International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

Factors Determining the Growth of Software Project Management

Rachida Radjabou[1], Subrata Sahana[2], N. H Umar[3]

[1, 2, 3]Sharda University, Greater Noida (NCR, Delhi) - 201306, India

Abstract: ***The software industry's competitive nature makes it natural that software managers and developers face several crucial decisions in managing the software project. These decisions are taken to enhance processes maturity and product quality with improved planning accuracy and monitoring control. In this study, the factors determining the growth of software project management were analyzed. This study used an online survey to collect the necessary data relating to the development, classification, consideration, priority setting, and preparation in software projects. It was observed that team incapability, time constraint, limited testing criteria, customer's inability to understand quality specifications, Budget limitation, limited ability to handle quality requirements, and lack of customer involvement are the major constraints in software project development. The analysis indicates that quality criteria, performance, security, usability, team capability, and customer involvement gained more consideration in the context of software development. Finally, it was recommended that project managers and developers should learn how essential it is to delegate specific roles to avoid difficulties resulting from a lack of clear accountability for the required specifications in the production of software.***

Keywords: Software Management, Software Developers, Quality Requirement, Risk Management, Constraint

1. Introduction

In the early 21st century, there is a monumentalincrementin the acceptance of agile software development. However, managers and developers started to understand shortcomings and deficienciescharacterized in the agile practice. Restricted predictability and failure of the project to go in the right direction are the major challenges relevant to the software management project. [1-5]. Software development companies pursue capability and maturity models as a procedure to enhance progress maturity and products quality. Software companies must statistically track the performance of their vital sub-processes in order to achieve high maturity levels. This can be used to predict and conduct an enhanced project management with improved planning accuracy and monitoring [6-10].

The software industry's competitive nature makes it quite obvious that there are logical consequences that software professionals will be confronted with several difficult choices. These decisions are taken at various points of the life cycle of software creation and at variouscooperates hierarchical levels. Decisions range from the implementation decisions of developers, going through project management decisions, portfolio decisions and eventually making crucial management decisions. The decisions may be effective or ineffective, resulting to successesstories or failures [11-13, 16]. These interactions can be registered and preserved within the organization and may be regarded as memory of the company. Memory might be rules, managing scenarios or advice about some key issues, such as improperly handled details, or over satisfactory outcomes, occurred or presented themselves in certain situations or cases. [11, 12, 14, 15].

Project management strives to produce projects on schedule, according to defined criteria and within the planned budget, with the agreed scope and quality. Project management achievement is viewed as correctly preparing the project at the outset and then implementing the project according to the schedule. Earned Value Management (EVM) is a tool widely used to critically evaluate project success in terms of reach, expense and timing. It effectively compares the project's expected work and completed work and measures the worth of this accomplished task [17-21].

In order to determine software efficiency, software maintenance is one of the most significant characteristics. Maintaining software includes a large number of subtasks, ranging from minimal to complexes, which make software much easier to meet the constantly changing requirements of customer. Maintenance can be achieved early in the development stage by using software forecasting, assuring substantial savings on maintenance costs. Although predicting early-stage maintenance of software is quite challenging, because software systems behave so idiosyncratically, this area remains unexplored due to its lack of researchers' understanding of system behavior [22, 25, 27]. The researchers behind the Software Maintainability Prediction (SMP) Framework use several different types of mathematical, machine learning, and evolving models on historical data in order to train various types of complex models with the purpose of keeping track of all kinds of software updates. [22, 24-27].

The risk management (RM) performs vitalfunction in the project development, since it enables risks to be detected and addressed promptly during project implementation. Risk Management for Software Projects is made up of procedures, approaches and techniques commonly used in various stages of the development of the project. Product risks can include late delivery, increased production costs, increased project timeline, or product failure, but anything that increases the period of time and adds to costs has a negative effects on quality.Right detection and control of the multiple risk factors would also help increase the success rate of the project and achieve quality software [28-32].
The objective of this study is to discover success criteria which will be used to assist with the best practices for software project management and implementation o. The research also defines and classifies the challenges of

managing the software project and identifies the components of the software risk and how to mitigate these risks.

2. Research Hypotheses and Framework

This research provides valuable information that may be used to aid software projects in deciding what they can focus on and optimizing their efforts on. The success software development project is characterized in terms getting quality ahead of schedule, under budget, and still satisfying the customer. The five success elements considered as variables are found to be most frequently connected to successful software projects.

