Learning Journal 2

Student Name: Hanieh Maleki

Course: Software Project Management

Journal URL: https://github.com/HaniMLK/Soen-6841-/wiki/Learning-Journal-2

Dates Rage of activities: 26 Jan - 9 Feb

Date of the journal: 9 Feb

Key Concepts Learned

This week, I explored two topics in project management: Risk Management (Chapter 4) and Configuration Management (Chapter 5). I studied both the slides and the book chapters along surfing the internet to find more information.

Risk Management:

One of the biggest takeaways was that risks in a project don't just arise from technology failures but can come from multiple areas. Risks can be categorized into several types, each requiring a different strategy to manage:

1. **Technological Risks** – These include outdated software, compatibility issues, and security vulnerabilities.
   Mitigation Strategy: To reduce the probability of these risks, we can adopt stable and well-documented technologies, conduct frequent testing, and maintain a backup system.

2. **Quality Risks**: occur when the delivered product does not meet requirements due to poor design, lack of testing, or rushed development. This can lead to costly rework and customer dissatisfaction.

   Mitigation Strategy: Quality Assurance (QA) & Iterative Testing – Implementing continuous testing (CI/CD pipelines), peer reviews, and quality gates at each development stage ensures high standards. Conducting pilot tests before full deployment helps identify and resolve issues early.

3. **Estimation Risks** – These happen when time, budget, or effort is underestimated. For example, if we assume our platform can be developed in six months but later realize it needs nine, it could delay the entire project.
   Mitigation Strategy: Using historical data, expert judgment, and a buffer in time and cost estimates can help prevent underestimation. Historical data for a project like our platform__ MindSafe __is not doable but the two other options sound perfect. If a team has enough budget to survive and pass Death Valley, it's good idea to consider a buffer time.

4. **Resource Risks** – If key team members leave, or there aren't enough skilled developers, the project could slow down or stop.

Avoidance Strategy**:** Hiring additional team members or training existing staff early on can help prevent resource shortages.

5. **Budget Risks**: These occur when the project runs over budget due to poor estimation, unforeseen expenses, or resource misallocation. If not managed well, the project could face financial instability.
   Mitigation Strategy: Contingency Planning – Allocating a buffer budget (typically said that 10-20% of the estimated cost) helps absorb unexpected expenses. Additionally, using cost estimation models and regularly reviewing financial progress ensures early detection of budget overruns.

6. **Schedule Risks** – If tasks take longer than expected, the project might not meet deadlines. This is common when requirements keep changing.
   Mitigation Strategy: Implementing agile project management and breaking work into smaller milestones allows flexibility while keeping track of progress.

Each of these risks presents different challenges, but the right strategy can minimize their impact or prevent them altogether. Moreover, there are other risks including Organizational Risks (company policies, team dynamics...), Legal & Compliance Risks and Estimation Risks

Configuration management

It showed me how important it is to track changes in software projects. Without proper configuration, we might lose track of critical versions, which could lead to chaos, such as running the wrong code or missing critical bug fixes. This made me appreciate the importance of having a structured version control system in any project. So, during this chapter interestingly I found that some of tools that I used to organise my previous works or project are among them. When I started working with my startup team (I had MatiKan and GreenLink), we needed a way to stay organized, track progress, and manage changes efficiently. Initially, we tried using spreadsheets and WhatsApp messages, but that quickly became chaotic. That's when we decided to use **Trello**—at first, just to track tasks, but now, I realized that it was actually functioning as a lightweight Configuration Management (CM) system for us. I also used Github but Trello was more convenient for me. Even in Word documents I used “track changes” a lot.
Another interesting aspect I learned is that Version Control is More Than Just Code! Even tasks, documents, and bug reports need to be tracked carefully. I also learned CM Doesn’t Require Complex Tools just like Trell or Github. One more thing through search I learned is that connecting Trello with GitHub can make most benefits of both. I’ve never thought of integrating simple CM tools.

Application in Real Projects

I applied some of these concepts while working with my teammates on our pitch presentation for our project, the Crowdsourced Mental Health Data Platform (MindSafe). Since we were brainstorming and aligning ideas during our 7-hour group meeting, I realized we were essentially practicing informal risk management. We discussed risks related to effort and cost estimation and resource allocation, particularly the challenge of calculating TAM (Total Addressable Market) and SAM (Serviceable Available Market) accurately. Nevertheless, we noticed this process wasn’t as straightforward as estimating lines of code or developer effort—something we discussed in our software engineering course. This hands-on experience made me more aware of how theory and practice come together in project management.

Peer Interactions

Collaboration was a major highlight of the week. Although I've had many speaking and teaching opportunities in the past, I chose not to present this time. Instead, I focused on creating the presentation file to help my teammates practice and present confidently. During our long brainstorming session, we reached consensus on the presentation's key points and also discussed budget breakdowns and estimation. Everyone contributed their perspectives, and it felt like a true team effort where every voice mattered. I was particularly excited to see how we applied shared learning from previous courses to estimate project timelines and deliverables. However, we didn't discuss the course materials fully or in details. We only picked usage of some needed for the Tuesday pitch.

Challenges Faced

One challenge we faced was aligning our cost estimates with real-world considerations. Since we're still in the idea evaluation stages, we don't know exactly how many lines of code we'll need or how complex the implementation will be. This made it tricky to calculate effort estimation accurately. We had to rely on assumptions based on typical software projects, but this exercise made us think critically about potential resource gaps and risks. Another thing that I don't consider a challenge but discussed was stakeholders, we genuinely consider them into two groups but may change some details as the project goes further. Finally, I think I'm good in theory and learning materials but I feel I need to spend more time on real math and calculation of the things I learned, sometimes on paper things are reasonable and fine but when we want to apply we'll see the inefficiencies these are occupying my mind.

Personal development activities

This week, I focused on strengthening my analytical and research skills. I calculated TAM and SAM for our project since I had experience with it previously, however, we didn't need it actually for this pitch because it is different from an investor pitch, but I talked about this extra topic during the meeting which was a good learning practice for all. Additionally, by stepping back from the spotlight and supporting my teammates from behind the scenes, I practiced humility and learned the importance of enabling others to succeed. It's a reminder that leadership isn't always about being front and center.

Goals for the Next Week:

Next week, I plan to:

1. Refine our project's risk assessment by categorizing risks more thoroughly (technical, operational, financial) and developing a clearer mitigation plan. I still need to get hands-on experience to fully understand the subject and details. I will search for best practices and try to bring the topic to our group this week.
2. Collaborate with the team to create a more detailed Work Breakdown Structure (WBS) for our software development timeline.
3. Continue learning about effort estimation techniques, particularly in relation to software coding, to improve our project's accuracy in resource allocation.