**Reflection on Software Engineering Project Management**

https://garciar.github.io/University-of-Essex-Richard-Garcia/Module%205.html

**Introduction**

Self-reflection is essential for unlocking your career potential because it develops the ability to reflect on one's own strengths and weaknesses and increase knowledge and credentials. It is constructed around three questions. What? So What? Now What? Software project management is the science and art of planning and leading software projects; as a result, software projects are planned, implemented, monitored, and even controlled (Alba & Chicano, 2007). Because of the various penalties incurred during software development and maintenance, which primarily result from poor management, software engineering project management has received a lot of recent attention. There should be a dependable method for developing high-quality software, reducing the time it takes to deliver it to the customer and reducing the overall cost of the software life cycle, thereby improving employee morale.

**What?**

Interpersonal communication, conflict resolution, and conflict management. Participated actively, frequently, and truthfully, ensuring that I communicated with the other members to increase the likelihood of project success and mitigate any problems (Fairley,1985). Insensitivity and misinterpretation of requirements can occur when others are not involved. Risk assessment and management entail being involved in assessing risks and developing risk-management strategies. As part of the software engineering process requirement analysis, I identified a client's needs and was able to design a solution.

In software engineering, change management entails identifying, documenting, analyzing, prioritizing, and agreeing on changes to the project management and controlling changes, and communicating with relevant stakeholders. I identified, documented, and prioritized software releases and oversaw the release schedule. Furthermore, I maintained and updated where customer needs were involved, because they are always finding bugs, requesting new features, and requesting different functionality and more updates. This was done to meet the needs and expectations of my customers.

**So What?**

Because putting together a successful development team is not as easy as it may appear, team members must have faith in their ability to achieve the common goals they have set for themselves. Most projects, as previously seen, fail due to a lack of collective and individual confidence in the project's success. In my experience as a member of the development team, I learned of the strategies and best practices that should be incorporated to ensure the software development team's success through collaboration (Howard, 2001) Self-organizing and cross-functional work at the individual level optimizes the overall effectiveness of the development team; however, team members must communicate and implement shared responsibilities. The team as a whole should be held accountable. Team members must leave the codebase as clean as possible, understand and share their customers' interests, criticize their own ideas rather than criticizing others' ideas, share past and present experiences to offer new team insights, and have fun and trust others throughout the process.

When putting together a software development team, it's best to include critical thinkers; ideally, every software development team should be able to provide solutions and optimal ways to get there. Prior projects that team members have completed, for example, would serve as an indicator of their quality of work because you get to see what kind of clients they worked with and how they met the project requirements to solve the given problem (Thayer et al.,1980) In addition, potential partners' references should be checked because experienced companies can easily provide you with case studies and overviews of completed projects, as well as all the necessary information to aid in decision making. High-quality work comes at a cost, and cheap is always expensive. Some of the problems that may arise as a result of low pricing include poor communication, a lack of experience, poorly written code, a lack of tests, and poor documentation.

**Now What?**

Software engineering project management has greatly impacted my personal and professional development because it has evaluated and cultivated my soft skills. A well-developed project necessitates more than just programming and a communicative and respectful team. A collaborative project necessitates emotional intelligence. It has prioritized my social and communication skills because it is easy to place too much emphasis on technical output at the expense of team building and communication when managing a software development team. I would therefore establish a clear priority for my team that empathy and emotional intelligence are just as important as delivering high-quality work; this I would achieve by creating opportunities for peer learning and teaching and recognizing distinct personality types.

I learned to listen and communicate proactively actively; communication is two-way, and someday as a business manager, I will actively listen to my team members to understand the project's current state, asking questions such as "What did you work on?" Did you live up to your promises? What is your reasoning for doing so? What could be done better? (Thayer et al.,1981) Furthermore, communicate proactively and follow up on individuals and teams without being asked. Taking a genuine interest in each member of the team and each division.

I would also make an effort to improve team communication by defining roles and goals from the start, as miscommunication stems from a lack of clarity about roles and responsibilities. I would also discuss what the various roles will be and their expectations. Because there is increased productivity, I would consider hiring remotely. Businesses save money, employees avoid the stress of walking long distances to and from work, employee turnover is reduced, employees save money, and I would gain access to global talent. Finally, software engineering project management is lovable and enjoyable regardless of the technical and business requirements. If users love the software, you will love the job as well, and members will be motivated to keep improving it.

**Conclusion**

With the ongoing demand for software and product development, assembling the right software development team has never been more critical. Projects can be accelerated and productivity maximized by keeping the aforementioned individual experience, contribution, and impact in mind.

**References**

A. Howard's formal name is Alexander Howard (2001). On-site project management for software engineers. The ACM Communications, 44(5), pp. 23-24.

A. Pyster, R. C. Wood, and R. H. Thayer (1980). The Task of Software Engineering Project Management is a Special Feature. 13(08), pp. 51-59.

Chicano, J. F., and E. Alba (2007). Management of software projects using GAs 177(11):2380-2401.

R. C. Wood, R. H. Thayer, A. B. Pyster, and R. H. Thayer (1981). Significant issues in the management of software engineering projects. (4), 333-342, IEEE Transaction on Software Engineering.

R. Fairley's (1985). Ideas for software development. McGraw-Hill, Inc. is a publishing company that specializes in educational materials.