2.1 Earned Value Management (EVM)

The Earned Value principle has been used as a tool since late 1960s; it is adopted in its simple form by calculating output of "anticipated expectations" using "Expected values" versus "actual costs". EVM is used as a reference to promote efficient management of projects [17, 19]. The project's variance and phase demonstrate the changes (phase shifts) on the screen demonstrate the project right now. If the SV is positive, this means the project is on track for a year ahead of schedule. The cost of the project must be in front of the cash value is less than zero, or else the budget will be missed. The indices (SPI and CPI) and PPI describe the progress and effectiveness of how much it costs to accomplish a task at different stages of development. If SPI is less than one, then the project is behind schedule. If SPI is less than one, then it means that the project is behind schedule. When there is only one CPI in a project is over budget, it shows that the work has not been completed within the anticipated time frame, and is done behind schedule. This approach follows the basic theory that the future is defined by what has happened in the past and depends on the decisions and patterns that have emerged from the past.

2.2 Risk management

Risk management can be describes as a collection of organized actions to guide and monitor the organization regarding risk. This involves the systematic application of principles, procedures, and processes to the recognition, risk assessment, risk response planning, and executionand communication relevant to the activities conducted with each stakeholder [28]. Inadequate risk and incentive management are the key causes of project failure; apart from these factors, project managers and organization lack the capacity to overcome the problems that occur. Risk management is a dynamic process that involves decision-making skills and expertise, as well as the analysis of project data that will be used to forecast future incidents and their impact on project outcomes. However, avoiding it will lead to new risks for project sponsors as well as additional expenses, which can sabotage a good client-company partnership [28-30].

2.3 Project Planning

Planning technique uses a modeling methodology for designing the project environment in order to complete the project on schedule and on budgetonce the project manager has imagined their project environment, they implement that as an iterative process on the development machine to run on the simulation. Then, they make decisions according to the schedule and budget based on the outcome of the simulation. The composition of the project team, strategy and project outlook and selecting which development approach to engage are the elements chosen in the first phase by the project manager [1, 2].

2.4 Project Implementation

The project implementation assessment is focused on reconstruction of the software creation process and a comparison of the simulation's effects to the real thing. The evaluation principle focuses on the treatment of simulation resulting as a benchmark to compare the actual process as such results will not be influenced by real-world delays or difficulties. Comparing the model process with the actual project process enables an examination of the completion of the project, which means that the project is in the correct direction and potential errors are identified. [1-3]

2.5 Organization Capability and Client Involvement

This relates to the use of experience, along with the circumstances that enable teams to execute their tasks effectively. A greatly competent group enables rapid delivery of functioning software that meets the needs of the customer. In addition to technical competence and experience, a range of other qualities have been used, such as the enthusiasm and dedication of team members, flexible, experienced managers with flexible administrative technique, and to provide the project team with sufficient professional preparation [33,34].

Customer participation represents customer representative-company experiences during the project. It has been stated that the extent of client involvement is firmly linked to software project performance, which makes it favorable with greater client participation [33]. The participation of the customers may also benefit from the project for customer satisfaction, and the first concept of an agile manifesto strongly promotes their satisfaction [34, 35]. This aspect is characterized by customer loyalty, project authority and an excellent partnership with the project organization.

3. Research Methodology

The case studies dataset contains all project records relating to the preparation and control of projects. In order to gather data, we used written documentation and semi-structured interview techniques. The records were collected primarily from project management and problem monitoring software.

To gather the necessary data, the study used an online survey. The questionnaire has been produced in English, and we have not confined it to expert of a particular organization or area. We used the technical LinkedIn social network to find respondents, looking for keywords “Agile methodologies” "Agile Project Management."

Considering the features of our survey, most respondents attended higher education (around 97.8 percent). The remaining 2.2% are people who concludedhigh school or similar. Regarding occupational status, most citizens (about 94.2%) are economically engaged, while the remaining 5.8% are either unemployed or students. Because respondents already have a good understanding of agile strategies and processes, we assumed that the remaining 43.8 percent have been practicing them for approximately three years. A percentage of 37.2% of study participants had at least five years of agile experience and attained higher levels of maturity as a result of this study. See Table 1.

In this research, to look for and classify primary studies, the snowballing guidelines for systematic reviews suggested by Wohlin [3, 4] were applied. At the beginning, backward and forward snowballing was done to classify additional primary studies. There have been 33 studies split into three classes for this reason. A shared spreadsheet was used to allow the researchers to monitor their progress in order to promote the snowballing process.

