**Learning Journal**

**Student Name:** Sannidhi Shetty

**Course:** SOEN-6841 Software Project Management

**Journal URL:** [**https://github.com/sannidhi470/SPM**](https://github.com/sannidhi470/SPM)

**Week 3:** Feb 4 – Feb 10

**Date:** Feb 7

**Key Concepts Learned:**

Below are the concepts I learnt from this week’s session:

**CHAPTER 4: Risk Management**

* Risk in software project management refers to any situation that could have a negative impact on the project's objectives, such as cost, schedule, quality, or scope. These risks can arise from various sources, including technical challenges, resource constraints, changing requirements, and external factors. So, it becomes very important to manage risks in software projects.
* Managing risks involves identifying, analyzing, and prioritizing them, as well as developing strategies to mitigate or address the risks. Risk management is an ongoing process throughout the project lifecycle, from planning and execution to monitoring and control.
* Key aspects of risk management include:
* **Risk Identification**: Identifying and documenting potential risks that could have an impact to the project. This step is about making a list of all the things that could potentially cause trouble during the project. It's like brainstorming for all the possible problems.
* **Risk Analysis:** Assessing the likelihood and potential impact of each risk on project objectives. Once we have the list, we need to figure out how likely each problem is to happen and how much damage it could cause. This helps us understand which problems are the most serious.
* **Risk Prioritization:** Prioritizing risks based on their significance and determining which ones require immediate attention. With the analysis done, we can decide which risk need immediate attention so that it gets mitigated and does not impact important part of the project.

**Types of risks:**

* Resource Risks:

These risks involve not having enough people, materials, or equipment to complete the project. For example, if valuable team members leave unexpectedly or tools are unavailable whenever necessary.

* Technology Risks:

These risks relates to problems with the technology stack being used in the project. This could include issues like software bugs, hardware failures, or compatibility issues with different systems.

* Budget Risks:

Budget risks occur when there are uncertainties or unexpected expenses that could impact the estimated project budget. This might include underestimating costs, unexpected price change or increase.

* Quality Risks:

Quality risks involve the possibility of delivering a product that doesn't meet the expected standards. This could be due to errors in development, lack of production testing. This leads to delay in delivery of product to users.

* Time Risks:

Time risks is related to delays or schedule adjustments that could affect the project delivery time. This can include unexpected setbacks, dependencies on other projects, or poor time management.

* People Risks:

People risks are associated with issues within the project team, such as conflicts among team members, skill shortages, or lack of communication.

* Organizational Risks:

These risks originate from factors within the organization, such as changes in leadership, restructuring, or shifting priorities, which could impact the project's progress and success.

* Requirements Risks:

Requirements risks involve uncertainties or continuous changes in project requirements that could affect the final deliverable of software product. This might include misunderstandings with client, evolving needs, or incomplete specifications at the beginning of development.

* Tools Risks:

Tools risks refer to problems with the software or equipment used in the project. This could include issues like software limitations, outdated tools, or lack of necessary technology infrastructure.

* Estimation Risks:

Estimation risks arise from inaccuracies or uncertainties in project. This could involve underestimating project scope by overestimating team’s capacity, or not analyzing complexities in the software solution at the initial phases of implementation.

**Risk Control**

* **Risk Planning**

Risk planning is about thinking the potential problems of the software solution beforehand and to prepare strategies to deal with it. It's like making a game plan for how to handle problems if they arise.

Strategies to manage risks:

* **Acceptance**

Acceptance means acknowledging the risk by deciding not to take any action to address it.

* **Avoidance**

Avoidance involves taking steps to eliminate the risk entirely or reduce its likelihood of occurring.

* **Risk Transfer**

Risk transfer involves shifting the responsibility for the risk to another party, such as hiring an expert who would be able to mitigate risk effectively. It's like passing the risk on to someone else who is better equipped to handle it.

* **Mitigation**

Mitigation involves taking actions to reduce the impact or severity of the risk if it occurs. It's like putting techniques in place to lessen the damage or consequences of the risk on the project.

Risk Mitigation can be done through:

* **Schedule buffer on projects**

Adding extra time to the project schedule for potential delays or setbacks.

* **Knowledge Management System**

Implementing a system to capture, share, and distribute knowledge and expertise within the team.

* **Quality gates on projects**

Setting checkpoints or milestones throughout the project to ensure that quality standards are met before proceeding to the next phase.

* **Risk Prioritization**

Identifying and focusing on the most critical risks first so that it gets resolved first and then the lower priority risks. This will minimize the overall risk of the project.

* **Risk Resolution**

Resolution means assigning someone to take care of a specific risk and setting a deadline for when it should be dealt with. It's like giving someone the responsibility to fix a problem by a certain date.

* **Risk Monitoring**

This involves keeping an eye on the project throughout its development to identify any new risks or changes in existing risks. It helps to ensure everything is going smoothly in order to catch any issues early on.

