Nicholas Sunga

January 27, 2018

CMSI 402

Professor Johnson

Homework One

**1.1 What are the basic tasks that all software engineering projects must handle?**

All software engineering projects must handle: requirements gathering, high level design, low level design, development, testing, deployment, maintenance, and wrap ups.

**1.2 Give a one sentence description of each of the tasks you listed in Exercise 1.**

Requirements gathering: pertains to getting data about what the vendor or customer wants in their product.

High level design: envisioning and designing how major components of the application work together. Essentially, an understanding of how the product works and is easy to comprehend.

Low level design: the nitty gritty details of high level design. Explains the internal workings of the application for programmers to understand and contribute.

Development: the process of writing code.

Testing: unit testing the application for bugs before deployment.

Maintenance: scaling and bug fixes for future renditions of the application.

Wrap-up: Highlight the events that went right and the ones that went wrong. This is a way to implement the good and fix the bad in future project development.

**2.4 Like Microsoft Word, Google Docs [sic] provides some simple change tracking tools. Go to**[**http://www.google.com/docs/about/**](http://www.google.com/docs/about/)**to learn more and sign up [if you do not have an account already]. Then create a document, save it, close it, reopen it, and make changes to it as you did in Exercise 1.**

I am unsure of what to say. Here is the document I used: <https://docs.google.com/document/d/1g8PSmqmbOs_qLjyk50mHxomkGfZTyJ4e6Hv-KWE49Q/edit?usp=sharing>

**2.5 What does JBGE stand for and what does it mean?**

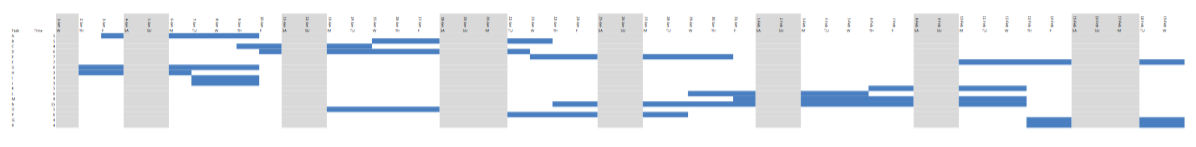
JBGE stands for “Just Barely Good Enough.” It is a saying that is centered around doing as much as needed. So, do not write more code than that which is needed.

**3.2 Use critical path methods to find the total expected time from the project's start for each task's completion. Find the critical path. What are the tasks on the critical path? What is the total expected duration of the project in working days?**

Critical path: GDEMQ

Tasks: rendering engine, character editor, character animator, character library, character testing.

Expected duration: 32 days

**3.4 Build a Gantt chart for the network you drew in Exercise 3. [Yes, I know, you weren't assigned that one — however, when you do Exercise 2 you should have enough information for this one.] Start on Wednesday, January 1, 2020, and don't work on weekends or the following holidays:** 

I know it is small… Blue areas are progress days for each task and grey areas are “no work days”. Project done by Feb 19th.

**3.6 In addition to losing time from vacation and sick leave, projects can suffer from problems that just strike out of nowhere. Sort of a bad version of *deus ex machina*. For example, senior management could decide to switch your target platform from Windows desktop PSs to the latest smartwatch technology. Or a strike in the Far East could delay the shipment of your new servers. Or one of your developers might move to Iceland. How can you handle these sorts of completely unpredictable problems?**

One possible solution is to add tasks at the end of schedule. This way, a team gives themselves a grace period if there are any other things to reconsider or develop.

**3.8 What are the two biggest mistakes you can make while tracking tasks?**

The two biggest mistakes are: when nothing is done if tasks fall past their deadline and the other is adding more people to tasks with the assumption that it will speed up the process.

**4.1 List five characteristics of good requirements.**

Productive requirements are easy to understand, unambiguous, consistent, prioritized and verifiable.

**4.3 For this exercise, list the audience-oriented categories for each requirement. Are there requirements in each category? [If not, state why not…]**

A. user, functional

B. user, functional

C. user, functional

D. user, functional

E. non-functional

F. non-functional

G. non-functional

H. non-functional

I. non-functional

J. functional

K. user, functional

L. user, functional

M. user, functional

N. user, functional

O. functional

P. functional

There are zero business requirements. None of them pertained to marketing are economic related requirements.

**4.9** **SKELETON GAME**

**Must**: center the game

**Should**: create a game logo and positioned in the header, FAQ button for help, implement a score tracker, and high scores list to promote competition

**Could**: utilize different colors, implement categories for similar words, difficulty levels

**Won’t**: implement the game in different languages, release the game for modern day consoles like Playstation 4, release the game in a virtual reality environment.