Sebastian Cruz

CMSI 4072 Assignment #1

1.1

* Requirements Gathering
* High-Level Design
* Low-Level Design
* Development
* Testing
* Deployment
* Maintenance
* Wrap-up

1.2

* Requirements Gathering
  + Learning what the customer needs and wants
* High-Level Design
  + Describing the major components of the application (such as platform and interface) and how they function/interact with each other
* Low-Level Design
  + Describing how to actually build/program the pieces of the application for programmers
* Development
  + Creating the code to implement the application
* Testing
  + Using the application in different situations to observe for any bugs or flaws in the functionality of the application
* Deployment
  + Rolling out your software to users
* Maintenance
  + Doing bug fixes and adding adjustments/enhancements to future versions of the application
* Wrap-up
  + Evaluating the project’s timeline to determine what worked well and what didn’t for future projects

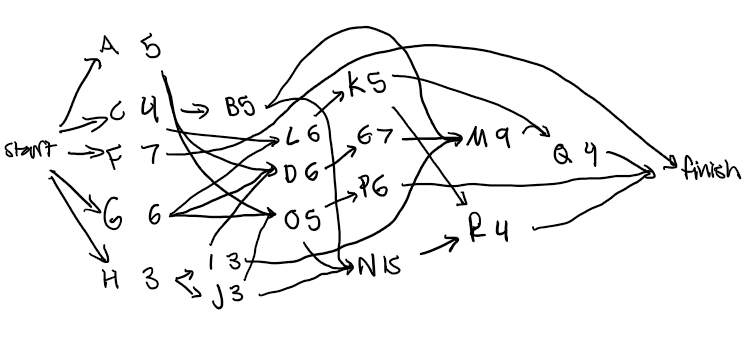
2.4

* Google Docs
  + Any changes are timestamped and shown who has edited the file
    - Words will be crossed out if deleted and highlighted in the users respective color if added in
* Compared to GitHub
  + Very similar!
    - Both can
      * Display specific changes made to the file
      * Show what has been added/deleted
      * Shows who made the changes with timestamps
      * Make branches/copies
  + Github can
    - Make changes be approved through owner/main user
    - Quickly show what specific files have been changed since last commit

