

Student Name: Priyanshu Adhikari(40262789)

Course: SOEN 6841 Software Project Management

Journal URL: <https://github.com/Priyanshu7175/SPM.git>

Week 4: (11 Feb to 17 Feb)

Date: 15 Feb 2024

Key Concepts Learned:

-This week we covered chapter 5 and Chapter 6 which had topics **Configuration Management** and **Project Planning**.

-**Software products** change as per the needs of end users or customers. The changes and many different versions of the project according to the changes introduced in the project. Thus, the management and documentation of the change is configuration management.

We can have changes due to many factors like change in requirements, change or improved version of a software that provides better solutions and budget changes.

- The next question covered was **Why bother with Software Configuration Management?**

Software Configuration Management (SCM) is important because software is easily changeable. SCM helps us handle these changes in an organized way, making sure our software changes can be maintained in an efficient way. It follows non-executable parts (like plans and documents), documentation, executable parts (the actual running software and its code), and the tools we use for making and testing our software.

If we don't use SCM properly, it can lead to problems. For example, not keeping track of different versions of our code can create confusion. If our documents are all over the place, it might lead to misunderstandings. Handling everything manually instead of using SCM tools can cause mistakes, and not keeping track of changes might make fixing problems harder. In short, SCM helps us handle the constant changes in software, making sure everything stays in control and works well.

-The **SCM integration to the project gives a lot of benefits** as it minimizes confusion and establishes order by organizing essential activities. SCM ensures the correct configurations of products, reducing the risk of errors and ensuring legal liabilities are covered by maintaining a

comprehensive record of actions. It also contributes to cost efficiency over the software life cycle, providing a stable working environment and giving compliance to standards.

-Parts of Configuration Management are as follows -:

1. Configuration Identification: Configuration identification involves defining and documenting the changes that are required by the system. These are called as configuration items. So that the change requests for these items can be generated.

2. Configuration Control: Configuration control focuses on managing changes to the configuration item. The changes need to be approved. Change requests can be initiated by individuals or groups responsible for evaluating, accepting, and tracking proposed changes in software products. Criteria for formal change control, change impact analysis, revision history, check-in/check-out procedures, SCCB approval process, linking to the trouble-reporting system, tracking, and resolution of change requests, as well as reviews and regression tests, ensure systematic and controlled software updates.

3. Configuration Status Accounting: Configuration status accounting covers Product description records, configuration verification record, change status and change approval. Overall, It involves the systematic recording and reporting of the configuration items throughout the software development life cycle.

4. Configuration Audits: Configuration audits involve systematic reviews and assessments of the configuration items and gives the differences in the build as compared to the requirement mentioned.

-Project planning is like creating a roadmap for a project. It involves breaking down the work into smaller tasks and figuring out the budget. Components include project scheduling, budgeting, manpower planning, communication planning, and quality planning. There are two ways to plan the schedule: top-down and bottom-up. Top-down starts with an overall timeframe for the entire project, then breaks it down into smaller tasks. Bottom-up begins with the smaller tasks, and their time estimates are added up to create the schedule for the whole project. This planning phase is crucial for guiding the project through its execution, monitoring progress, and staying within budget.

-Project planning components are scope planning, risk planning, schedule planning, effort estimation, cost estimation, communication planning, configuration management planning, tools planning, suppliers planning and quality planning.

-Top Down approach: Top-down project planning begins by examining the whole project at a high level and allocates timeframes for them. It allows for a comprehensive understanding of the project's timeline and structure, facilitating effective organization and management. Top down inputs include Project scope, requirements, service level agreements and project budgets. Top down outputs project duration, defect prevention, communication, configuration managements etc. This approach involves breaking down the project into smaller tasks. The process starts with a broad perspective, gradually moving into more detailed planning for smaller tasks. Overall, top-down planning provides a systematic and structured approach to project management.

-Bottom-up approach: Bottom-up project planning involves starting with individual tasks of a project and estimating the time and resources needed for each one the tasks divided. Each small task is assigned a time duration, and these estimates are then aggregated to determine the overall project timeline and resource requirements. Bottom Up outputs project duration, defect prevention, communication, configuration managements etc. Top down inputs include Project scope, requirements, service level agreements and project budgets. It provides flexibility as adjustments can be made to individual task estimates, accommodating changes as the project progresses. Include Project scope, requirements, service level agreements and project budgets. Bottom-up planning is particularly effective when the specifics of each task are important, allowing for a detailed and realistic project plan that reflects the true nature of the work involved.

-A Work Breakdown Structure (WBS) breaks down the entire project into smaller tasks. It helps understand which tasks come before or after others. Once you have this structure, you can assign people to each task. Think of it like assembling a team for different parts of a project. When deciding how many people to assign to a task, you match the needed skills with what your team has. So, WBS helps organize the work and figure out who does what.

-We have the terms Activity and Milestones. Activity in a project should be organized to produce outputs whereas Milestones are the end point of process activity. Activities examples can be feasibility study, Requirement analysis, Prototype development, Design study and Requirement specification and their corresponding milestones would be feasibility report,

-The **critical path** in a project determines how long the entire project takes. It's found by adding up the times it takes for each step in the project. The critical path is the one that takes the longest time. If anything on this path is delayed, the whole project will be delayed. However, if things outside this main path speed up or slow down a bit, it doesn't change the total project time. So, managing the critical path is crucial for making sure the project stays on track and is completed on time.

-Project planning used in iterative models is different from the waterfall model. For iterative projects, the project span is planned which can be either an iteration or the whole project.

Reflections on Case Study/course work:

-In Chapter 5, a case study highlights a software company in the U.S. that created a system for managing orders, inventories, and logistics. Using an incremental iterative development model, the vendor coordinates with teams at places like India and Russia. The central configuration management system operates 24/7, providing secure access and facilitating collaboration. The testing ensures code compatibility, and developers maintain local builds with the main central system, contributing to an effective development process.

-Project planning should be used in every phase of the iterative projects and helps in planning resources, budgets and other points of the project. The two main approaches to project scheduling are top-down and bottom-up approach.

Collaborative Learning:

-We had a team meeting where the final documentation of the project and the distribution of content was done i.e. which parts need to be covered by whom. A survey of financial literacy app was also initiated so that the market needs could be understood and displayed better in the project.

-A study session where we talked about Configuration Management and some examples like Git, BitBucket and IBM Engineering Workflow Management.

Further Research/Readings:

- Read the book "A Practical Guide to Configuration Management," Suzanne and James Robertson. It gives practical advice on how to organize and control changes in software and project documents. Covers topics like keeping track of different versions, managing changes, and organizing releases. The book uses real-world examples and emphasizes the need for teamwork. Overall, it's a helpful guide for making sure projects stay organized and successful.

Adjustments to Goals

- Survey design for market analysis and final content writing for the project on all the points that have been taken by me.Final project review meet with team and submission of project.
- Prepare for the Pitch next week.