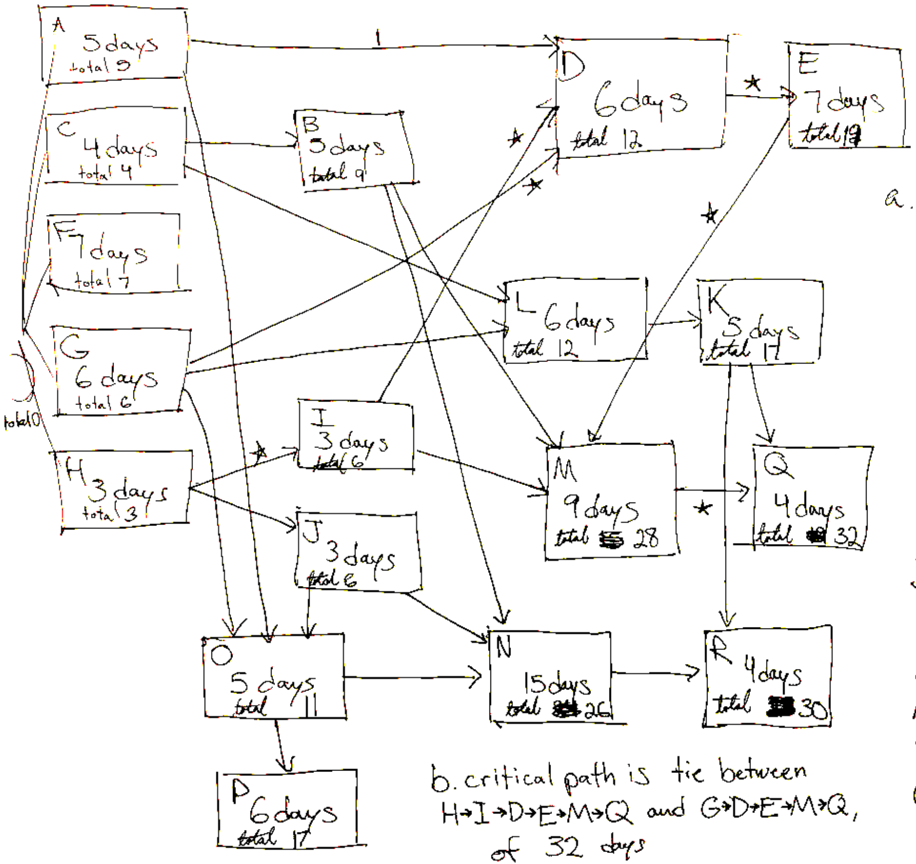
1.1
All software engineering projects must:
gather requirements, high-level design, low-level design, testing, development, deployment, maintenance, wrap-up

1.2
gather requirements: Identify your customers and figure out what the customer wants and needs.
high-level design: Make decisions about what platform, data design, interfaces, and other systems to use in your project.
low-level design: split larger pieces of the design into smaller lower-level pieces.
development: Program is continually refined until everything is implemented in code.
testing: perform different kinds of tests to check for edge-cases or mistakes in the code.
deployment: roll out the software, check for new vulnerabilities, mistakes, user needs, etc, that one can't possibly predict.
maintenance: users find bugs, devs fix bugs.
wrap-up: evaluate the project and decide what went right and what went wrong.

2.4
Differences between versions are displayed as highlights, things removed are displayed as the previous version struck through. This is similar to GitHub's ability to show you lines of code that are specifically added to or removed from a file. However, it does not give you the ability to fork changes between versions into separate branches of versions, nor is it giving me the ability to commit only specific changes between versions directly. I would still have to copy and paste the parts I want to revert manually.

2.5
JBGE stands for "Just Barely Good Enough". It refers to the need in software engineering to provide just enough code documentation and comments so that revising and updating it isn't too cumbersome as changes are made to the code. The idea is that you can minimize the amount of documentation that you have to produce. This doesn't mean that documentation can be skipped entirely, however.

4.2



a.

A: 5 days
B: C→B 9 days
C: 4 days
D: G→D or H→I→D 12 days
E: G→D→E or H→I→D→E 19 days
F: 7 days
G: 6 days
H: 3 days
I: H→I 6 days
J: H→J 6 days
K: G→L→K 17 days
L: G→L 12 days
M: A→D→E→M 28 days
N: G→O→N or H→J→O→N 26 days
O: G→O or H→J→O 11 days
P: G→O→P H→J→O→P 17 days
Q: G→D→E→M→Q 32 days
R: G→O→N→R 30 days

b. critical path is tie between H→I→D→E→M→Q and G→D→E→M→Q, of 32 days

H→I→D→E→M→Q or

c. 32 days

a)

