LEARNING JOURNAL

Student Name: NADINE LISA MATHEW

Course: SOEN 6841 – PROJECT MANAGEMENT

Journal URL: https://github.com/ProjectPiglet/LearningJournal

Dates Rage of activities: JAN 13, 2025 – JAN 26, 2025

Date of the journal: 02-02-2025

Journal 1: Key Takeaways from Project Management & Project Initiation Introduction Class

The following are the main concepts I took away from today's class on project management and project initiation:

- **Project definition and it's key characteristics.**
- **Various project phases.**
- **Industry and project-related subprocesses.**
- **General understanding of various tasks and activities under project initiation, product initiation, and product implementation**
- **Waterfall model operation**
- **Quality metric characteristics of a software project**
- **Key roles and responsibilities in a software project and how they differ.**
- **Meaning of a project charter, how important is it, and what does it contain?**
- **Project objectives are SMART goals that quantify and define a project's success.**

Additional Observations and Insights

Beyond the core concepts, the following points caught my attention:

- A project is any activity that has a specified goal to be achieved within a specific timeframe (has a definite start and finish date) and budget. Software projects are no different from other types of projects, however they can present challenges during development.
- Routine jobs are less uncertain on the uncertainty scale than exploration tasks, which have the highest level of uncertainty, while projects are in the centre.
- The same tasks and activities from the project's initiation phase are included in the software product implementation tasks.
- The planning, monitoring, and control phase of the waterfall model includes industry-related subprocesses.
- A project manager's primary duty is to supervise the software project lifecycle, however I'm not clear how this differs from the manager's job description on **slide 20**.
- The project charter is a crucial document prepared by top management that outlines the project's purpose and objectives. It defines the project scope and boundaries, provides an initial schedule, and includes cost estimates covering human resource salaries, hardware, and software expenses. Additionally, it details stakeholder information and requires their approval through signoff.
- Once the initial cost and project schedule are refined and finalized, they will be established as a baseline. This baseline will serve as a reference for comparing actual project expenses and the time taken for project completion in the future.

- One aspect that piqued my curiosity was the ability to define the required quality level of a software product within the project scope. This led me to explore whether developing an acceptable low-quality software product is possible. I found that in certain cases, such as MVPs, prototypes, and low-risk projects, lower quality may still be considered acceptable. However, the level of acceptability ultimately depends on the specific use case, target audience, and industry standards.

Dates Rage of activities: JAN 20, 2025 – JAN 26, 2025

Date of the journal: 09-02-2025

Journal 2: Key Takeaways from Effort, Cost Estimation & Risk Management Class

The following are the main concepts I took away from today's class on Effort, Cost Estimation & Risk Management

Effort, Cost Estimation understanding:

- Learned the various estimation techniques like experience based and algorithmic cost modelling
- Under experience estimation techniques I learned the detailed steps for estimation using analogy including the calculations. I see the main factor is the Multiplication factor as well as the size ratio that drives the estimation by comparing the new project to past similar projects.
- Following analogy, I learned how estimation is done using Function Point which is an experience-based technique. This technique seemed to be more tedious for me while studying since I need to first list out all the data functions, transaction functions , calculate the unadjusted function points, then based on the scale applied on the 14 general system characteristics we must calculate the Total Degree of Influence using which the Value adjustment factor is calculated. The constants that are used in the calculations are based on real time observations, real world data. The final Function point is a product of Unadjusted FP and Value adjustment factor.
- I learned that compared to other techniques like Analogy or FP , Delphi is an estimate based on Team effort where individual estimates are discussed and finally the whole group arrives at an accepted estimate.
- In the initial phase due to higher level of uncertainty a +-25% range of the accepted estimate is considered and as the project progress since the level of uncertainty decreases a range of +-15% is considered.
- Though I have learned the experience-based techniques, the most important point that stood out is ultimately if the team has no experience on new technologies as required in the new project, then these experience-based techniques will fail to provide accurate results.
- Finally, I learned Algorithmic Cost Modelling, which is based on mathematical function where project, product, process attributes are considered and due to its complexity, it is applicable only on aerospace and the defense systems.
- As an example of algorithmic cost modelling, I have learned COCOMO cost modelling and that it has 4 models . The early design stage example seemed to be too difficult to understand .

Risk Management understanding:

- Learned definitions for Major types of risk, Risk, Risk Category and how risk can affect a project.

- All the steps in doing risk assessment.
- Learned risk analysis using both qualitative and Quantitaive.
- Learned the 4 risk response strategies.
- Learned formula for Risk exposure and Risk reduction leverage.
- Learned resource risk strategy ie. mitigation of risks using KMS.
- Learned Mitigation using quality checks/gates
- Finally Learned Risk Prioritization, risks of waterfall models and why risk is minimal in case of iterative projects.

