

Student Name: Krishna Alpeshkumar Patel

Course: Software Project Management & Software Engineering

Journal URL: https://github.com/krispatel1001/soen6481_SPM_weeklyjournal

Dates Range of activities: 9th September 2024 - 16th September 2024

Date of the journal: 21st september 2024

Key Concepts Learned:

Chapter 1: Introduction to Project Management

I now have a strong understanding of what distinguishes a project from routine work or exploratory study thanks to this chapter. I discovered that a project must have clear objectives and a start and finish. It was intriguing to observe how comparable software initiatives are to other kinds of projects, yet having particular difficulties of their own like adaptability, invisibility, and complexity.

Beyond only supervising work, I also gained a greater grasp of the duties of a project manager. It involves managing personnel, balancing resources, and making sure the project continues on course in the face of uncertainty.

Chapter 2: Project Initiation

In this case, the emphasis was on the preliminary actions required to launch a project. It became evident how crucial a project charter and scope are—they provide the foundation for all that comes after.

The idea of SMART goals—specific, measurable, achievable, relevant, and time-bound—was really helpful to me. It's similar to laying down the rules of the game ahead of time and making sure that everyone is aware of what constitutes success.

Early on, we also looked at budget and schedule estimation. Although it might be difficult, forecasting expenses and timelines is essential to starting a project with realistic expectations.

Chapter 3: Effort & Cost Estimation

One of the most difficult aspects of project management was covered in this chapter: calculating the amount of time and money that would be required. I came to see that there isn't a universally applicable solution. Selecting the appropriate approach for the given task is more important.

Methods such as algorithmic cost modelling and experience-based estimate were presented. Experience-based approaches depend on previous work, whereas algorithmic models make

use of mathematical calculations. Each has advantages and disadvantages that vary based on the job.

I had never heard of Function Point Analysis (FPA). It's a means of gauging software functionality from the viewpoint of the user. It appears to be a sensible strategy for organising complicated tasks.

Application in Real Projects:

It seems clear to me how these ideas will be put to use in practical situations. For example, establishing precise project goals and a comprehensive scope can help to avoid a great deal of uncertainty in the future. It's similar to establishing the guidelines for a fruitful endeavour.

The estimating methods will be useful in software development projects for resource planning and budgeting. They offer a methodical approach to managing ambiguity and arriving at well-informed choices.

Peer interactions:

My classmates and I had some excellent conversations on various estimating methods. Hearing about their experiences and potential approaches to different project difficulties was intriguing.

Challenges Faced:

I had some trouble comprehending how estimating models like COCOMO are used in real-world scenarios. Real project data is necessary to fully understand their potential.

Determining project objectives that are both flexible and explicit enough to direct the project presented another difficulty. Striking a good balance is necessary.

Personal Development Activities:

I watched a few Microsoft Project online lessons. I'm excited to use it in future projects because it's a great tool for managing project resources and timeframes.

Goals for the Next Week:

My objectives for the upcoming week are to learn more about the COCOMO model and how it might be applied to contemporary software projects.

In an effort to learn from my friends' varied viewpoints and views, I'll also make an effort to contribute more to class debates.