Tasks & Durations

A: 5 days

B: C → B = 9 days

C: 4 days

D: G → D or H → I → D = 12 days

E: G → D → E or H → I → D → E = 19 days

F: 7 days

G: 6 days

H: 3 days

I: H → I = 6 days

J: H → J = 6 days

K: G → L → K = 17 days

L: G → L = 12 days

M: A → D → E → M = 28 days

N: G → O → N or H → J → O → N 26 days

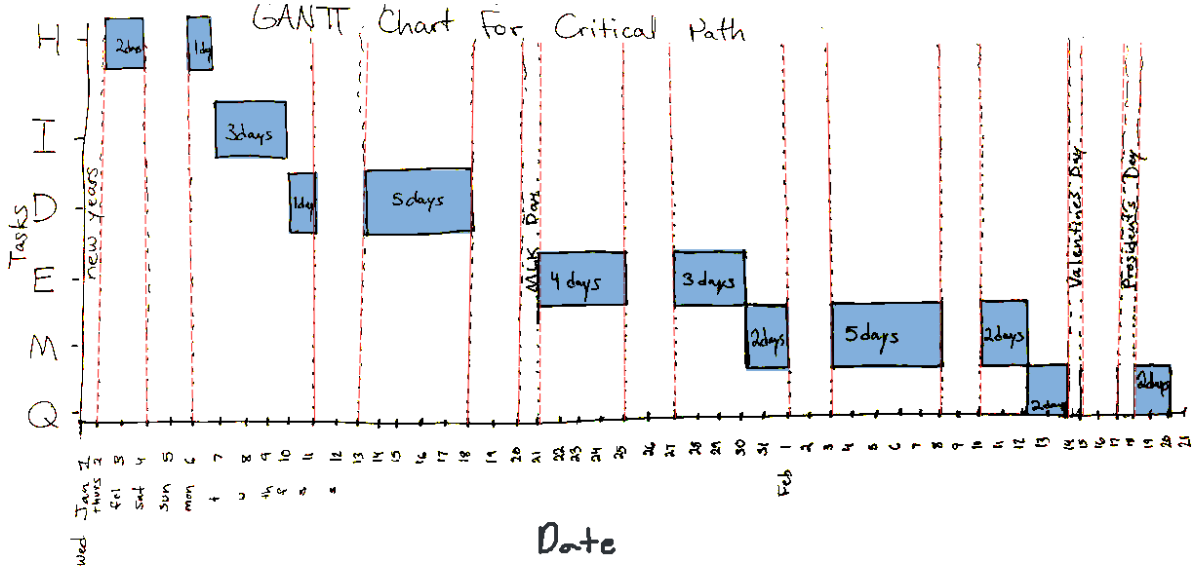
O: G → O or H → J → O 11 days

P: G → O → P or H → J → O → P = 17 days
Q: G → D → E → M → Q or H → I → D → E → M → Q = 32 days
R: G → O → N → R = 30 days

b) The critical path is a tie between:
H → I → D → E → M → Q
G → D → E → M → Q
Duration: 32 days

c) Total project duration: 32 days

4.4



4.6
You can treat deus ex machina problems the same way you handle unexpected sick leave. Add tasks at the end of the schedule to account for completely unexpected problems. When one of these problems does occur, insert its lost time into the schedule.

4.8
The biggest mistake you can make while tracking tasks is not taking action when a task slips. At a minimum, you need to pay closer attention to the task so that you can take action if it's in trouble.
The second biggest mistake is piling more people on the task and assuming they can cut the total time. Unless the new people have particularly useful expertise, bringing them up to speed may make the task take even longer (Like the Mythical Man-Month).

5.1
good requirements are clear (easy to understand), unambiguous, consistent, prioritized, and verifiable.

5.3
Requirement Category
a. Allow users to monitor uploads/downloads while away from the office. Monitoring & Logging
b. Let the user specify website log-in parameters such as an Internet address, a port, a username, and a password. User Interaction & Configuration
c. Let the user specify upload/download parameters such as number of retries if there's a problem. User Interaction & Configuration
d. Let the user select an Internet location, a local file, and a time to perform the upload/download. User Interaction & Configuration
e. Let the user schedule uploads/downloads at any time. Scheduling & Execution
f. Allow uploads/downloads to run at any time. Scheduling & Execution
g. Make uploads/downloads transfer at least 8 Mbps. Performance & Constraints
h. Run uploads/downloads sequentially. Two cannot run at the same time. Scheduling & Execution
i. If an upload/download is scheduled for a time when another is in progress, it waits until the other one finishes. Scheduling & Execution
j. Perform scheduled uploads/downloads. Scheduling & Execution
k. Keep a log of all attempted uploads/downloads and whether they succeeded. Monitoring & Logging
l. Let the user empty the log. Monitoring & Logging
m. Display reports of upload/download attempts. Monitoring & Logging
n. Let the user view the log reports on a remote device such as a phone. Monitoring & Logging
o. Send an email to an administrator if an upload/download fails more than its maximum retry number of times. Error Handling & Notifications
p. Send a text message to an administrator if an upload/download fails more than its maximum retry number of times. Error Handling & Notifications
Yes, there are requirements in every category.

5.9
Must-Have (M):
Basic Hangman Gameplay – Clicking letters to guess the mystery word.
Letter Disabling After Selection – Prevents users from clicking the same letter again.
Win/Loss Conditions – Display messages for winning or losing.
New Game Button – Allows starting a fresh game.
Random Word Selection – Ensures variety in gameplay.

Should-Have (S):
Hint System – A button that provides a small clue.
Difficulty Levels – Easy, Medium, and Hard word lists.
Scoring System – Track number of attempts and wins/losses.
Timer-Based Challenge Mode – Adds an extra level of difficulty.
Sound Effects – Play sound when selecting a letter, winning, or losing.

Could-Have (C):
Custom Word List – Let users enter their own words.
Multiplayer Mode – Compete against a friend by taking turns.
Character Animations – Make Mr. Bones react dynamically.
Leaderboards – Track top scores globally or locally.
Themed Word Categories – Movie titles, animals, countries, etc.

Won't-Have (W):
Augmented Reality (AR) Mode – Display the skeleton in a 3D environment.
Voice Recognition – Allow guessing letters via speech.
Cross-Device Syncing – Save progress across multiple devices.
AI Opponent – A competitive bot that plays against the user.
Customizable Skeleton Designs – Change Mr. Bones' appearance.