**Student Name:** Andres Ardila – 40274138

**Journal URL:** https://github.com/AndresArdila544/SOEN-6841-LJ

**Dates Range of activities:** 23 Sept-06 October 2024

**Date of the journal: 06 October 2024**

**Key Concepts Learned**

**Risk** refers to the potential for an event to occur that negatively impacts the project. Probability of an event happening and its consequences. Risks can happen from deviations in expected outcomes, unforeseen events, or challenges in resources, tools, or technology.

*Types of risks*: technology, resource, organizational, tools, budget, estimation, quality. *Categories*: Product, project, business. Impact: Rate and quality.

First, identify potential risks, evaluate each risk qualitative and quantitative. Prioritize based on risk exposure = probability x impact. Strategies to mitigate and manage risk: *Acceptance* - acknowledge the risk, *Avoidance* - alter the project plan to eliminate it, *Mitigation* - mitigate it through schedule buffer on projects, knowledge management systems, quality gates; and/or *Transference* - transfer it. It is necessary to continuously track the identified risks. Risk Reduction Leverage is the ratio of reduction in risk over the cost of the reduction.

In the *waterfall model* risk arise because software is only reviewed at the end and correcting mistakes is more costly than the *iterative model*, where smaller sets of features are reviewed regularly.

A **Configuration Management System** is a tool for managing and controlling the evolution of a software product, ensuring consistency, traceability, and control of all changes. It ensures that all changes to code, documents, and components are tracked and properly documented, which reduces the risk of confusion, errors, and rework. The key functions include *configuration identification, control, status accounting,* and *audits*. It is critical because software is prone to frequent changes due to new requirements, bug fixes, or technology upgrades.

It's important to establish clear baselines, implement formal change control processes, and integrate automated tools like Git for version control. Regular audits and centralized systems for tracking changes are also crucial to maintaining transparency and preventing unauthorized modifications. This tool reduces lifecycle costs, ensures compliance with standards, and provides a stable, organized environment for development, ultimately contributing to the success of a project.

**Application in Real Projects:**

In real-world projects, **risk management** and **configuration management systems** play critical roles in ensuring smooth project execution and successful delivery. In a software development project, risks related to technology obsolescence or resource unavailability can derail timelines or impact product quality. By using strategies such as risk avoidance, mitigation, or transference, project teams can address these challenges. Regularly updating risk assessments throughout the project lifecycle helps to identify emerging risks and adjust plans accordingly.

However, implementing risk management strategies can be challenging due to the difficulty in predicting certain risks or the high cost of mitigation measures. Using schedule buffers to handle unexpected delays, while reducing the likelihood of project slippage, it may also increase costs. Nonetheless, the benefits of having a structured approach to risks often outweigh these challenges, especially in fast-paced or high-stakes projects. Risk management increases the likelihood of meeting project objectives on time and within budget.

A **Configuration Management System** provides significant value by ensuring that all changes made to a system are carefully documented and tracked. This is particularly important in software development, where frequent updates, bug fixes, or new feature requests can lead to confusion if not properly managed. It helps developers avoid working on outdated versions of code and ensures traceability from requirements to final product delivery. However, establishing and maintaining one requires an initial investment in tools like Git and disciplined processes for change control, making the setup phase challenging. The benefits, including reduced lifecycle costs, better product integrity, and compliance with standards, far outweigh the challenges, enabling more efficient and error-free development.

**Peer Interactions**

This week we had our group pitches. We also finished the first deliverable for the project. All the activities were completed successfully. We also discussed the quiz within the group and received feedback from each other for all the activities.

**Challenges Faced:**

The quiz was fine, although I expected a better grade. There were some tricky questions, but the biggest challenge was the pitch. Writing a pitch is not difficult, however memorizing it and reciting it it’s the challenging part. Most of the groups did fine, with some having issues with voice tone. I had to present the pitch for my group. It was impossible to memorize so I had to hold notes while delivering it but overall, I think I had a good stage management.

**Personal development activities:**

The pitch was great for me. I know I must force myself to improve my public speaking skills, and this was a great opportunity for it. I’m getting more comfortable every time I do it.

**Goals for the Next Week:**

Next week I must finish my notes to study for the midterm. Also, I hope I do better on this week’s quiz than the previous one. I need to improve my understanding on all the topics from this 4 week of class. I’ll have the midterm break for it.