone includes too much documentation. Therefore, in software engineering, the comments and code documentation provided should be “just barely good enough.”

Problem 4.2

- a. Use critical path methods to find the total expected time from the project's start for each task's completion.
- b. Find the critical path. What are the tasks on the critical path?
 - The critical path is $G \rightarrow H \rightarrow I \rightarrow D \rightarrow E \rightarrow M \rightarrow Q$
 - The tasks on the critical path are rendering engine, humanoid base classes, character classes, character editor, character animator, character library, and character testing.
- c. What is the total expected duration of the project in working days?
 - $G(6) + H(3) + I(3) + D(6) + E(7) + M(9) + Q(4) = 32$
 - The total expected duration of the project is 32 working days.

Problem 4.4: Build a Gantt chart for the critical path you drew in Exercise 2. Start on Wednesday, January 1, 2024, and don't work on weekends or the following holidays:

[illegible]

Problem 4.6: ...How can you handle these sorts of completely unpredictable problems?

- These sorts of completely unpredictable problems can be handled by using risk management. This proactive method finds possible risks, determines their potential impacts, and studies any possible work-arounds ahead of time. Risk management determines the likelihood, severity, consequences, and work-arounds for each task.

Problem 4.8: According to your textbook, what are the two biggest mistakes you can make while tracking tasks?

- The two biggest mistakes you can make while tracking tasks are ignoring a problem and hoping to make up the time later as well as piling extra developers on a specific task and assuming they will be able to reduce the time needed to finish it.

Problem 5.1: List five characteristics of good requirements.

- Good requirements are clear, unambiguous, consistent, prioritized, and verifiable.

Problem 5.3: ...For this exercise, list the audience-oriented categories for each requirement. Are there requirements in every category?

- a: user requirements
- b: functional requirements
- c: functional requirements
- d: functional requirements
- e: user requirements
- f: nonfunctional requirements
- g: nonfunctional requirements
- h: functional requirements
- i: functional requirements
- j: functional requirements
- k: functional requirements
- l: user requirements
- m: user requirements
- n: user requirements
- o: functional requirements
- p: functional requirements
- No, these requirements were missing requirements from the business and implementation categories.

Problem 5.9: ...Brainstorm this application and see if you can think of ways you might change it. Use the MOSCOW method to prioritize your changes.

- Must:
 - Adding a sign that says “This letter has already been selected” if a player clicks on a letter they had already chosen
 - Making sure the game works well on different sizes of smartphones
- Should:
 - Including a scoreboard to keep track of a player’s wins and losses after many games have been played
 - Incorporating sound effects for when a letter is chosen correctly / incorrectly
- Could:
 - Providing an option for a player to receive a hint
 - Allowing users to select categories of words or change the theme / color of the game’s current design
- Won’t:
 - Adding a multiplayer option so that more than one user can compete against each other