Learning Journal Week 3

Student Name: Jasmanpreet Kaur Bedi

Course: SOEN-6841: Software Project Management

Journal URL: https://github.com/jasmanpreet0209/Software-Project-Management-Learning-Journals

Week 3: February 4 – February 10

Date: February 9

Key Concepts Learned:

This week started by revising the concepts about Risks, Risk Analysis and Risk Mitigation strategies as studied in chapter 4. Moving further I learned about the concepts in Chapter 5:

1. **Configuration Management:**

   Configuration Management is a crucial Process in the field of Product Development, which involves handling changes to the numerous artifacts generated throughout the process of development from project initiation to closure. Configuration Management helps in handling artifacts systematically and storing these artifacts, such as documents, requirements, code, or any other relevant document, ensuring that they can be easily accessed and retrieved when needed. This is done to manage versions, accommodate changing requirements and prevent team members from working on incorrect versions.

2. **Need For Configuration Management:**

   Configuration Management is needed to maintain the integrity of a product. Without effective configuration management, there is a risk of version control issues, leading to potential errors and inefficiencies in the development process. It is required to maintain consistency across various teams and artifacts. CM also provides a platform for documentation.

3. **Centralized vs Decentralized Configuration Management:**

   Centralized Configuration Management is essential for the smooth execution of software projects involving teams distributed across various locations and time zones. In software projects, it is important for teams working in different places and time zones to use a centralized configuration system as if there is a decentralized system, i.e. each team has its own way of managing things, it can lead to problems. For example, the same item might have different names in each team's system, one team may be working on a different version of the same item causing confusion and problems later on for integration. A centralized system helps avoid these issues, ensuring that everyone works together smoothly with the same set of rules and organization.

4. **Best Practices for Configuration Management:**

   The following practices can make configuration management more effective and organized:

- **Centralized System:** Using one system where everything is organized helps teams in managing their work, maintain consistency, integrity, and ease of integration.
- **Secure Access:** Making sure the system has a safe way for people to access it and Using roles to control who can do what, keeping things secure.
- **Continuous Integration:** Keeping the software build process regular, continuous and including a "smoke test" to catch any big issues quickly so that they don't cause issues later in development.
- **Easy Branching:** Having a simple way to create branches for different versions of the software helps in managing different versions easily and store everything in one main repository.
- **Audit Facility:** This means including a feature to keep track of changes and actions. This way, one can review and understand what happened at different times.

5. **What is Smoke Testing**

   Smoke testing is a quick check done to ensure that the critical functions of a software application work properly even after making changes. This helps in identifying major issues early on in the development process.

6. **Version Control**

   Version control provides a systematic way of organizing and storing different versions of a project at one place. Teams can work on their respective tasks in parallel, and the system keeps a track of every change made. There is one Main repository and there can be as many branches as needed so that there is consistency across the teams and one repository to store what is the final version of the project. This not only facilitates collaboration but also serves as a safety net, allowing teams to revert to previous versions if issues arise.

Reflections on Case Study/course work:

The case study presents the software vendor with a system that helped businesses manage orders and inventory. Incremental iteration development method is used. The company had teams working in the U.S. and other countries like India and Russia. They go for outsourcing parts of development to reduce the development cost. To keep everything organized, they use a special configuration management system which was available all the time, and it made sure everyone, no matter where they were, could work smoothly without any security risks.

The system has different access levels: some people could edit and manage documents, while others could only view them. They also had a super-user role for more advanced tasks. The main branch of the system had the main version of the software with all the important updates.

They used smoke testing to quickly check if new code worked well with the existing system. If there were any issues related to the build process or any inconsistency, it alerted the concerned people. Each developer also had a local copy of the system, and they tested their code there before adding it to the main system. This helped make sure the main system was free of problems and that the central build did not fail. This way of working allowed them to fix any issues quickly and kept the development process running smoothly, even with teams in different parts of the world.

Overall, the case study explains the importance of a centralized configuration management system and all the concepts learned in the lecture and from the book. The access rights explain the importance of secured access and role-based access control. It also highlights the concepts of Version Control and maintaining a main central repository as well as Importance of Continuous Integration.

Application in Real Projects:

The concepts learned resonate with real life development process and practices. In real world projects, Configuration management plays the most important role in maintaining order in chaos, control over project artifacts and teams like code, documentation, versions. Configuration Management helps a project stay organized, well-managed, maintains a software's integrity no matter how large the number of teams is. Continuous integration with smoke testing reduces the risk of bugs/failures in the software.

Peer Interactions/Collaborations:

This week, I collaborated with my peers to dive deeper and discuss in detail about the research findings on market analysis of the Educational Gamification Problem. We met once on Thursday, to discuss the changes as suggested by the professor and constantly kept in touch otherwise as well. We maintained a Github Repository to log the tasks each of us was working on and the progress of the project. We defined a Continuous Integration approach and implemented Change management.

Challenges Faced:

I faced challenges researching and finalizing about some aspects for example, while doing market analysis I came across a lot of companies that had a similar domain and it was challenging, narrowing the companies down by researching each of their products and examining their features to determine what business values our project should have.

Personal development activities:

The concepts learned helped me gain a deeper understanding of Configuration Management and its importance. Reading through the case studies helped me understand better. I also gained a lot of insights

doing market research and formulating a proposed solution for the project. It invoked my creative side to come up with features that didn’t already exist.

Adjustment to Goals and Future Readings:

Chapter 6 was discussed during the class session, but unfortunately, I didn't have the opportunity to review it at that time. Consequently, I plan to catch up by reading Chapter 6 on my own to ensure I grasp the content thoroughly. I will also prepare for the pitch for the project and try to articulate its key features, benefits, and potential impact. Additionally, I aim to address potential questions or concerns that may arise during the presentation, ensuring a comprehensive and compelling proposal for our stakeholders.