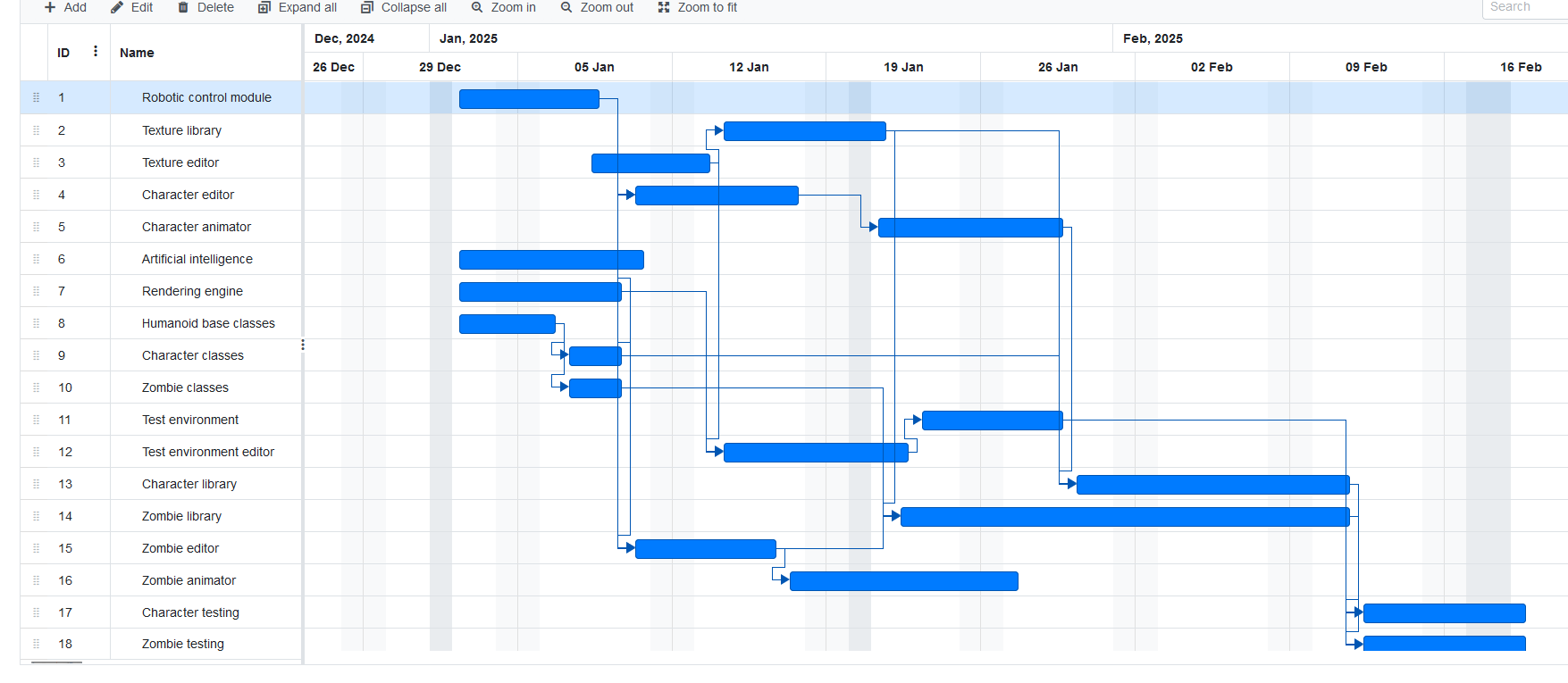
2.5

* Just Barely Good Enough
  + School of thought that says you should provide code docs and comments that are no more than what is absolutely necessary
    - Not being excessive

4.2

* 
* The critical path passes through tasks G -> D -> E -> M -> Q for a total expected duration of 32 working days

4.4

* NOTE: I am just using 2025 so that January 1 starts on a Wednesday
* 
* Expected end date is Feb 19

4.6

* To handle these types of unpredictable problems, you can handle them in the same manner as a sick leave
  + Add tasks at the end of the schedule to account for these deus ex machina problems
    - Insert its lost time into the schedule

4.8

* Not doing anything when a task slips out of alignment with the schedule
* Putting more people on the task, people who may not be up to speed on the task at hand

5.1

* Clear/easy to understand
* Unambiguous
* Consistent
* Prioritized
* Verifiable

5.3

* a. Allow users to monitor uploads/downloads while away from the office.
  + Business
* b. Let the user specify website log-in parameters such as an Internet address, a port, a username, and a password.
  + User/Functional
* c. Let the user specify upload/download parameters such a number of retries if there's a problem.
  + User/Functional
* d. Let the user select an Internet location, a local file, and a time to perform the upload/download.
  + User/Functional
* e. Let the user schedule uploads/downloads at any time.
  + Nonfunctional
* f. Allow uploads/downloads to run at any time.
  + Nonfunctional
* g. Make uploads/downloads transfer at least 8 Mbps.
  + Nonfunctional
* h. Run uploads/downloads sequentially. Two cannot run at the same time.
  + Nonfunctional
* i. If an upload/download is scheduled for a time whan another is in progress, it waits until the other one finishes.
  + Nonfunctional
* j. Perform schedule uploads/downloads.
  + Functional
* k. Keep a log of all attempted uploads/downloads and whether the succeeded.
  + Functional
* l. Let the user empty the log.
  + User/Functional
* m. Display reports of upoad/download attempts.
  + User/Functional
* n. Let the user view the log reports on a remote device such as a phone.
  + User/Functional
* o. Send an e-mail to an administrator if an upload/download fails more than its maximum retry number of times.
  + User/Functional
* p. Send a text message to an administrator if an upload/download fails more than it's maximum retury umber of times.
  + User/Functional
* There are no requirements for implementation
  + An implementation requirement could be buying new network hardware in order to support the network functionality of the application, but since it seems like the application can already do downloads/uploads, there is no implementation requirement

5.9

* Must
  + Score system/streak
* Should
  + New UI layout
  + Highscore leaderboards
* Could
  + Animations for win/lose/guessing
  + Different difficulties for guessing
  + Sounds/audio
  + Updated graphics
  + Different Mr. Bones poses
* Won’t
  + Time limit
  + New letter selection method