Learning Journal 3

Student Name: Janmitsinh Panjrolia

Course: SOEN 6841

Journal URL: GitHub

Date of the Journal: 23rd February

Dates Rage of activities: 9th February to 23rd February

Key Concepts Learned

Software project planning:

- **Project scheduling techniques**: Top-down (assigning time to larger tasks first) vs. bottom-up (summing smaller task durations).
- **Work Breakdown Structure (WBS)**: Breaking projects into manageable tasks (e.g., frontend, backend, testing) and defining dependencies.
- **Resource allocation:** Matching skills to tasks (e.g., React developers for frontend, Node.js experts for APIs).
- **Critical Path Method (CPM) and Goldratt's critical chain method:** Identifying task sequences that dictate project timelines.
- **Milestones/deliverables:** Setting checkpoints (e.g., API completion) and outputs (e.g., requirements document).
- **Communication and quality planning:** Ensuring alignment with stakeholders and defining quality benchmarks.

Challenging Component: Understanding how to balance top-down and bottom-up planning in iterative models, especially when client requirements change mid-project.

Application in Real Projects

In my full-stack e-commerce project, I applied these concepts:

- **WBS Creation:** Divided the project into tasks like user authentication (backend), product catalog (frontend), and payment gateway integration.
- **Top-down scheduling:** Allocated 3 weeks for front-end development, then subdivided into smaller tasks (e.g., 2 days for UI design).
- **Critical Path Analysis:** Identified that database schema finalization was critical before API development could begin.
- **Resource Allocation:** Assigned team members based on expertise (e.g., a DevOps engineer for deployment pipelines).

Challenging Component: Proposed using agile-CPM hybrid for flexibility re-evaluating the critical path after each sprint to accommodate changes like adding a new search feature.

Peer Interactions/Collaboration

- Collaborated with peers to refine task dependencies. For example, a teammate highlighted that the shopping cart UI couldn't start without finalized API endpoints, prompting us to prioritize backend tasks.
- Received feedback on our WBS, which initially overlooked testing

milestones. We added "Unit Testing Completion" as a milestone.

Challenging Component: A peer's suggestion to use parallel tasking (e.g., frontend prototyping while APIs were developed) reduced delays by 20%. This taught me to leverage overlapping dependencies creatively.

Challenges Faced

- I initially struggled to identify the critical path in my full-stack project's activity network. For example, I assumed backend authentication (Task A) and frontend login UI (Task B) could run in parallel. However, after revisiting the chapter's explanation of hard vs. soft dependencies, I realized Task B couldn't start until Task A's API endpoints were finalized (a hard dependency). This misalignment delayed prototyping by 3 days.
- The chapter's emphasis on removing unnecessary buffers (Goldratt's critical chain) conflicted with my team's agile approach, where sprint buffers are common. I initially eliminated all contingency time, leading to burnout when third-party payment gateway integration faced unexpected delays. After re-reading the section, I understood that strategic buffers should be placed only after critical chain tasks, not removed entirely. I added a 10% buffer post-critical-path tasks, improving team resilience without compromising efficiency.

Personal Development Activities

- I completed a Jira certification to improve task tracking.
- Attended a webinar on agile project management to better handle scope changes.
- Practiced creating Gantt charts in MS Project for clearer visual scheduling.

Goals for the Next Week

1. Master risk-adjusted backlog planning to preempt delays in the next sprint.
2. Implement a peer review system for task estimates to improve accuracy.
3. Explore AI tools like Forecast. App for automated resource allocation.

Challenging Component: Align these goals with long-term career growth in technical leadership, aiming to lead cross-functional teams efficiently.