Google Scholar quotes have been used to perform forward snowballing and to execute backward snowballing studies reference lists. Additional experiments have been included, some studies are recognized from snowballing forward and some are recognized from snowballing backward.

To extract the data, an Excel spreadsheet was used. The data characteristics that answer our research questions about bibliometrics, management strategies, and challenges have been identified. Before beginning the extraction, two tests were performed to ensure the extraction tool would work the same two times to expand the data file for our test purposes. This study had a direct and broad impact; making us much more certain of our assumptions about the data as well as helping us improves our methods of data extraction. The study conducted a second time, this time consisting of the research in the form of five full-length pilot studies. Lastly, data extraction was performed on 56 primary studies.

Table 1: Demographic data

Status of Education	%	Status of Profession	%	Software Knowledge	%	Software Experience	%
Basic Knowledge	0	Unemployed	1.6	<1 year	2.7	<1 year	9.4
High School	2.2	Employed	94.2	1-2 years	28.1	1-2 years	28.1
Bachelor Degree	53.3	Retired	0	3-5 years	25.2	3-6 years	20.3
Master Degree	44.5	Student	4.2	>5 years	44.0	>5 years	42.2

4. Results and Discussion

In this study, the factors determining the growth of software project management were analyzed. The research used an online analysis to gather the important data relating to the preparation and control of projects, quality requirements, high team ability, and customer participation in software projects. The research shows 46.3 percent of the change in project success for software development. The findings show that individual characteristics, team ability, social costums, client engagement, guidance, and knowledge are important factors and have a major effect on the performance of software projects. The study shows that company performance is more effective on the accomplishment of a software development project than client engagement. These suggest that when training and learning are poor, client participation has a more important effect on project performance.

However, besides the people-factors that affect the success of the software project, certain physical consequences are going to be revealed as a means to assist managers make choices. Our analysis shows that clients shall be engage in the project, which can result in a major increase in project risks and consequent failure due to their lack of involvement. We, therefore, emphasize that software projects shall be handled as an effective member of the team by at least one customer representative. The client agent shall be allowed to be part of the project-related judgement like permits, refusals, and to prioritize project requirements.

As far as team capability is concerned, we emphasize that if possible, a project team should be comprised of highly committed specialistswhich may contribute to project achievement. An appropriate professional training, giving more emphasis to agile related approach, should also be offered, ensuring team synchronization. The position of team coordinatorshould be controlled by a person with experience in agile techniques and concepts, and shall again carry on flexible management pattern, fostering and versatility.

Furthermore, the analysis also revealed techniques for handling the quality requirements of software development. The techniques will allow developers to track quality demands through software development induction, classification, consideration, priority setting, and preparation. Among the basic studies conducted in this research, 22 studies described the activities of managing quality requirement, 16 studies described the activities of managing security, 8 studies described the activities of managing usability, 4 studies described the activities of managing efficiency, and 2 studies described the activities of managing durability and stability.

It was observed that the method for managing quality requirements comprises prototyping adapted to consistency features, complete experimentation, committed design, and enhanced delivery. It has been found that tools such as descriptions about misuse, descriptions about abusers, and security backlogs are also being suggested to enhance security in software development. We also noticed activities granting usability expert positions, deduction in usability, and backlogs to handle operation in software growth.

The analysis specifically addressed activities among clients and software developers, raised awareness of quality standards via education and training, and conducted

usability training for software designers. We noticed numerous software project management recommendations, such as improving the understanding of team members and undertaking systematic threat analysis and risk evaluation to decide which factors are centered on software development. Some respondents recommended that project management and Scrum teams have a strong perspective on usability and improve team members' usability skills to better handle usability in Scrum.

Time constraint is the second most frequently documented group found in 21 primary studies. A practice for handling consistency criteria is among time management. Moreover, current agile software development techniques are incompatible with brief iteration cycles, making it difficult to implement quality criteria without losing time and expense.

Limitations to the appropriate testing criteria are the third most frequently evident group of issues. Methodical waysto evaluating these criteria are lacking. For example, software development groups may be missing structured protocols and ground rules for examiningdefinitive specifications, such as reliability, protection, and usability.

A further type of challenge is the lack of understanding of quality specifications by customers in software development. Some agile clients are lacking sufficient expertise to enumerate and convey their demands. For example, consumers cannot recognize the subjective consistency of the structure, like maintenance, portability, reusability.

