**Student Name: Yang Cao**

**Course:** SOEN 6841

**Journal URL:** <https://github.com/itshisher/SOEN-6841-learning-journal/tree/main>

**Week 3:** February 18 – March 8

**Date:** March 1

**Key Concepts Learned:**

It is often difficult to control and monitor software projects due to unclear specifications and assumptions made by the project team. This ambiguity leads to difficulties in managing the project work effectively until clarity of the project progresses. Consequently, project monitoring and control become challenging tasks.

Despite these challenges, there are various tools and techniques available to assist project teams in monitoring and controlling software projects. Resource leveling, schedule optimization, and corrective actions against deviations can help overcome setbacks without depleting buffers.

Furthermore, the Earned Value Management (EVM) technique is a valuable tool for taking corrective action and providing a project dashboard with performance indicators. With EVM, project managers can promptly identify any deviations from the planned metrics and take necessary actions to keep the project on track. However, EVM requires that baseline information is available. Overall, while software projects may present monitoring and control challenges, incorporating the right tools and techniques can mitigate risks and ensure successful project progress.

Before finishing the project, there are still many tasks that are left unfinished. If the project didn't go smoothly, there could be lots of loose ends to tie up. But the main things to do before closing are freeing up resources, documenting what we've learned, managing our source code, and saving our project data. It's crucial to keep track of changes in our source code, especially after all the fixing we've done during testing. We also need to decide which version of the software to give to the customer. When it comes to our project data, we need to make sure it's clean and organized before we store it. Archiving it properly is important too, so we can find it easily when needed later.

**Application in Real Projects:**

Software monitoring and control in real projects contains critical activities, including budget management, schedule adherence, quality assurance, risk management, resource allocation, performance measurement, and stakeholder communication. By continuously monitoring project progress, such as project cost, schedule, quality of software, and project risks, project managers can identify deviations from planned objectives and take corrective actions to mitigate issues and keep the project on track. Control mechanisms, such as schedule optimization techniques, risk registers, resource leveling, and Earned Value Management (EVM), provide the necessary framework for effective monitoring and control, ensuring the successful delivery of software projects within scope, time, and budget constraints while meeting stakeholder expectations.

In real projects, the application of project closure is crucial for efficient resource reallocation, documentation of lessons learned, and risk management. By formally closing a project, resources can be released and applied elsewhere, while documentations ensure future projects benefit from past experiences. Overall, project closure ensures successful outcomes, satisfied stakeholders, and valuable lessons for future projects.

**Peer Interactions:**

Meet virtually with teammates to start the project deliverable two. Went over the project phase two scope and requirements. Since there are more staff this time, we planned to split tasks and do some research based on our project topic. After the research we can meet again and exchange ideas.

**Challenges Faced:**

The challenges faced in software monitoring and control are uncertainties in project requirements, scope creep, resource constraints, communication issues, risk management complexities, technology issues, stakeholder expectations, and the need for adaptability to change. These challenges often arise due to unclear specifications, evolving project scopes, limited resources, ineffective communication, and the dynamic nature of software development. Successfully handling these challenges requires proactive techniques such as schedule optimization techniques, risk registers, resource leveling, and Earned Value Management (EVM).

Challenges in project closure include ensuring all deliverables are completed before dure, releasing resources efficiently, effectively documenting project information and transferring knowledge, managing stakeholder satisfaction, addressing remaining risks. Overcoming these challenges requires clear communication, planning, and proactive problem-solving to ensure a smooth and successful closure process that satisfies stakeholders, customers and even development teams.

Challenge for our project: Since we have more requirements for deliverable two, we need to do more searching online to get the project feasibility study, solution proposal, project paln, risk assessment and budgeting documents. We also need to come up with our own experiences based on our topic.

**Personal development activities:**

Readings for chapter 7 and 8. Make summaries for the chapter and read the case study. Online research for our project and discussion with teammates on our project phase two.

**Goals for the Next Week:**

Readings for chapter 9 and 10, exercise for each chapter and review case study. Make a summary for each chapter and go over the lecture slides.

Search online for more information based on our project feasibility study, solution proposal, project paln, risk assessment and budgeting documents. Meet with teammates to initiate our second project with what we have found online.