1. All software engineering projects must gather requirements, design at both the high and low level, developed, tested, and then deployed and maintained. Then a post-mortem can be written.
2. Description of tasks:
    a. Gather Requirements: Discover the needs and wants of the customer and decide what to build.
    b. High-level Design: Deciding the architecture of the project at a high-level, such as what platform to use, or what major pieces/tasks are needed to be fulfilled.
    c. Low-level Design: Deciding how each element should work, in terms of how it is built.
    d. Development: Team members work towards building each element, bringing the low-level designs into reality.
    e. Testing: Testing the code to ensure it works on its own and alongside other components.
    f. Deployment: Rolling out the product for users to be able to use it, involves distribution and support for the users.
    g. Maintenance: After deployment, users will find bugs and they need to be fixed.
    h. Wrap-Up: Writing or compiling a post-mortem to decide what went right and wrong.
3. Newly added content is highlighted and unchanged content is left as is. This is very similar to github version control. Changes and deletions are highlighted between versions. One notable difference is that, in larger files, Github will omit large portions of unchanged code in the changes summary to focus on the changes.
4. “Just barely good enough” means that if you spend too much time on documentation you could have spent that time on development. The idea is to document just enough, rather than document every little thing.
5. H→J→O→N requires Humanoid base classes, Zombie classes, Zombie editor and Zombie library. This will take 26 days.
6.

Task	January 2024																								
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
C		▮	▮	▮	▮																				
B						▮	▮	▮	▮	▮															
N											▮	▮	▮	▮	▮	▮	▮	▮	▮		▮	▮	▮	▮	▮

7. Leaving room in Risk Management for unexpected risks can help mitigate losses from unpredictable problems.
8. The two biggest mistakes you can make are assuming that you can make up lost time later, and piling on more workers on a behind task to reduce the time to finish it. Often lost time will not be made up without serious tradeoffs, and additional manpower usually comes with more training, thus adding more time to the task.

9. Good requirements are clear, unambiguous, consistent, prioritized, and verifiable.
10. Audience-oriented categories:
    a. Business Requirement
    b. Functional Requirement
    c. User Requirement
    d. Functional Requirement
    e. User Requirement
    f. Business Requirement
    g. Nonfunctional Requirement
    h. Nonfunctional Requirement
    i. Nonfunctional Requirement
    j. Functional Requirement
    k. User Requirement
    l. User Requirement
    m. User Requirement
    n. User Requirement
    o. User Requirement
    p. User Requirement
11. MOSCOW items:
    a. Must: Make the keyboard larger so a user can press the buttons easier with their fingers.
    b. Should: Add a confirmation prompt when pressing the New Game button to ensure users do not accidentally start a new game.
    c. Could: Refine the aesthetics, especially by removing the whitespace behind Mr. Bones.
    d. Won't: Add a system to generate new items to add to the list of mystery words.