Dates Rage of activities: FEB 03, 2025 – FEB 10, 2025

Date of the journal: 23-02-2025

Journal 3: Key Takeaways from Configuration Management & Project Planning

The following are the main concepts I took away from today's class on Configuration Management & Project Planning.

- In this class I learned the importance of configuration management, why it is needed and why CM is considered the foundation of the project?
- Purpose of CM and Benefits of CM for a software project.
- Document attributes to keep a document unique
- Good Configuration Management System Characteristics.
- Learned that the 4 key functions of CM are Identification, status accounting, auditng and change control.
- Learned what all are the elements of change control process.
- Studied the flow of change request. From the flow chart it is evident - how and on what, all the change requests are or can be applied, its dependencies, how it can be verified.
- Learned the key function of CM like – Configuration Identification, Configuration control, Configuration status accounting, Configuration auditing.
- Learned about the elements of the change control policy.
- The concept of SCCB – software change control board was something new to me. I wasn't aware of such a board from my work experience.
- From Project planning class I understood that out of all the project phases project planning is the most time-consuming phase as it involves chalking out detailed plans for Risk, Quality, Resource, Project scope, Supplier Management, Tools, Communication, Configuration Management, Effort and Cost Estimation etc..
- Learned about Top-down and Bottom-up approach for Project scheduling.
- Learned that at time of top-down approach the Project budget, duration of project, start date, end date and is very much suitable for projects that has the release date as well as the product features already fixed whereas in case of bottom-up approach it is more suitable for projects that doesn't have all the requirements ready upfront. In the case of projects like these, the requirements are captured as project progresses and based on this the release date is decided by the team. Bottom-up approach is more suitable for agile projects. In case of bottom-up approach the outputs consists of Project duration, budget, start date and end date.
- Understood the definitions of Milestones, deliverables, WBS
- Learned the most common ways of presenting scheduling
    - Bar charts which is a calendar-based method that show schedule as activities or resources against time.
    - Activity Network charts

- I was already aware of the critical path method and how to draw activity charts both using excel and Gantt chart. But this is the first time I am hearing about Goldratt's critical chain method where buffers for tasks with certainty are removed and all the buffers for uncertain tasks are moved to the end of the schedule and it is by tracking this buffer the Project manager monitors or decides if the project is in critical stage or not.
- Also learned that based on the methodologies like waterfall or iterative project planning can be different.
- Initially the line 'Depending on the lifecycle model, these iterations can be full or partial' was confusing for me as I wasn't able to understand the meaning of full or partial in iterations. But then I learned by partial it means the iteration will not consist of the whole software life cycle. Whereas by full it means the iteration consists of whole software life cycle.

Additional Observations and Insights

Beyond the core concepts, I spent some time to understand the below terminologies in detail

Importance of Impact analysis:

1) From the below references, I understood why impact analysis is needed as part of a change request
2) It was indeed amusing to learn that impact analysis can either make or break the career of a software engineer.
3) With every change request it is a necessity to perform an impact analysis, and the ideal case is having a less impact so that with every CR the software product gets better and better.
4) In the worst-case scenario, software gets a lot better, but a whole lot worse.
5) Negative impacts can result in new bugs which are called regression defects and can be extremely problematic from marketing perspective since clients may start to lose faith
6) Regression testing is usually used to detect regression defects well in advance and resolve them before sending a status update.
7) Since waiting to catch all regression defects is not advisable, it is critical to perform an Impact analysis which is a proactive process of assessing an impact/risk

Tangled Commit:

A commit is termed Tangled Commit if it has a bug fix along with a new feature or a refactored code. As per the narrator, these commits happen when the developer tries to be a hero, and this usually happens when firms have a 'leave it better than you found it policy'. But in case due to the refactored code if regression happens then we will have to revert the refactored code and retain the bug fix alone which is quite tedious and messy.

References:

- https://www.youtube.com/watch?v=imG0LyEZTVo
- https://www.youtube.com/watch?v=FckiWBFUfq8

Dates Rage of activities: MAR 10, 2025 – MAR 16, 2025

Date of the journal: 16-03-2025

Journal 4: Key Takeaways from Project Monitoring & Control and Project Closure

The following are the main concepts I took away from today's class on **Project Monitoring & Control** as well as **Project Closure**.

- Understood how Project Monitoring and Control is done.
- Understood how the Project plan is monitored and control by comparing the actual events against initial timelines and cost estimates.
- Understood how a Project Monitoring and Control system is designed.
- Understood that any change to initial cost and timeline baselines should be done only after a review and approval by the change control system.
- Understood different methods for gathering information for monitoring purposes.
- Learned about EVM and understood the key point that EVM measures the project progress in dollar value that is how much we have earned so far and not by how many days of effort have passed.
- EVM is mainly used to understand Schedule variance and Cost Variance.
- Understood the definition of Project metrics or Performance indicators
- Project metrics related to Project schedule is schedule variance and related to Cost Budget is Budget variance.
- From Project Closure slides, understood what all the mandatory deliverables before project closure are.
- Understood that before project closure the raw project data should be cleaned by filtering for archiving.
- Finally, the lessons learned should be documented.

