**SOEN 6841 - SOFTWARE PROJECT MANAGEMENT**

**Student Name:** Vishnu Rameshbabu

**Course:** Master Of Software Engineering

**Journal URL:** <https://github.com/VishnuRameshbabu/SOEN6841>

**Week 4:** Feb 11 – Feb 17

**Date:** 17-02-2024

**Key Concepts Learned:**

I got to learn two different chapters this week, configuration management and project planning. There were many topics like what config management is, its practices, benefits, it’s four key functions, impact analysis reports. And similarly in project planning, its fundamentals, project scheduling, presentation, Charts to visualize like gantt chart, PERT chart and such.   
  
In config management, I got to know why it’s really necessary because changes are often present in a software project because of various reasons by the client. We have to create different variations just in case there is a need to roll back to an older version and in such cases, configuration management is really useful.   
  
There are also the four questions, why, who, what and when changes are made since, these are necessary when there is a need for a config management as it traces back to the change and debates if the change is really necessary.

There are scenarios when a latest version can’t be found, a version is corrupt, features going missing, tested programs don’t work or programmers working on the wrong version and so on. Because of this the project can take a hit and cause delays in delivery and potentially incur expenses at the project’s side.  
  
To reduce all the above problems and provide a stable working environment for the developers then configuration management is necessary. This ensures that all teams authorized, can access the latest or specific versions, continuous integration, auditable and secure.

But to ensure config management is smooth, there is a document that needs to updated whenever a new version is pushed like, project name version, timestamp, doc number, author and type. It’s easy to look up this way.

I also got to learn four main functions of Config management, like identification, change control, status accounting and auditing. Identificaiton helps to identify which version I am exactly working on. Change control to help devs to audit and approve/disapprove changes. A status accounting system to maintain product and config verification records and maintain history of changes. And Auditing to provide a baseline to check if the system developed is approximately same.

Project planning might be one of the most important steps as it defines the trajectory for the project to progress throughout the lifecycle from concept to delivery. There are many components to it, like Schedule planning, effort estimation, Resource planning, quality planning, config mgmt planning and so on. There are also two steps in project scheduling, a top down or bottom up approach where time duration is assigned for all tasks at the start of the project and then assign time for small tasks with their bigger ones. And the vice versa is for bottom up. I have tried the bottom up approach in my professional experience.   
  
Work breakdown Structure is a schematic for breaking down a project into smaller ones and checks for dependencies between each task. Once this is done, resources can be allocated to each task and care must be taken not to overburden any resource or to provide necessary compensation for overtime work.   
  
Always necessary to have a buffer in the project while scheduling as productivity doesn’t equal to number of people on a task because resources being added late to the project can induce comms overheads and make the project delivery late.   
  
Graphical charts can be used to show a project schedule as it can visualize the subtasks that are created. Like bar charts can be used for activities against time, activity charts show task dependencies.

**Reflections on Case Study/course work:**

After reviewing the slides for the this week, I got to learn concepts that can assist in my technical and management role in my professional career. Configuration management in software projects emphasized the critical role of managing change requests and software versions throughout the project lifecycle. Configuration management serves as a foundational process crucial for project success, mitigating chaos and uncertainty. The coursework emphasizes the necessity of robust config mgmt practices to address evolving requirements, technological advancements, funding changes, and scheduling constraints.

Furthermore, the coursework highlights the determential impacts of uncontrolled change, such as schedule delays, quality issues, and legal liabilities, emphasizing the importance of config mgmt. Understanding key components such as configuration identification, control, status accounting, and auditing is vital for maintaining work product integrity and aligning with project requirements and standards. Clear guidelines for documenting, evaluating, and approving change requests are essential, as discussed in the course, to ensure effective config mgmt implementation.

In software project planning, I got to explore various planning technique. This chapter highlights the comprehensive nature of project planning. One significant takeaway was the emphasis on continuous revision and adaptation of plans as new information emerges, reflecting the dynamic nature of software projects.

This chapter teaches the essential elements of project planning, such as project scheduling, budgeting, resource allocation, and quality assurance. It explains the challenges inherent in estimating task durations and managing dependencies, emphasizing the need for careful consideration and contingency planning. Additionally, the discussion on supplier management, configuration management, and communication planning shed light on crucial aspects often overlooked in project planning.

Moreover, the exploration of project planning in iterative software lifecycle models provided valuable insights into adapting planning methodologies to suit different project environments. Understanding how project planning differs in iterative models compared to traditional waterfall models was particularly enlightening, highlighting the need for flexibility and adaptability in project management approaches.

To conclude, both chapters shared the importance of config mgmt. and project planning as it’s necessary for a developer to initially handle the technical aspect and once in progression as a team leader or a project manager, proper project planning is crucial for the success of the whole project.

**Applications in Real time projects:**The insights I gained from this week's class project planning is essential for a project to see success and provides valuable guidance for real-world projects. Let’s take a mobile app project for a bank. In this project, effective project planning is essential to ensure timely delivery and meet stakeholder expectations.   
  
By breaking down the project into manageable tasks using techniques like Work Breakdown Structure, the team can allocate resources efficiently and estimate task durations accurately. For instance, tasks may include requirements gathering, Front end design, backend development, testing, and deployment and maintenance.

Moreover, implementing a robust configuration management system becomes crucial to handle changes effectively throughout the project. Suppose during the development phase, stakeholders request additional features or modifications to existing functionalities and if there is a well-defined configuration management system helps track these change requests, manage different versions of the software, and ensure that the development team works on the latest codebase.

This ensures consistency in the software product and facilitates seamless collaboration among team members. Even though challenges such as estimating task complexities and managing dependencies may arise, implementing these project planning and configuration management practices can significantly benefit the project.   
  
Clear communication channels, formalized processes, and adherence to best practices enhance project predictability, mitigate risks, and ultimately contribute to the successful delivery of the mobile banking app.

**Collaborative Learning:**

For this week I had the opportunity to attend quizzes which refreshed my understanding of the previous week’s concepts.

We got to meet with our designated TA and he specified some valuable points for us to rectify in the project initation, deliverable 1. We made some discussions after the meeting and upon a common agreement some additions were made and submitted.   
  
For the week’s project pitch, we collaborated together to see how we could deliver the pitch in an exemplary way. We thought of adding illustrations about our features on the black board while explaining the concepts.

I would say team collaboration was great for this project pitch and it encouraged us to diversify our searches and stich together the features to deliver the perfect pitch. Infact, our team did great at the pitch and we were confident enough to deliver our pitch to the audience.

**Adjustments to Goals:**

For previous week, I have achieved my goals which was going through chapter 5 and 6 and collaborating with my project mates for the project pitch.

For next week I would have to prepare for the mid term test which is scheduled for the coming week during the lecture hours. I also have to go through the book for the respective chapters and prepare accordingly.