

Observatory and Participatory Project Factors for Software Projects

Abdulaziz Alsadhan

Department of Software Engineering,
College of Computer & Information Sciences,
King Saud University

Naveed Khan

Department of Computer Science,
College of Computer & Information Sciences,
King Saud University

ABSTRACT

The success of the software project is measured in terms of attainments of its success. To ensure the success, the software process must be mature enough to produce trustworthy software and also fulfil requirements according to the needs of the users. Hence, in literature review numerous software project factors have been identified. This paper evaluates the project factors by categorizing it in participatory and observatory factors. Therefore, it is deemed essential to perceive evocative legends to these factors by quantifying them. The identification of these factors plays an important role in successful software project completion.

Keywords

Software Development Life Cycle (SDLC), Software Risk Management (SRM), Reusable code

1. INTRODUCTION

A project is measured in successful category if it delivered in time, within budget and attain adequate quality. However, the measurement of successful project is hardly to achieve due to its tangible characteristics and difficult to be agreed upon. Generally a successful project must achieved its goals and expectations. Different project experts have different sentiments i.e. what is critical for project success, what is measured during project phases and what should be the project priorities. Due to this fact, which critical factors may have strong impact on the final success of project is still unsolved and difficult to identify [1]. Moreover, the researchers are still trying to identify more and more risk factors which helps in project success. Since better performance and cure can be achieved by identifying more risk factors [1].

2. LITERATURE REVIEW

A huge number of project factors have been identified which may cause project success or failure. The most common factors were identified in the literature study.

Boehm et al. [2] argues that estimation of a software project required four basic factors such as software size computation, effort estimation in person hour, cost and budgets calculation and resource allocation. COSMIC-FFP as an international standard in software estimation has also declared by author. In conclusion the authors determine the three valuable factors to be considered for any successful project that is cost, effort and quality. In this study the authors also proposed SMART techniques to make the project successful.

Nasir et al. [3] proposed five factors for project estimation that is time, effort, cost, Resources and specifications. In this study the authors mainly focus on the effective balancing among cost, time and space, effort and resources.

Reel et al. [4] described theoretical guidelines about critical success factors for a successful project. The key focus on to elaborate the improvement standards of a software process. The authors also identified the critical success factors for a successful project such as budget and estimation, documentation, tools technology and quality in this domain.

Maloney et al. [5] illustrated the study on evolution of project performance in terms of time, cost and quality to evaluate whether project objectives achieved. But attaining succeed in order to complete project must be more significant than simply achieving cost, schedule and performance specifications.

Parfitt et al. [6] suggested that the objectives and prospects relate to a diversity of elements comprise of technical, financial, educational social and professional concerns.

To summarize the above literature review according to critical success factors, it can be concluded that the success factors 'cost & budget' and 'effort' have universally identified by all authors in this domain. However the 'resource' factor also got ample consideration. The other identified factors comprise of 'Time', 'Specification', 'quality and documentation'. Some other project factors also exists in HI-SDLC, hence the success or failure of the project are not only limit to the above mentioned factors.

3. PROPOSED MODEL

3.1 The Study of Specific SDLC

The objective of this study is to analyze the findings in a way that the ultimate outcome of resource optimization by risk minimization becomes practical. The focus of this study also evaluates the findings with least amount of complexity and most amount of understanding. This model defines two core objectives:

3.1.1 Optimize Resource Allocation

Optimize resource allocation plays an important role in successful project management. Resource allocation often referred as constrained-resource scheduling. The exertion of rescheduling project tasks is to efficiently utilize the limited number of resources while keeping the unavoidable extension of the project to a minimum.

3.1.2 Minimize Software Risks

In recent years software has become backbone for enterprise business. Most advanced technology and high level of knowledge associated with software development. The main issue is to identify and minimize risks in software development. To mitigate or minimize risks in software development leads to a successful software system. The common cause of project failures are due to lack of defined approach to risk management. To improve project of success, Software risk should be mitigated or minimized.

3.2 Observatory and Participatory Factors

A detailed model is presented in the Fig. 1 that represents the implementation of this study. In this model there are eight project factors that have been mentioned, including Time, Cost, Team size and Computational resources. It is observed that the time, cost, team size and computational resources are four factors that participate in the software development life

cycle and they make the lifecycle run by injecting the resources, while other four factors are observatory factors to observe that if the SDLC is working well? Or is there any inconsistency that makes it difficult for the SDLC to work properly? The inconsistencies have been channelized by partitioning them into specific classes of the risk factors.

