**LEARNING JOURNAL**

**Student Name:** Habeeb Dashti

**Course:** Software Project Management

**Journal URL:** <https://github.com/habeebdashti/SOEN6841journal>

**Duration:** 10/03/2024 – 14/04/2024

**Date:** 10/04/2024

**Final Reflections:**

Under this period of Software Project Management course, we dive into several crucial concepts that are vital from chapters 8 to 14 and reviewed chapter 7. Here's a breakdown:

* **Project Monitoring and Controlling**

It was reviewed that the baseline is crucial and that it comes from the planned schedule and budget, then they are compared to true/actual performances to determine EVA via deviation, which allows for the EVM check up to take place. After careful monitoring and understanding why it is occurring, the controlling begins to adjust appropriate areas affectively. Instead of project scope, we take deliverables into account.

* **Software Project Closure**

This chapter emphasized upon releasing resources and lessons learned. The released resources are passed onto other projects to occupy them, and the lessons may include better alternatives, management, solutions, negotiation, risk dealing, and noting down what works and what does not. The raw project data is filtered before archived data to obtain clean data, clear app, etc., that can be used for further activities.

* **Software Life-Cycle Management**

In this chapter, the clear difference between waterfall model and iterative model is depicted so that we understand which one to go with depending on the project size and dependent factors such as rework. If we choose to go with iterative model, then it explains about agile, concurrent programming, SCRUM, sprint, XP programming, incremental, etc., as the available variety of options that would suit better depending on the type of project. Therefore, we tend to understand top-down and bottom-up approaches clearly in development. The phases are the same for any model and its types, and they contain requirements, design, construction, testing and release. The approach towards them varies and the below topics focus deeply upon these phases and their managements.

* **Software Requirements Management**

It was conveyed that requirements are the initial stage where user interaction is carried out to understand what product the user is expecting as best as possible and any changes in the requirement should be updated immediately such that design and construction update takes minimal rework. Its types, levels, sources, and process, change and validation cycles were discussed even in Iterative model.

* **Software Design Management**

We learnt that design is nothing but the architecture in one word. It includes user interface and databases. The creation of design also includes top-down and bottom-up approaches where the whole design is prepared first and then the internal components design, and vice-versa in the other case respectively. Different versions harness different own designs, it should be robust to be able to incorporate any changes easily. There are distinct approaches to design such as Object-Oriented Design, ER Design, etc. The iterative models specially use refactoring as their design technique to keep a clean code and get rid of bad code smells as it allows re-design to become seamless. A good design entrusts good security and scalability of the software project. An emphasis on coupling (large no. of object/parameter calls) and cohesion (classes with many concepts) was made in the class and explained that they work like a seesaw in opposite directions of increase-decrease.

* **Software Construction Management**

The construction management mainly includes desk checks, walkthroughs, reviews, and final inspection of the source code built. Following the coding standards strictly and its reasons are shared. Coding methods such as pair programming, review programming, test driven developments, etc., were introduced and defined. The importance of having a controlled branching is learnt as extra branches are not only unnecessary but also problematic in case of available codes. It is highlighted that testing should be done at both unit and integration level to avoid high-cost support after release. Pair and review programming were focused upon their differences in XP. The advantage of Pair programming is that it helps with knowledge transfer and ownership.

* **Software Testing Management**

What are automated and manual testing, when should they be used and by how much should they be is learnt. Testing is divided into 2 parts i.e., Verification and Validation and how they are distinct by a single word ‘execution’ is elementary to grasp as this defines Verification to be static testing (no execution of the code) and Validation to be dynamic testing (uses execution of the code). The defect tracking, logging, correction, and closure is also covered under the Testing Management.

* **Software Release Management**

Post-deployment of Software Release comes Maintenance and Support that has about 70% higher cost than building the Software that is why it is critically important that during the previous phases, especially development and testing to work on finding and fixing faults as much as possible to reduce the cost of the maintenance. Maintenance techniques such as re-engineering, reverse engineering and forward engineering were introduced. Types of maintenance such as preventive, corrective, and adaptive are explained thoroughly. Release types such as Alpha, Beta, Internal and Alpha release are learnt. It is understood that under different maintenance processes, Boehm’s model is based on economic models such as calculating Return On Investment (ROI) and Osbron’s model is based on communication gaps which is fixed in stepwise by maintenance plan, QA plan, metrics and reviews.

