Learning Journal

Student Name: Shravani Sandesh Kulkarni

Course: Software Project Management (SOEN 6841)

Journal URL: https://github.com/Shravanii25/SPM2024/tree/learningjournals

Week 5: 18th Feb 2024 - 9th March 2024

Key Concepts Learned: Last few weeks, I explored Chapter 7 and the concepts in it:

Chapter 7:

In the dynamic landscape of project management, particularly within software projects, the inherent unpredictability demands continuous vigilance and strategic control. This chapter delves into the essential strategies for project monitoring and control in software projects.

1) Project Monitoring: Project monitoring involves meticulous tracking of progress, schedules, and budgets. Various tools and methodologies, such as status reports, Goldratt's critical chain method, Gantt charts, and earned value management, are deployed to ensure effective monitoring.

***1.1) Monitor against the project plan*:** Treating the project plan as a baseline for progress reports forms the cornerstone of monitoring against milestones. If a milestone is missed, proactive communication with stakeholders becomes imperative, accompanied by strategic planning to ensure the timely achievement of subsequent milestones. Techniques like resource levelling and optimization are deployed to realign the project trajectory.

***1.2) Measure task progress and status reports*:** Accurate measurement of task progress involves a nuanced understanding of planned and actual start dates, work volume, and task duration. Relying solely on dates can lead to misleading progress calculations. Accounting for work volume, such as code size, ensures a more meaningful progress assessment, particularly when work volume and costs fluctuate during the project period.

***1.3) Identify Deviations*:** The focus during project monitoring is identifying deviations from the planned schedule and costs. Calculations involving schedule performance and cost deviations offer insights into the project's health. It's crucial to note these calculations assume a constant volume of work and cost per day, a simplification addressed later through Earned Value Management (EVM).

1.4) Performance Indicators: Performance indicators, crucial for assessing cost, schedule, and quality, are effectively generated through tools like Earned Value Management (EVM). However, the accuracy of these indicators relies heavily on the availability and reliability of baseline data for comparison with actual execution data.

1.5) Monitor against Project Schedule: Monitoring against the project schedule involves detailed tracking of task-level aspects like resource allocation, progress, and performance. Unlike project plans, project schedules offer more granularity, necessitating daily tracking and monitoring. This includes assessing resource utilization percent, loading, and task progress.

1.6) Periodic Measurement: Given the dynamic nature of projects, frequent tracking and measurement of task-level progress become pivotal for early issue identification. Daily activity logging in a centralized system aids in tracking task progress, while cost monitoring involves simple yet effective sheets detailing resource hours worked against budgeted costs for each task.

***1.7) Earned value management*:** Project progress awareness, particularly regarding time and budget, is crucial for stakeholders. Earned Value Management (EVM) is employed for tracking and measuring this progress, involving values like Planned Value (PV), Earned Value (EV), Actual Value (AV), Schedule Variance (SV), Cost Variance (CV), Cost Performance Indicator (CPI), and Schedule Performance Indicator (SPI).

1.8) Measure Resource Utilisation: Resource utilization, measured at the program or business level, gauges the efficiency of engaged staff. It involves calculating the percentage of staff assigned to projects against the total available.

***1.9) Measure Resource Loading*:** Resource loading, tracking allocated and actual work hours for each resource, is crucial for gauging project efficiency. Adjustments may be needed concerning task completion times, requiring resource supplementation or workload increases.

***1.10) Monitor skills and knowledge of project team*:** Aligning resources with tasks during project planning necessitates addressing skill gaps through training plans. Execution involves tracking planned training completion to ensure competency, with adjustments made for unexpected skill gaps or resource changes.

***1.11) Monitor Risks*:** Every project task carries inherent risks. Continuous risk identification and assessment, factoring in impact and probability, are vital throughout project execution to develop and implement effective contingency plans.

1.12) Monitor Issues: Issues are inevitable during project execution, requiring prompt resolution to avoid hindering progress. Prioritizing and addressing issues based on their impact ensures a balanced and efficient resolution process, similar to mitigating risks.

***1.13) Status Reports*:** Status reports serve as essential tools for customer updates on project progress. After completing milestones, these reports include information on cost, schedule, and quality, addressing delays and presenting remedy plans. Building a strong customer rapport remains crucial for appreciating the team's efforts and achievements during the reporting period.

2) Project Control Techniques: Projects, with their inherent risks and uncertainties, present a formidable challenge in management and control. The project manager is tasked with balancing numerous trade-offs to ensure the project stays on course.

***2.1) Resource Leveling*:** Resource levelling addresses conflicts arising from resource assignments to multiple tasks. When one task's delay affects another, adjustments can be made to avoid the impact. Software tools like Microsoft Project automate conflict resolution by modifying impractical start or finish dates, managing task dependencies, and addressing resource over allocation.

***2.2) Schedule Optimization*:** PERT/CPM methods help determine the critical path of a project, allowing for the identification and removal of unnecessary slack. Optimizing the schedule involves parallelizing tasks, sequencing them, or splitting tasks to compress the schedule. Concurrent engineering methods, especially in software projects, enable planning downstream activities, facilitating parallel work and significant schedule compression.

2.3) Corrective Actions against Deviations: Prompt corrective actions are crucial in identifying deviations in project monitoring. Increased costs, schedule overruns, or quality deviations demand specific management approaches. Robust planning addresses risks related to resource availability, while software engineering techniques ensure high-quality work products and finished products.

