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

Gathering requests (as much as possible), do the high-level design and the low-level design, write code and test them during the development, release it, keep up with it, and before putting it on the side, and wrap-up everything.

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

Gathering requests: get information on what is wanted, and how it is wanted.

high-level design: deciding the environment the project is going to be used on, and the parts the project will be made in.

low-level design: deciding how each part will function, and the connection they have with each other.

development: the phase of writing code.

testing: test all the possible situations, find errors, and fix them.

deployment: hand the project to the user and make sure they can use it properly.

maintenance: get the user's feedback, and fix more bugs or add more features accordingly.

Wrap-up: conclude and evaluate the right and wrongs during the whole process.

2.4 Compare this process (of versions in Google Doc) to what you can do with GitHub versions. How are the two tools different? How are they the same?

Github will provide different branches for different people to work on, and only on merging them will make one see what others are doing.

Google doc will mark out the changes made on the same document on live, with different color represent different user.

They both have version control and can be edit by multiple people. Github will show commits on the branch while Google doc don't. Google doc's new version is created before any change should made, while Github's release happens after the content is ready.

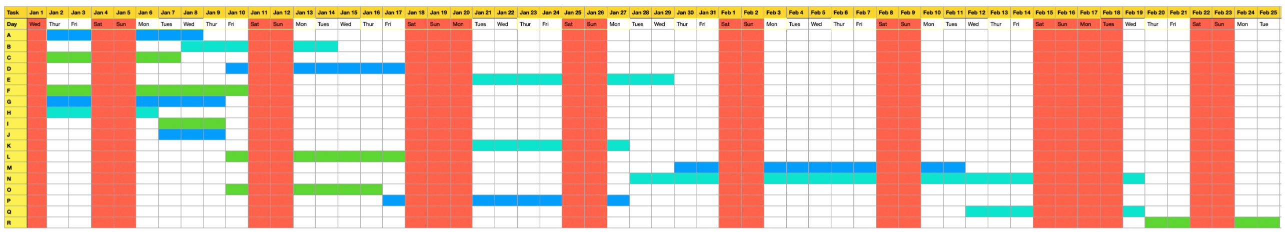
2.5 What does JBGE stand for and what does it mean?

"Just barely good enough". It is a way to write documents and comments for the code, to suggest not to waste too much time on writing them, but that all the things needed are included in them.

4.2 Use critical path methods to find the total expected time from the project's start for each task's completion. Find the critical path. What are the tasks on the critical path? What is the total expected duration of the project in working days?

The critical path is R -> N -> O -> G, representing Zombie testing, Zombie library, Zombie editor and Rendering engine, in total 30 days.

4.4 Build a Gantt chart for the network you drew in Exercise 3.



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

Extend the time on the schedule, or add tasks on the schedule that represent the time for any unpredictable problem.

4.8 What are the two biggest mistakes you can make while tracking tasks?

Ignore the problem and thinking of making it up, which will cause in a slip in schedule and usually will cause a delay on the schedule. The second mistake is thinking that having more developers can speed up the process of development.

5.1 List five characteristics of good requirements.

Clear, unambiguous, consistent, prioritized, and Verifiable.

5.3 Suppose you want to build a program called TimeShifter to upload and download files at scheduled times while you're on vacation. The following list shows some of the applications requirements. For this exercise, list the audience-oriented categories for each requirement. Are there requirements in each category?

- a. Allow users to monitor uploads/downloads while away from the office.
  user requirement
- b. Let the user specify website log-in parameters such as an Internet address, a port, a username, and a password.
  function requirement
- c. Let the user specify upload/download parameters such a number of retries if there's a problem.
  user and function requirement
- d. Let the user select an Internet location, a local file, and a time to perform the upload/download.
  user requirement
- e. Let the user schedule uploads/downloads at any time.
  user requirement
- f. Allow uploads/downloads to run at any time.
  nonfunctional requirement
- g. Make uploads/downloads transfer at least 8 Mbps.
  functional requirement
- h. Run uploads/downloads sequentially. Two cannot run at the same time.
  nonfunctional requirement

- i. If an upload/download is scheduled for a time whan another is in progress, it waits until the other one finishes.
  nonfunctional requirement
- j. Perform schedule uploads/downloads.
  functional requirement
- k. Keep a log of all attempted uploads/downloads and whether the succeeded.
  functional requirement
- l. Let the user empty the log.
  user requirement
- m. Display reports of upoad/download attempts.
  functional requirement
- n. Let the user view the log reports on a remote device such as a phone.
  user requirement
- o. Send an e-mail to an administrator if an upload/download fails more than its maximum retry number of times.
  functional requirement
- p. Send a text message to an administrator if an upload/download fails more than it's maximum retury umber of times.
  functional requirement

I don't see any business requirement in this list. I think it is because this list is about what the application will do but not how it will, and it is not within the business requirement range.

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: There must be a "are you sure" notice when new game button is pressed.

There must be a "do you want a new game" notice when the current game ended.

The word for guessing must be a rather common word, not from some old medical book back in 10th century.

Should: There should be config buttons to let the users change settings such as BGM volumes, keyboard style or background color.

The records should be saved, on how many games have been played, the count on successes, the rate of success, the fastest speed record.

The user should be able to save their process.

Could: The letters on the right does not have to be in keyboard shape, and the letters could be bigger since the users are clicking with finger.

The new game button could be somewhere else away from the letters.

The user could choose the range of word (like in a category).

Won't: There won't be any internet connection requirement for this game.