Application of Project Monitoring and Control in a Marketing Campaign

In a major product launch, effective project monitoring and control are crucial for ensuring a successful marketing campaign. As a project manager, tracking key performance indicators (KPIs) such as website traffic, social media engagement, and sales helps assess the campaign's progress.

If engagement levels fall below expectations, corrective actions, such as reallocating resources to paid advertisements, can enhance audience reach. By continuously monitoring performance and making strategic adjustments, the campaign remains on track, maximizing its chances of success.

Reference : https://resourceguruapp.com/blog/project-management/project-monitoring-and-control

Gantt Charts: The Importance of Gantt Charts in Monitoring and Control

Gantt charts continue to be a versatile and evolving tool, ideally suited for managing the complexities of modern project management. As a widely used tool for project monitoring and control, they offer a clear visual representation of tasks, dependencies, and timelines, helping teams stay organized and on schedule. Beyond simplifying timelines, Gantt charts foster collaboration, enable real-time tracking, and support data-driven decision-making. By incorporating Gantt charts into project workflows, organizations can optimize processes, boost efficiency, and ensure the successful completion of projects.

Reference: https://www.omnitas.com/advantages-of-gantt-charts-in-project-planning-and-monitoring

Dates Rage of activities: MAR 17, 2025 – MAR 30, 2025

Date of the journal: 30-03-2025

Journal 5: Key Takeaways from Software Lifecycle Management, Requirement Management, Software Design, Construction, Testing, Release Management.

The following are the main concepts I took away from **Software Lifecycle Management**:

- Learned the definition of Software Lifecycle
- Learned about two major lifecycle Models – **Waterfall** and **Iterative** Model and why it is important.
- Learned about the different stages of the Software lifecycle models and the contrast between the models.
- Learned about **Capability Maturity Model Integration** and how it helps with an organizations' process.
- Learned about CMMI's key features, Maturity Level's and benefits.

Real-world application of CMMI:

A mid-sized e-commerce fashion firm faces operational strain as it scales its product range and global customer base, struggling to maintain quality and efficiency. The root issue appears to be an underdeveloped Capability Maturity Model (CMM), with two key hypotheses: lack of standardized global processes and misaligned IT systems with business strategy. To address this, the company adopts a structured 5-phase CMM enhancement methodology. First, it conducts assessment and benchmarking to identify gaps against industry standards. Next, it redesigns processes for efficiency gains and aligns technology to support new workflows. The firm then implements changes with robust change management before establishing continuous improvement mechanisms. This disciplined approach aims to achieve greater operational maturity, improved agility, and stronger IT-business alignment to sustain competitive growth.

Reference:
https://flevy.com/topic/capability-maturity-model/case-ecommerce-retailers-capability-maturity-model-advancement-fashion-industry

The following are the main concepts I took away from **Requirement Management**:

- Learned about Requirement Gathering
- Gained an understanding into functional and Non-Functional Requirements & Requirements Document.
- Target Qualities for Requirements Engineering and Flaws and the errors associated with it.

Main Concepts from **Software Design, Construction, Testing, Release Management.**

- Learned about the Software Design Management process including the various approaches like Top-Down and Bottom-Up approach.
- Learned about the Software Construction Process, Software testing Management process and the Release Management Process.
- Gathered an understanding into Defect Tracking.

Real-world incidents caused by software design flaws underscore its critical importance in system safety and reliability.

1) Boeing 737 Max crash : The report by Ethiopian investigators into the March 2019 Boeing 737 Max crash concluded that the accident was primarily caused by flaws in the aircraft's software design rather than pilot error or airline performance. The investigation highlighted issues with the **Maneuvering Characteristics Augmentation System** (MCAS), an automated flight control feature that repeatedly forced the plane into a nosedive due to erroneous sensor data. This finding aligns with broader concerns in software design management, where robustness, compatibility, and rigorous testing are critical to ensuring safety—principles emphasized in software lifecycle practices such as iterative refinement, version control, and design validation (as discussed in Chapter 11 of Software Design Management). The incident underscores the catastrophic consequences of inadequate software design and the need for stringent oversight in safety-critical systems.

   Reference : https://www.nytimes.com/2019/09/18/magazine/boeing-737-max-crashes.html

2) **Ariane flight V88:** In 1996, a European Ariane 5 rocket carrying a satellite payload veered off course just 37 seconds after launch due to a software bug, leading to its self-destruction—a safety measure to prevent ground casualties. The failure was traced to reused code from the Ariane 4, which was incompatible with the Ariane 5's flight conditions. The disaster resulted in over $370 million in losses but prompted significant improvements in software engineering practices. This incident remains a key case study in the risks of repurposing legacy code without proper validation.

   Reference : https://en.wikipedia.org/wiki/Ariane_flight_V88