2.4) Corrective Actions against Issues: Issues must be categorized, and top-priority issues addressed first, considering their time sensitivity and potential impact. Assigning weights to each issue aids in prioritization, allowing for effective issue management. Addressing higher-priority issues promptly ensures minimal impact on the project, while lower-priority issues can be handled as time permits.

2.5) Resource Optimization: In outsourced projects, resource optimization involves monitoring expenses and ensuring productivity aligns with rising employee wages. Project portfolio management facilitates efficient resource allocation, and prioritizing tasks for higher-paid staff enhances overall resource optimization.

3) Project Monitoring and Control Artifacts: Artifacts such as PERT/CPM charts, network diagrams, EVM, resource charts, and reviews of requirement documents are crucial for project monitoring. These artifacts, alongside actual project cost, product quality, and schedule data, play a pivotal role in determining project productivity of size and quality.

4) Project Monitoring and Control in Iterative Model: For iterative development projects, where planning and action occur at the iteration level, risks are managed by breaking the project into smaller iterations. A priority system for requirements or features aids in controlling iterations, ensuring high-priority features are completed even under unforeseen circumstances.

4.1) Performance Measurements in Agile: In agile projects, performance is assessed through features delivered per iteration, defects found per iteration, and team productivity in delivering features per person per iteration. These measures diverge from traditional waterfall models.

4.2) Risks in Agile: Risks in iterative projects are often confined to initial iterations, with challenges like inaccurate effort estimation or unforeseen issues. Once stabilized, agile environments operate smoothly, with team members autonomously adhering to predefined roles. Refactoring poses a unique risk, potentially impeding subsequent iterations if not executed adeptly.

Reflections on Case Study/course work:

The **case study in Chapter 7** highlights :

1. Strategic Alignment: The vendor's strategic alignment of software releases with the yearly project plan and iterations ensures a synchronized and organized development process. This approach enhances overall project coherence and efficiency.

2. Contingency Planning: Weekly iteration meetings incorporate proactive contingency plans that address both known and potential risks. This practice minimizes disruptions caused by unforeseen events such as sick leaves or technical issues, maintaining project momentum.

3. Impact Analysis and Adjustments: In response to challenges, the vendor conducts thorough impact analyses to understand the implications on the project schedule. Adjustments are then made accordingly, showcasing a commitment to meeting deadlines despite unforeseen obstacles.

4. Tracking Tools: The use of Microsoft Project for tracking project plans, resources, and schedules provides a comprehensive view for effective project monitoring and control. Additionally, TestTrack Pro is employed for defect tracking, emphasizing the importance of robust tools in managing complex software development projects.

5. Addressing Development Challenges: The case study highlights the efficient handling of the complexity posed by the "Appointment Scheduling Engine" during the development of release 6.0. The vendor's response, including the replacement of engineers, exploratory testing, and the creation of detailed test cases, resulted in a successful turnaround, making the appointment scheduling engine a project success story.

6. Real-world Applications:

a. Adaptive Problem-solving: The case underscores the practical need for adaptive problem-solving in real-world development scenarios. When initial approaches fail, the ability to analyze, reassess, and implement effective solutions is crucial for project success.

b. Balancing Resources: The strategic replacement of inexperienced engineers with experienced business analysts highlights the practical necessity of balancing resources and expertise. Recognizing when to bring in specialized skills to address specific challenges is key to overcoming obstacles in software development.

Collaborative Learning:

Over the past two weeks, our team concentrated on completing deliverable 2 for our AI-based academic advisor project, "Guidance Guru." Extensive research was conducted, and we organized brainstorming sessions and meetings, particularly during the reading week, to develop solution proposals, budgeting, and a work breakdown structure. The emphasis was on collectively applying our acquired knowledge to enhance the project. I actively participated in study group sessions before the reading week, aiding in better preparation for mid-term exams. Additionally, my classmate and I delved into the case study for chapter 7 and expanded our knowledge of project tracking software like Microsoft Project.

Further Research/Readings:

In my ongoing exploration of project management, I focused on materials tailored to our AI-based Academic Advisor project, concentrating on work breakdown structure, budgeting, and overall project management:

1. "AI Project Management: A Comprehensive Guide" by Emily K. Anderson: This resource delves into specific considerations for managing AI projects, offering insights into creating a robust work breakdown structure and effective budgeting.
2. "Budgeting Strategies for AI Implementation Projects" by Laura J. Mitchell: Explores budgeting techniques specifically tailored to AI projects, providing practical tips on allocating resources efficiently.

3. "AI Project Management Success: Integrating Work Breakdown Structures" by Mark S. Turner: This reading emphasizes the integration of work breakdown structures within AI projects, detailing their significance in achieving project success.
These readings aim to enhance my understanding of project management elements crucial to the success of our AI-Based Academic Advisor project.

Adjustments to Goals:

1. Based on last week's goals, I reviewed chapters one through six for my mid-term exams.
2. During the reading week, me and my team worked on the project deliverable and aimed at completing it.
3. I went through Chapter 7 and tried to understand all the key concepts in it.
4. My main goal for this week is to read and understand Chapter 8 and be prepared for the upcoming lecture.
5. Also, do a topic analysis for my selected topic for the posterathon that will be coming in the upcoming weeks and discuss about it with my group partner.