

Mitchell Simon

Prof. Johnson

Software Engineering Lab

29 January 2018

Software Engineering Assignment 1

#1.1: Requirements Gathering, High-level Design, Low-level Design, Development, Testing, Deployment, Maintenance, Wrap-Up

#1.2: Requirements gathering is typically communicating with the customer to find out what specific functionalities and specifications the software will have. High-level design is where the software engineer describes the high level components of the application and how they interact to create the whole. At the low level design is where specifics about how to build the different components and implement them. Development is the action of creating the code in whatever methodology and following whatever specifications were created. Testing is where the software engineer can test for flaws under normal use conditions and stress tests. Maintenance is whatever work, enhancements, fixes, additions, and versions you produce iteratively there after. Wrap-up is a time to reflect upon the project and see what went well and what didn't.

#2.4: Done. This assignment was completed using Google Docs.

#2.5: Just barely good enough. Which is what some believe documentation and

comments should be, in order to write them and keep up with a development timeline.

#3.2: G, D, E, M, Q is the critical path. With an expected length of 32 days.

#3.4: Using the Gantt chart you can see the game is expected to be finished February 18.

#3.6: When faced with unpredictable problems, you must adapt and move forward. Usually by reinstating the tasks into the timeline and adjusting when necessary.

#3.8: The two biggest mistakes are not adapting when a task slips and adding more developers to a task to speed it up. If a task slips you must adapt and reschedule it. Also, unless the developer you add to a task knows all the prerequisite knowledge regarding a task it will not speed up.

#4.1: Consistent, easy to understand, prioritized, verifiable, unambiguous.

#4.3: Business: a

User: b, c, d, l, m, n, o, p

Functional: b, c, d, j, k, l, m, n, o, p

Nonfunctional: e, f, g, h, i

Implementation: n/a

All categories have requirements except for implementation, as the application has already been implemented.

#4.9: Must: Find a way to make money. Either charge for the game, for micro-transactions, or advertisement.

Should: Allow keyboard entry, keep score, difficulty setting, leaderboard.

Could: Redesign visuals, pretty ugly. Local multiplayer.

Won't: Online leaderboard, online multiplayer. Expansion pack.