Question 1 and 2: Basic tasks and description of each:

1. Requirements Gathering: Identifying and documenting what the software application must do to meet users' needs.
2. High-Level Design: Defining the system's architecture and major components, outlining how the system will meet the requirements.
3. Low-Level Design: Detailing the design of individual components and modules, specifying how they will be implemented.
4. Development: Writing the actual code to build the software application based on the design specifications.
5. Testing: Evaluating the software to ensure it functions correctly and meets the specified requirements.
6. Deployment: Releasing the completed software to users and making it operational in the target environment.
7. Maintenance: Updating and improving the software after deployment to fix issues and add new features.

Question 3.

For GOOGLE DOCS: Allows users to name and view different versions of a document. Displays a list of versions with timestamps and editor information. Highlights changes made between versions when a specific version is selected.

For GITHUB: Enables comprehensive version control for code repositories using Git.Tracks changes at the file and line level, showing additions and deletions.Supports branching and merging, facilitating collaborative developmentProvides commit histories with messages, timestamps, and author information.

Similarities

- Both tools track changes over time and allow users to view previous versions.
- Both provide information about who made changes and when.

Differences:

- Google Docs is primarily designed for text documents and offers a straightforward interface for viewing version history.
- GitHub is tailored for software development, offering advanced features like branching, merging, and detailed commit histories.
- GitHub's version control is more granular, tracking changes at the code level, whereas Google Docs focuses on document content changes.

Question 4: What does JBGE mean and stand for?

JBGE stands for "Just Barely Good Enough." It refers to a development approach where the software is built to meet the minimum requirements necessary for functionality, avoiding over-engineering or adding unnecessary features.

Question 5 and 6: Zombie game problem

a. Critical method is D → E → M → Q
b. The tasks are Character Editor, Character Animation, Character Library, Character Testing
c. Working days: 32 days

Question 6: How to handle completely unpredictable problems?

To handle completely unpredictable problems, the book suggests breaking unknown tasks into simpler pieces, looking for similarities with past projects, and expecting the unexpected. This involves anticipating potential issues and planning accordingly to manage unforeseen challenges

Question 7: Biggest mistake when tracking tasks?

Not tracking progress accurately: Failing to monitor the actual progress of tasks can lead to misunderstandings about the project's status.

Ignoring risks: Overlooking potential risks can result in being unprepared for issues that may arise, adversely affecting the project's outcome.

Question 8: Five characteristics of good requirements:

Clear: The requirement is easily understood without ambiguity.

Unambiguous: The requirement has only one interpretation.

Consistent: The requirement does not conflict with other requirements.

Prioritized: The requirement's importance is ranked relative to others.

Verifiable: The requirement can be tested or measured to confirm its implementation.

Question 9: Timeshifter Problem

Categories

Matching Requirements:

User Interface (UI) Requirements- Features that relate to how the user interacts with the software.:

- Allow users to monitor uploads/downloads while away from the office.
- Let the user specify website log-in parameters such as an Internet address, a port, a username, and a password.
- Let the user specify upload/download parameters such as the number of retries if there's a problem.
- Let the user select an Internet location, a local file, and a time to perform the upload/download.
- Let the user schedule uploads/downloads at any time.
- Let the user empty the log.
- Display reports of upload/download attempts.
- Let the user view the log reports on a remote device such as a phone.

Functional Requirements- Features describing what the software must do.**:**

- Allow users to monitor uploads/downloads while away from the office.
- Let the user specify upload/download parameters such as the number of retries if there's a problem.
- Let the user select an Internet location, a local file, and a time to perform the upload/download.
- Let the user schedule uploads/downloads at any time.
- Perform scheduled uploads/downloads.

Performance Requirements- Expectations related to speed, efficiency, and resource usage.**:**

- Make uploads/downloads transfer at least 8 Mbps.

Operational Requirements- Constraints on how the software operates.**:**

- Allow uploads/downloads to run at any time.
- Run uploads/downloads sequentially. Two cannot run at the same time.
- If an upload/download is scheduled for a time when another is in progress, it waits until the other one finishes.

Security Requirements- Requirements ensuring data protection and user authentication.**:**

- Let the user specify website log-in parameters such as an Internet address, a port, a username, and a password.

Logging/Notification Requirements- Features related to tracking events and alerting users.:

- Keep a log of all attempted uploads/downloads and whether they succeeded.
- Let the user empty the log.
- Display reports of upload/download attempts.
- Let the user view the log reports on a remote device such as a phone.
- Send an email to an administrator if an upload/download fails more than its maximum retry number of times.
- Send a text message to an administrator if an upload/download fails more than its maximum retry number of times.

Question 10: MR. Bones

MUST-HAVE

1. Ensure the game correctly selects a random word from a predefined word list.
2. Implement clickable letter buttons that are disabled after selection.
3. Display the current state of the word, showing correct guesses.
4. Show a visual representation of Mr. Bones that updates with incorrect guesses.
5. Display win/loss messages when the game ends.
6. Provide a New Game button to restart.

SHOULD-HAVE

7. Multiple difficulty levels
8. Hints system
9. Sound effects for correct/incorrect guesses and game-over scenarios.
10. Animated Mr. Bones
11. Score tracking

COULD-HAVE

12. Themed word lists
13. Customization options
14. Multiplayer mode

15. Daily Challenge Mode
16. Leaderboard for high scores.

WON’T-HAVE

17. 3D animations
18. Augmented Reality
19. Voice input for guessing letters
20. Complex online multiplayer