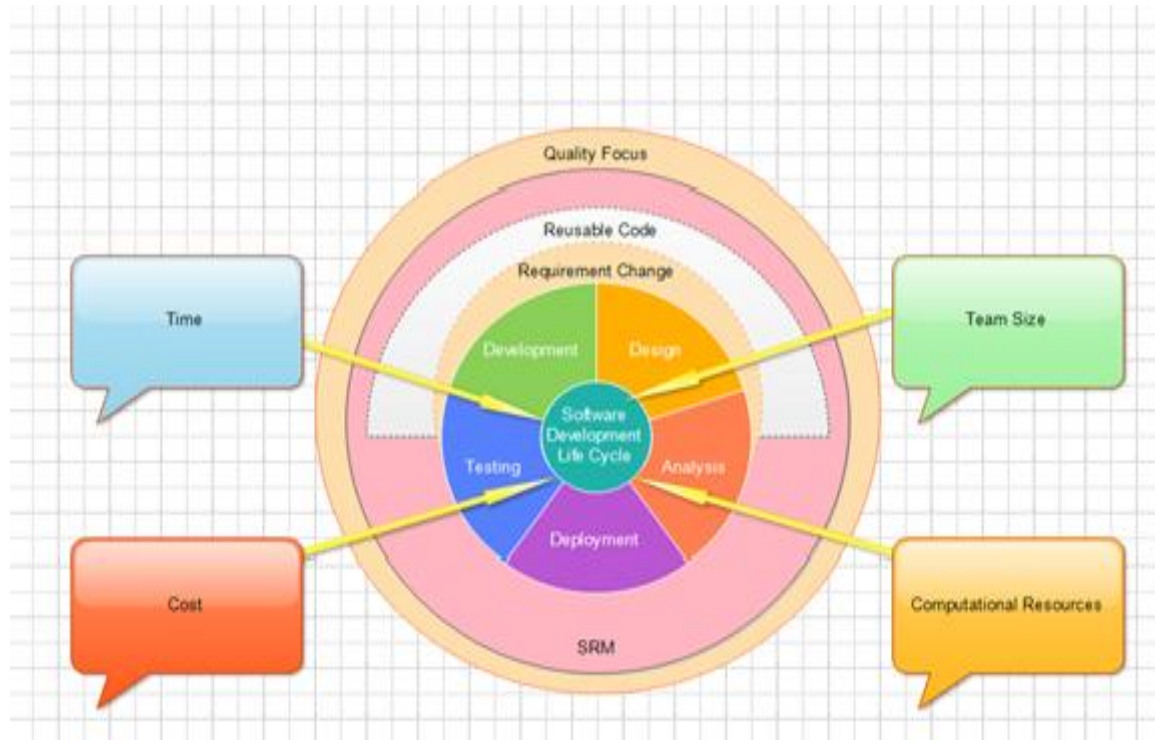


Fig 1: SDLC specific to this study

3.2.1 Participatory Factors

3.2.1.1 Cost: It is the essential factor to develop a product. It includes cost of technological resources, human resource, tools, environment, technologies and risk management. It is considered as a significant development factor so massive percentage of this cost is consumed on the salaries and supplementary benefits of the personnel [7].

3.2.1.2 Time: Time is required for the development of software or components which can highly effect the total cost of software [7]. Time is assigned to specific problem with respect to its size. In addition, available resources can also be utilized which could help and coordinate in the assigned problem.

It is the most important feature which has a high impact on the success of a project if it's achieved within time. "Effort in person month is a yardstick to measure the time needed to develop a project by a single programmer. The EPM measure describes that how many human resource is required to be deployed for the execution of a project. If a project can be completed by four team members in 4 months, the project is said to have EPM of 16. Smaller projects possess lower EMP." [7].

3.2.1.3 Team size: A specific task is accomplished by specific number of team members. Each team comprise of

specialized personnel who perform their duties in specific domain. However, a specific team can be specialized in all domain who perform their assignments from analysis to deployment of the software. It will very prominent for small project with low budget but will be infrequent for medium and large projects. Large projects have separate specialized teams for each phase in specific domain to execute their tasks for SDLC [7].