**Causes of risks on projects:**

* Bad negotiation: Making unfair deals or agreements during project discussions with client can lead to problems later on.
* Cost constraints : Not having enough money to do what's needed for the project can cause difficulties and risks.
* Quality constraints: Having to rush or ignoring important functionality can affect quality that can lead to mistakes and risks.
* Disinterest: When people aren't interested or motivated in the project, they might not put in their best effort, causing problems.
* Resource unavailability: Not having the right people or tools available when needed can slow down the project and create risks.
* Attrition: When team members leave the project unexpectedly, it can impact progress and cause problems.
* Scope creep: Adding too many extra tasks or changes to the project's original plan can make it harder to finish on time and within the estimated budget.
* Poor management: Not having proper management or leadership can lead to confusion and mistakes.
* Human error: People making mistakes or forgetting things can lead to problems in the project.
* Unrealistic estimate: Making assumptions about the project that are too optimistic or not based on reality can lead to problems later on.

**Risks due to large number of Requirements**

* **Long gestation period:** Having a lot of requirements can make the project take a long time to start or complete which can lead to delay in delivery time.
* **Large upfront commitment:** Needing to commit a lot of resources or effort upfront before knowing if the project will succeed can be risky.
* **High management costs:** Managing a large number of requirements can be complex and expensive, increasing project management costs.
* **Requirement changes:** With many requirements, there's a higher chance of changes or updates, which can impact the progress and cause additional work.
* **Miscommunication:** More requirements mean more chances for misunderstandings or miscommunication, leading to mistakes or conflicts in the project.

**Reflections on Case Study/course work:**

* With these learnings on chapter 4 (risk management), I see myself to apply these concepts when analyzing risks at the initial phases of real-world projects. If I consider my assigned project (Financial Literacy App) as an example, identification of potential risks such as technical glitches and low user engagement is crucial, followed by thorough analysis and prioritization based on their impact. Risk response plans, including educational content updates and customer support channels, are essential for addressing evolving user needs. Mitigation strategies such as scheduling buffers and quality assurance processes can be used by continuously monitoring the project.

Some of the risks that can arise in financial risk apps are:

* **Resource Risks:**

Losing key team members with specialized financial expertise or experiencing shortages in necessary educational materials for the app's content development can cause problems.

* **Technology Risks:**

Encountering software errors or compatibility issues with different mobile device sizes can lead to user frustration or inability to access app features.

* **Budget Risks:**

Underestimating the costs associated with app development, marketing, or maintenance will result in budget overruns.

* **Quality Risks:**

Delivering inaccurate financial information or educational content due to errors in development or lack of thorough testing will result in poor user satisfaction.

**Time Risks:**

Experiencing delays in app development due to unforeseen technical challenges or dependencies will impact the project timeline and user expectations.

* The case study 4 focuses on the risks encountered by a SaaS vendor while developing its software product. Key risks included managing offshore teams, attrition, communication gaps, development costs, unmanageable schedule, and software quality. To mitigate these risks, the company implemented strategies such as standardized communication templates, virtual meetings, schedule buffers, prioritization of features, overtime work allocation, and thorough quality checks at each stage of development. These measures helped the company mitigate risks effectively and ensure the success of the project.

**Collaborative Learning:**

* We are having regular project team meetings to touch base on the assigned topic “Financial Literacy App”. We are documenting market analysis and project initiation for the project. Each of us has taken 2 existing financial literacy apps in the market to determine their features and problems to perform market analysis. I’ve been assigned Robinhood and Credit Karma. We have finalized an opportunity on the domain around which we will be proposing our software solution for the project.
* Had a study session with a classmate to go through previous chapters and to discuss Chapter 4 and how we were involved in mitigating risks on our software projects. For example, in one project, I faced challenges with team coordination and communication due to remote work arrangements. To mitigate this risk, our project manager implemented regular virtual meetings using tools like Zoom and Teams to ensure everyone remains updated and aligned with project objective. Additionally, we used to prioritize tasks so that when risk occurs, important tasks would be completed. Like the strategy mentioned in the case study. By exchanging our experiences and discussing practical application, we gained insights into effective risk management techniques that we could apply in future projects.

**Further Research/Readings:**

* I went through the book "Effective Risk Management: Some Keys to Success" by Edmund H. Conrow to understand risk management better. It is a helpful book that teaches us how to deal with uncertainty and potential problems in our projects. Conrow explains that risk management involves four main steps: first, identifying what could go wrong, then figuring out how frequently it can occur and how bad it could be. After that, we come up with ways to prevent the impact of those risks. The book also gives practical tips, like involving everyone in the organization and using different ways to understand and handle risks. Overall, it's a useful guide for anyone who wants to learn how to mitigate risks effectively. One unique concept covered in "Effective Risk Management: Some Keys to Success" by Edmund H. Conrow is the emphasis on integrating both qualitative and quantitative approaches in risk assessment. Conrow highlights the importance of not only considering numerical data but also qualitative factors such as expert judgment, historical analysis, and stakeholder insights when evaluating risks.

**Adjustments to Goals:**

* The goal for last week was to start with the first deliverable of the project. We have been working and documenting it throughout this week with regular team meetings. Each of us are assigned 2 existing financial literacy applications to perform market analysis. Compared to last week where we had shortlisted few opportunities in the app, we have finalized a problem/opportunity around which we will be delivering our software solution for the project.
* After attending the class for chapter 4 and reading extra materials, I have gained knowledge on how to manage various types of risks in software projects. This will help me analyze and identify risks at the initial phases by avoiding any potential issues in the later stages of development.
* Goal for next week is to continue with the project analysis, prepare for midterms by going through the covered chapters in class and read extra materials on the topics to enhance my practical knowledge on the concepts.