

Learning Journal 3

Student Name: Siya Hasmukhbhai Patel (Student Id: 40321144)

Course: Software Project Management (SOEN 6841)

Journal URL:

Date Range of Activities: February 6,2025 to February 13,2025

Date of the Journal: February 16, 2025

Chapters Covered: Chapter 5 (Configuration Management) and Chapter 6 (Project Planning)

Introduction

This journal reflects on my learning from Chapter 5 (Configuration Management) and Chapter 6 (Project Planning) covered in lectures and presentations. These topics provided critical insights into managing software changes, ensuring version control, and structuring a project for success. Additionally, the lecture emphasized traceability, Git/GitHub integration, auditing, continuous integration, centralized repositories, scheduling techniques like Gantt charts, PERT charts, and Critical Path Method (CPM), effort estimation, resource allocation, and milestone tracking to enhance project execution.

Key Concepts Learned

1. Configuration Management (CM) and Change Control

- **Definition & Purpose:** CM ensures that software modifications are tracked systematically, preventing version conflicts and security issues.
- **Key Functions:**
 - **Configuration Identification:** Defines and organizes project components.
 - **Configuration Control:** Regulates how changes are proposed, reviewed, and implemented.
 - **Status Accounting:** Maintains a record of software evolution and historical changes.
 - **Auditing:** Ensures that changes align with project baselines and requirements.
 - **Continuous Integration:** Allows teams to track and merge updates in real-time for a stable working environment.
- **Real-World Example:** In large-scale projects like Microsoft Windows updates, ineffective CM has led to regressions where fixed bugs reappear due to poor version control.
- **Challenges & Solutions:**
 - **Scope Creep:** Excessive changes disrupt schedules. Solution: Strict change evaluation processes.
 - **Traceability Issues:** Untracked modifications reduce stability. Solution: Automated tools like Git, Azure DevOps, and Jira to link changes with milestones.
 - **Versioning Errors:** Mismatched versions cause lost progress. Solution: Maintain centralized repositories and version-controlled change logs.

2. Project Planning & Scheduling

- **Importance of Planning:** A structured plan ensures a project remains on track by outlining schedules, budgets, and resource allocation.
- **Work Breakdown Structure (WBS):** Divides a project into manageable tasks.
- **Effort Estimation:** Measures the number of person-days or person-months required to complete tasks, ensuring efficient resource allocation.
- **Scheduling Tools:**
 - **Gantt Charts:** A visual representation of tasks over a timeline, showing when tasks start, end, and overlap.
 - **PERT Charts:** A network-based approach emphasizing task dependencies and relationships to monitor project progress.
 - **Critical Path Method (CPM):** Identifies the longest sequence of dependent tasks, determining the minimum project duration.
 - **Critical Chain Method:** Optimizes scheduling by reducing buffer time and ensuring critical tasks are completed efficiently.
- **Milestones & Deliverables:** Milestones act as verification points for project progress, while deliverables mark completed project phases.
- **Real-World Example:** NASA's Mars Rover project used critical path analysis to optimize scheduling and resource allocation, reducing mission delays.
- **Challenges & Solutions:**
 - **Estimating Task Durations:** Poor estimates cause delays. Solution: Use historical data and expert input.
 - **Resource Allocation:** Overburdening team members leads to inefficiencies. Solution: Balance workload using resource leveling techniques.

Application in Real Projects

- **Case Study on Poor CM Practices:** Researching untracked versions in IT projects revealed how security vulnerabilities arise, such as in Apache Struts, where a missing patch led to a massive data breach.
- **Exercise on WBS:** Creating a Work Breakdown Structure (WBS) for a mobile app project reinforced how task structuring improves efficiency.
- **Lecture Insight on Scheduling Techniques:** The professor's discussion on PERT charts and Gantt charts inspired me to integrate these techniques into software project tracking for better visualization.

Unique Takeaway: Smart Parking System with AI

Inspired by the lecture discussion on parking applications, I explored how AI-driven smart parking systems could be integrated into urban planning:

- **Concept:** A real-time parking availability system that uses AI, satellite tracking, and IoT sensors to guide drivers to available spots.
- **CM Relevance:** Using GitHub to link user stories, milestones, and tasks ensures version control, avoiding redundant work and tracking all iterations.
- **Project Planning Application:** The WBS structure could help divide tasks into sensor deployment, software development, and network infrastructure.
- **Real-World Impact:** Reducing congestion in high-traffic areas like downtown business districts by cutting search time for parking.

Challenges Faced & How I Overcame Them

1. **Understanding Configuration Management Tools:** Initially, I struggled with industry tools. Solution: Explored Git branching strategies (e.g., Gitflow model) and how GitHub links user stories to tasks.
2. **Estimating Time in Project Planning:** Difficulties in estimating realistic timelines. Solution: Practiced scheduling through Gantt charts, PERT charts, CPM analysis, and effort estimation techniques.
3. **Applying AI Concepts to Planning:** Integrating AI into traditional project planning was complex. Solution: Studied existing smart city projects and their WBS structures.

Goals for the Next Week

1. **Enhance CM Skills:** Research best practices for linking repositories, tasks, and milestones in GitHub and Azure DevOps.
2. **Improve Scheduling Techniques:** Explore AI-based project management tools like Microsoft Project and Monday.com.
3. **Engage More in Class:** Discuss how traceability, version control, resource allocation, and scheduling tools enhance software project management.

Personal Reflection

Studying Configuration Management and Project Planning has expanded my understanding of how structured approaches improve software and infrastructure projects.

- The AI-based parking system discussion showed me the value of connecting project management to real-world challenges.
- **Long-Term Impact on Career:** These skills will be vital in my future role as a software project manager, ensuring efficient execution and adaptability to change.

Final Reflections

- **Overall Course Impact:** Gained a structured approach to handling project changes, auditing, and planning schedules effectively.
- **Application in Professional Life:** These lessons will help me manage complex software projects with better adaptability.
- **Peer Collaboration Insights:** Class discussions helped me explore new applications of project planning beyond software development.
- **Personal Growth:** This journal reinforced the importance of effort estimation, milestone tracking, structured documentation, and AI applications in project planning.

Conclusion

This Learning Journal 3 has provided a structured reflection on Configuration Management and Project Planning. By exploring real-world applications, Git/GitHub for traceability, scheduling techniques like Gantt and PERT charts, auditing, effort estimation, and structured project planning, I have gained valuable insights that will shape my future career. Moving forward, I aim to deepen my knowledge in risk management and Agile methodologies to prepare for complex real-world challenges.