3.2.1.4 Computational Resources: To develop a software system usually it required hardware and software resources which is covered by computational resources. The hardware resources may include but may not limited to computers, printers, stationary, communication and network devices where in software resources usually required licenses to registered software. These registered software further used for employees training. In recent years hardware and software resources significantly increased in large projects to be utilized for development purpose.

3.2.2 Observatory factors:

3.2.2.1 Requirement Change: Generally the requirements reflects contents and basic structure of software that what kind of software should be developed. The probability of success about software increased with precise and accurate requirements while software failure may occur if requirements are frequently changed. The requirement output may become

radical specifically in large projects where massive financial and technological resources involved. Likewise, small projects may have same outcomes but due to limited financial and technological resources, it may not be visible and effect the progress of the project as it may happen with large projects [7].

3.2.2.2 Re-Usable code: The concept of reusable code originated from the orientation of Object oriented Programming. In different ways the availability of reusable code helps the development effort. It not only reduces the development time and effort but also decrease the time consumption on testing software components.

However, if the accessibility of a specific component inside the organization is not available, so the component is perceived from online stores. The components which are available commercially are called Commercial Off-The-Shelf (COTS) [8]. It is possible that the required components may not be available in COTS so the component is developed and kept in COTS to be used in future [8].

3.2.2.3 Software Risk Management (SRM):

The project risks are managed by using software risk management which have defined processes, methods and tools for it. The main purpose of risk management is to point out potential risks prior to their occurrence in order to plane the risk handling activities, such that it enable us to cough with any risk that could affect the objectives of the product or project during its life. This practice should be started in the beginning of the project planning and carried out throughout the life cycle of the project [9]. Software project risk management is basically a principle in which the project team investigates about the events that may have negative impacts on the project and determines the probability of occurrence of such events. It enables us to make proactive decisions about what risks are more important to us; and how to deal with such risks. The strategy for risk management, the risk management process, and the techniques, methods, and tools to be used to support the risk management process is provided by the Risk management planning.

3.2.2.4 Quality Focus: The main emphasis of each project is to attain better quality. It can be the basic and default requirement for any project but certain project need really high quality and for achievement, a huge amount of time is utilized specifically in large projects as it have massive financial and technological resources.

4. CONCLUSION

This study concluded with identifying successful software project factors. The software project factors divided into two

categories that is participatory and observatory factors. The participatory factors comprise of four factors Time, cost, team size and computational resource to review and participate in the software development life cycle and also make the lifecycle run by injecting these resources. The observatory factors also consist of four factors quality focus, reusable code, SRM requirement change that govern as well as observe the overall functionality and consistency of SDLC. In future, we will emphasis on empirical study using the proposed model to classify software project factors.

5. REFERENCES

- [1] S. Rashidah, M. Nasir, I. C. Maznah, J. Maizatulriah "Comparative Studies on Factors Influencing Success Completion of a Project" IEEE Colloquium on Humanities, Science & Engineering Research, 2012.
- [2] B.W. Boehm "Software risk management: principles and practices" Journal of IEEE Software vol. 8(1). Pp.32-41, 2002.
- [3] A. Nasir, F. Mehwish "A Survey of Software Estimation Techniques and Project Planning Practices" International conference on Software Engineering, Artificial Intelligence, Networking and Distributed Computing, Computing, pp.305-310, Las Vegas, 2006.
- [4] F. Reel, S. John "Critical Success Factors In Software Projects". Journal of IEEE Software, vol.16 (3), pp.18-23, 1999.
- [5] W.F Maloney "Framework for analysis of performance" Journal of Construction Engineering Management, vol. 116(3), pp. 399-415, 1990.
- [6] M.K. Parfitt , V.E. Sanvido "Checklist of Critical Success Factors For Building Projects" Journal of Management in Engineering, vol. 9(3),pp. 243-249,1993.
- [7] B. Shahzad, A. Said " Application of Quantitative Research Methods in Identifying Software Project Factors" International Journal of Information Technology and Electrical Engineering ,vol.1,2012.
- [8] McKinney, Dorothy "Impact of Commercial Off-The-Shelf (COTS) Software and Technology on Systems Engineering", Presentation to INCOSE Chapters, Accessed on 1st May, 2013.
- [9] A. Chowdhury, S. Arefeen "Software Risk Management: Importance and Practices" IJCIT, vol. 2, 2012.