

Owen Hunger, Evan Yu, Connor Savage, Mitchell Cootauco Senior Project2 HW1

You should have read the selections from chapters 1 through 4 of the Stephens text, as specified on the readings page. Having done so will assist you in completing the following assignment.

Here are the problems for easy reference.

Problem 1.1, Stephens page 13

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

The basic tasks in software engineering projects typically include requirements analysis, design, coding, testing, and maintenance.

Problem 1.2, Stephens page 13

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

Requirements Analysis: Identifying and documenting the needs and constraints of the system, ensuring a clear understanding of what the software is expected to accomplish.

Design: Creating a blueprint for the software, defining the architecture, components, modules, data, and interfaces based on the gathered requirements.

Coding: Translating the design into actual code by writing programs in a programming language, following coding standards and guidelines.

Testing: Evaluating the software to identify defects or bugs, ensuring that it meets the specified requirements and functions correctly under various conditions.

Maintenance: Making modifications and improvements to the software after its initial release, addressing issues, adding new features, and adapting to changing requirements over time.

Problem 2.4, Stephens page 27

Like Microsoft Word, Google Docs [sic] provides some simple change tracking tools. Go to <http://www.google.com/docs/about/> to learn more and sign up [if you

do not have an account already]. Then create a document, open the File menu's Version History submenu, select Name Current Version, and name the file 'Version 1'. Make some changes and repeat the preceding steps to name the revised document 'Version 2'. Now open the File menu's Version History submenu again but this time select See Version History. Click the versions listed on the right to see what changed between versions. Document what you've noticed about the information you see, and how the differences between versions are displayed.

Investigation: Compare this process to what you can do with GitHub versions. How are the two tools different? How are they the same?

Google Docs and GitHub both provide version control, but they differ in their implementation. Google Docs seems to allow you to track changes and view version history within the document itself. GitHub, on the other hand, is a platform for version control and collaborative coding, allowing for more complex branching, merging, and collaboration among multiple contributors. Github would be more useful for a development task like a project in this class, while docs would be more useful for tracking who did what.

Problem 2.5, Stephens page 27

What does JBGE stand for and what does it mean?

JPGE stands for just barely good enough, referring to code documentation that is just sufficient enough to understand the code while not being overly detailed. This helps to avoid spending unnecessary time updating documentation when code changes.

Data for Problems 4.2 and 4.4

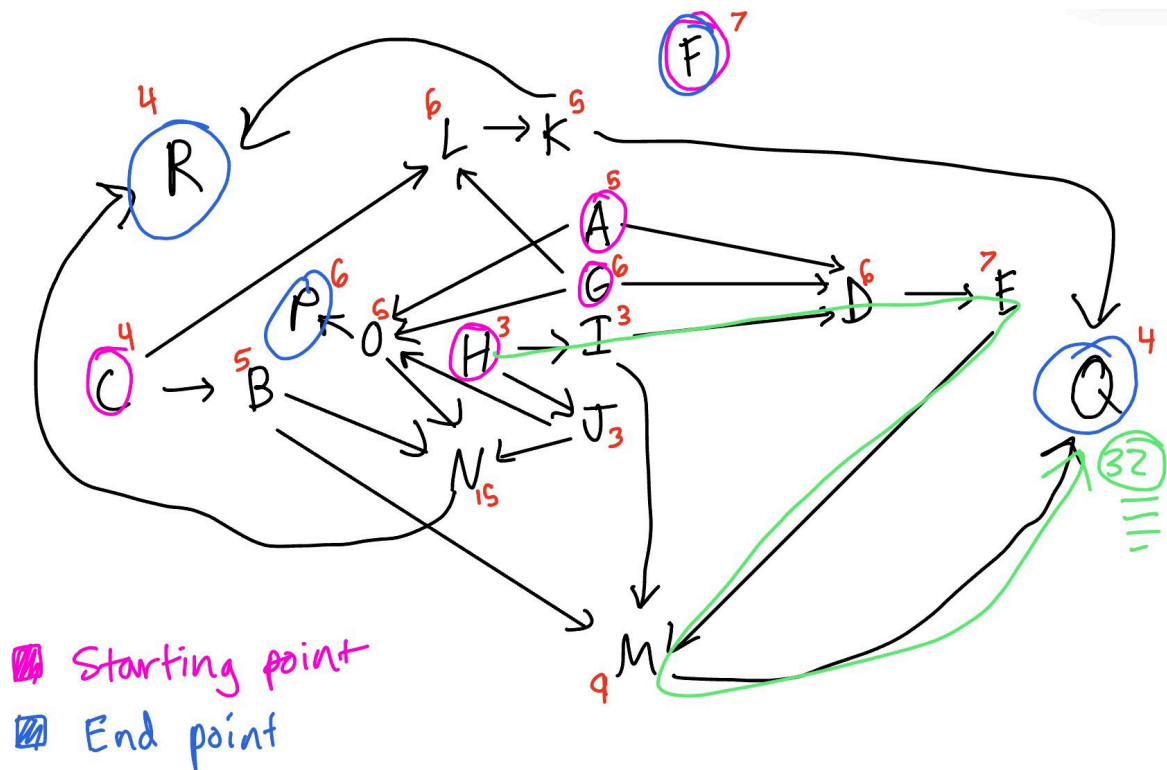
Table 4.2 [below] summarizes some of the classes and modules you might need (and their unreasonably optimistic expected times) to develop players and zombies for the game. (The program would also need lots of other pieces not listed here to handle other parts of the game.)