**Overall Course Impact:**

The course has overall helped me understand a broader perspective of a complete software project. It has not alone enlarged my knowledge about development and testing but also all the other aspects that come before and after development encapsuled under a project management. It has opened my eyes to understand that experience in a single field is not enough to become a manager, instead it will take a general knowledge of various things considered in a project, and specific knowledge and experiences in handling things properly. The essence of becoming a successful manager is achieving the best possible in the deliverable time. The project included in the course has helped me understand aspects taught in class better and as the course came to near end my understanding regarding previous topics turned to become even crystal clear.

* **Clarity on Software Projects –** At first, I understood the difference between an operation and a project which helped me realize that most of the time in my life, I have done operations in the name of project. A project is something that brings a unique possible solution to an existing problem(s).
* **Deliverable 1 outcome –** I got clear picture of Project identification and initiation by working on charter, scope and objectives. Stakeholder analysis helped me truly understand who stakeholders are related to the project, identifying current real problems in existing similar software systems and bringing in real do-able unique solutions to them was amazing.

Market analysis helped to talk to existing users of the similar systems from university and outside which helped me understand that just doing research based on existing business such as TAM, CAGR, etc., is not enough but also listening to people, their feedback and opinion brings in more answers than you could find. The catch is to respect what people are saying and provide accordingly as the end users always decide who stays in the market pool.

* **Deliverable 2 outcome –** It has helped me understand how to find the cost of the project by estimating it using history of the similar projects and considering the cost of the unique solutions we are brining in the form of cutting-edge technology.

Assessment of feasibility under technical, operational and economical form helped me understand the importance of it in a project. The risk assessment and planning section of the project helped me grasp well why we need to plan before managing risks because having a risk plan before risk management gives us an upper hand.

By creating SDLC timeline using Gantt’s chart and WBS Structure before that helped me understand and be confident about my plan working out in the project successfully. Choosing wisely to go with Agile method under Iterative development considering the future of our project was call I would never make or even think about before this course.

* **Topic Analysis outcome –** I picked a topic that is under looked but causes a serious problem in the current industry i.e., meetings and shed light on how we could have efficient meetings and yet keep our actual work more productive. This gave me an opportunity to express what I consider a real problem and proposed a working solution for it. Here, I learnt that there is more wastage globally in meetings than I expected from my own experience.
* **Final Presentation –** In this presentation, we got an opportunity to realize what were the missing holes in our deliverable 1 and 2, and we patched fixes to it according to the knowledge gained from 10th March to 14th April. One of the best example is reworking and fixing our timeline according to project closure, maintenance and future support and showcasing it in our presentation with an updated SDLC project timeline.

**Application in Professional Life:**

As of now, my professional life in Canada being an international student is at a restaurant. Before taking this course, I just used to do my work and what I was asked to do and just leave without talking to anyone. Now, I talk at my work, proactively provide smart ideas because at quick restaurant when a wave of flooded orders gets in you can’t do everything alone. So yes, in my professional life even though it is not in technical or software field, I have still considered to start using my ideas, communicating clearly, initiating teamwork, and providing results by being prepared with experience.

Each time I leave from my job now, I take into account lessons learned as taught in project closure and remember them so that I can perform better and help the restaurant team perform better because of the experience I am collecting without even being a manager yet.

**Peer Collaboration Insights:**

Peer Collaboration has played an enormous role in successfully learning, revising, understanding better and presenting a successful implementation of the concepts learnt. The exchange of ideas, sharing and respecting different perspectives and constructive feedback has contributed to make it possible since the beginning and at increased rate till the final presentation.

* **Knowledge Exchange –** Peer interactions through discussion and group activities helped me in gaining insights into project management and problem-solving strategies. It enabled us to study complex topics collectively and improve comprehension of the subject.
* **Team Collaboration –** It helped in implementing theoretical knowledge into our project efficiently as my peers belong from diverse backgrounds and skills sets which encouraged creative and innovation project management. Effective communication and action being taken as team resulted in better decision making.

**Personal Growth:**

The course and its activities including examinations have played a significant role in nurturing my growth with valuable skills and perspectives for both academic and professional development. The journey has been enriching, challenging, and ultimately empowering by laying a foundation of Software Project Management in me. It makes me feel confident having these basics set down in me.

*Most importantly, giving presentations and pitches has helped me in working on my introvert nature to be able to present myself publicly*. The following are some of the personal growth attributes I feel that I have developed in my opinion which I will take away as lessons learnt just like in Project closure:

1. **Passion for lifelong learning in what I enjoy.**
2. **Deep self awareness and reflection**
3. **Strengthened communication skills**
4. **Problem solving using improved critical thinking**
5. **Gained adaptability and resilience**
6. **Confidence and happiness in myself because of the sincerity of the journey and not the results**