Finance Buddy
A’Kaia Natasha Temi

Problem 1.1, Stephens page 13

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

- The basic tasks that software engineering projects must handle include determining what and how the software should do. Building the software, removing bugs, making sure the software does what it should, and deploying the finished software.

Problem 1.2, Stephens page 13

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

- Determining what the software should do: Gathering and defining requirements that describe the desired behavior and constraints of the system.
- Determining how the software should do it: Designing the architecture and structure that will satisfy the requirements.
- Building the software: Implementing the design by writing source code.
- Removing bugs: Testing and debugging the software to find and fix errors.
- Making sure the software does what it should: Verifying that the software meets its requirements and behaves correctly.
- Deploying the finished software: Delivering the completed system to users and making it operational.

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?

Comparison with GitHub Versions Similarities: Both tools track changes over time. Both allow users to view earlier versions and compare differences. Both provide a historical record of modifications.

Differences: Google Docs focuses on document-level editing and collaboration, while GitHub focuses on source code management. GitHub tracks changes at a precise line-by-line level and uses commits, branches, and merges. Google Docs is simpler and more visual, while GitHub is more powerful and structureZd for large software projects.

Problem 2.5, Stephens page 27

What does JBGE stand for and what does it mean?

- “just barely good enough”

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)	Predecessors
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

1. Use `critical path methods` to find the total expected time from the project's start for each task's completion.
    a. G → O → N→ R
2. Find the critical path. What are the tasks on the critical path?
    a. Rendering engine → zombie editor → zombie library → zombie testing
3. What is the total expected duration of the project in working days?

a. 32 working days

Problem 4.4, Stephens page 78

Build a Gantt chart for the critical path you drew in Exercise 2. 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
St. Valentine's Day	February 14
Alien Overloard Appreciation Day	March 26
Income Tax Day	April 15

Tasks	Dates
6 days	Jan 2 - jan 9
5 days	Jan 10 - jan 16
15 days	Jan 17 - feb 6
4 days	Feb 7 - feb 12

Finance Buddy
A'Kaia Natasha Temi

Problem 4.6, Stephens page 79

In addition to losing time from vacation and sick leave, projects can suffer from problems that just strike out of nowhere, like a bad version of *deus ex machina*. For example, senior management could decide to switch your target platform from Windows desktop PCs to the latest smartwatch technology. Or a pandemic, hurricane, trade war, earthquake, alien invasion, and so on could delay the shipment of your new servers. [Not that anything as far-fetched as a pandemic might occur, right?] Or one of your developers might move to Iceland, which is a real nice place to raise your kids up. How can you handle these sorts of completely unpredictable problems?

- You handle unpredictable problems by building schedule slack for events lie that, monitoring progress frequently, reassessing priorities quickly, and being prepared to replan when assumptions change.

Problem 4.8, Stephens page 79

According to your textbook, what are the two biggest mistakes you can make while tracking tasks?

- 1. Tracking too many tasks
- 2. Failing to update tasks when there is more information

Problem 5.1, Stephens page 114

List five characteristics of good requirements.

- Clear and unambiguous
- Complete
- Consistent
- Testable
- Necessary

Finance Buddy
A’Kaia Natasha Temi

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.
- b. Let the user specify website log-in parameters such as an Internet address, a port, a username, and a password.
- c. Let the user specify upload/download parameters such a number of retries if there's a problem.
- d. Let the user select an Internet location, a local file, and a time to perform the upload/download.
- e. Let the user schedule uploads/downloads at any time.
- f. Allow uploads/downloads to run at any time.
- g. Make uploads/downloads transfer at least 8 Mbps.
- h. Run uploads/downloads sequentially. Two cannot run at the same time.
- i. If an upload/download is scheduled for a time whan another is in progress, it waits until the other one finishes.
- j. Perform schedule uploads/downloads.
- k. Keep a log of all attempted uploads/downloads and whether the succeeded.
- l. Let the user empty the log.
- m. Display reports of upoad/download attempts.
- n. Let the user view the log reports on a remote device such as a phone.
- o. Send an e-mail to an administrator if an upload/download fails more than its maximum retry number of times.
- p. Send a text message to an administrator if an upload/download fails more than it's maximum retury umber of times.

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

a. User
b. User
c. User
d. User
e. User
f. User

g. System
h. System
i. System
j. System
k. System
l. User
m. User
n. User
o. Administrator
p. Administrator

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."

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

Hangman Game — MoSCoW Prioritization
Must Have

- Fix typo (“Contratulations” → “Congratulations”)
- Prevent repeated letter guesses

Finance Buddy
A’Kaia Natasha Temi

- Display win/lose messages clearly

Should Have

- Difficulty levels
- Hint system
- Sound effects

Could Have

- Themes or skins for Mr. Bones
- Score tracking
- Timed mode

Won’t Have (for now)

- Multiplayer mode Online leaderboards