Use the following table of data for Exercises 4.2 and 4.4.

Task	Time (Days)	Predecessor s
A. Robotic control module	5	—
B. Texture library	5	C
C. Texture editor	4	—
D. Character editor	6	A, G, I
E. Character animator	7	D
F. Artificial intelligence (for zombies)	7	—
G. Rendering engine	6	—
H. Humanoid base classes	3	—
I. Character classes	3	H
J. Zombie classes	3	H
K. Test environment	5	L
L. Test environment editor	6	C, G
M. Character library	9	B, E, I
N. Zombie library	15	B, J, O
O. Zombie editor	5	A, G, J
P. Zombie animator	6	O
Q. Character testing	4	K, M
R. Zombie testing	4	K, N

Problem 4.2, Stephens page 78

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?



The tasks on the path start from H then to I, D, E, M, and finally Q. The total expected duration is 32 days.

Problem 4.4, Stephens page 78

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, 2024, and don't work on weekends or the following holidays:

Holiday	Date
New Year's Day	January 1
Martin Luther King Day	January 20
President's Day	February 17
Alien Overload Appreciation Day	February 18

One of the biggest mistakes you can make is not taking action or accountability once a task slips, you must pay close attention to the task to make sure that it doesn't snowball. The second biggest mistake is piling more people on a task expecting that to cut more time. It may hinder the speed, and also people tend to have specific expertise.

Problem 5.1, Stephens page 114

List five characteristics of good requirements.

Clear, Unambiguous, Prioritized, Consistent, Verifiable

Problem 5.3, Stephens page 114

Suppose you want to build a program called TimeShifter to upload and download files at scheduled times while you're on vacation. The following list shows some of the applications requirements.

- 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 when another is in progress, it waits until the other one finishes.

Nonfunctional

- j. Perform scheduled uploads/downloads.

functional

- k. Keep a log of all attempted uploads/downloads and whether they succeeded.

functional

- l. Let the user empty the log.

Functional, user

- m. Display reports of upload/download attempts.

Functional, user

- n. Let the user view the log reports on a remote device such as a phone.

Functional, user

- o. Send an e-mail to an administrator if an upload/download fails more than its maximum retry number of times.

Functional, user

- p. Send a text message to an administrator if an upload/download fails more than its maximum retry number of times.

Functional, user

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

Problem 5.9, Stephens page 115

Figure 5-1 [right] shows the design for a simple hangman game that will run on smartphones. When you click the New Game button, the program picks a random mystery word from a large list and starts a new game. Then if you click a letter, either the letter is filled in where it appears in the mystery word, or a new piece of Mr. Bones's skeleton appears. In either case, the letter you clicked is grayed out so that you don't pick it again. If you guess all the letters in the mystery word, the game displays a message that says, "Contratulations, you won!" If you build Mr. Bones's complete skeleton, a message says, "Sorry, you lost."

Multiple High scores/Singular high scores- could be modified to track high scores for multiple users or single user high scores.

Multiple Difficulties- The program should allow users to pick a skill or difficulty level. From there, it could have different sets of words for each difficulty.

Brainstorm this application and see if you can think of ways you might change it. Use the MOSCOW method to prioritize your changes.