Accordingly, giving more priority to business benefits especially in terms of specifications and growth goals might results in making the management of a software project being difficult. Management teams generallydonot take quality specifications like security into account and prefer to prioritize targets for prospective growth. On the other hand, if agile software development teams concentrate solely on prioritizing business profits, and failed to consider the necessary specifications.This might bring unnecessary withhold in implementation, thus impacting the project's success.

Budget constraints relating to expense and time contribute to difficulties in maintaining the quality specifications of software development. The analysis recorded difficulties in managing software as a result of an inadequate budget allocated for particular requirements. For example, when clients refuse to earmark a safety budget as results of lack of awareness.

Finally, the analysis indicates that quality criteria, performance, security, and usability gained more consideration in software project. We noticed that there are more approaches and strategies than models, instructions, and guidance. The findings reflect the lack of quality control tools and guidance. These provide guidelines for testing, defining, and prioritizing quality criteria to help control software development. Project managers and developers should learn how essential it is to delegate specific roles to avoid difficulties resulting from a lack of clear accountability for the required specifications in the production of software.

5. Conclusion

Project management strives to produce projects on schedule, according to defined criteria and within the planned budget, with the agreed scope and quality. However, the software industry's competitive nature makes it natural that software managers and developers face several crucial decisions in managing the software project. These decisions are taken to enhance processes maturity and product quality with improved planning accuracy and monitoring control. In this study, the factors determining the growth of software project management were analyzed. The study used an online survey to gather the important data relating to the preparation and control of projects, quality requirements, team performance, and client participation in software projects. The findings show that individual characteristics, team ability, social culture, client engagement, training, and guidance are all essential factors and have a major effect on the performance of software projects.

Furthermore, the analysis also revealed techniques for handling the quality requirements of software development. The techniques will allow developers to track quality demands through software development induction, classification, consideration, priority setting, and preparation. It was observed that time constraints, limited testing criteria, customer’s inability to understand quality specifications, Budget constraints, and limited ability to handle quality requirements are the major constraints in software project development.

Finally, the analysis indicates that quality criteria, performance, security, usability, team capability, and customer involvement gained more consideration in the context of software development. Our findings reflect the lack of quality control tools and guidance. These provide guidelines for testing, defining, and prioritizing quality criteria to help control software development. Therefore, it was recommended that project managers and developers should learn how essential it is to delegate specific roles to avoid difficulties resulting from a lack of clear accountability for the required specifications in the production of software.

References

[1] Włodzimierz W. (2020) A hybrid software processes management support model. In: 24rd Intern. Conf. on Knowledge-Based and Intel. Inform. &Engin. Systems, Szczein, Poland, West Pomeranian University of Technology, pp. 2312–2321
[2] Woubshet B, Pertti K, Lidia L, Xavier B, Silverio M.F, Anna M.V, Pilar R, Xavier F and Markku O. (2020) ‘Management of quality requirements in agile and rapid software devpt: A syst. Mapp. study’, Inform. andSoftw.Techno, 123, 106225
[3] Jones C and Bonsignour O. (2011) ‘The economics of software quality, Addison-Wesley Professional
[4] Ramesh B, Cao L and Baskerville R. (2010) ‘Agile requirements engineering practices and challenges: an

empirical study', Information System Journal, 20, 449–480.
[5] Behutiye W.N, Rodríguez P, Oivo M and Tosun A. (2017) 'Analyzing the concept of technical debt in the context of agile software development: A systematic literature review', Information Software Technology, 82, 139–158.
[6] Cristina C and Gleison S. (2019) 'Software project management in high maturity: A systematic literature mapping', Journal of Systems and Software, 148, 56-87
[7] Bharathi, V., Shastry, U and Raj, J. (2012) Bayesian Network Based Bug-fix Effort Prediction Model. In: Mas, A and Mesquida, A and Rout, T and O Connor, RV and Dorling, A. (Ed.), Software Process Improvement and Capability Determination, Communications in Computer and Information Science. pp. 233–238.
[8] Crawford, J.K. (2006) James K., Project management maturity model, 2nd ed. Auerbach Publications
[9] Fenton, N., Marsh, W., Neil, M., Cates, P., Forey, S and Tailor, M. (2004) Making resource decisions for software projects. In: Software Engineering, 2004. ICSE 2004.Proceedings.26th International Conference on. Institute of Electrical and Electronics Engineers Computer Society, Piscataway, NJ 08855-1331, United States, Edinburgh, United Kingdom, pp. 397–406.
[10] Project Management Institute (2017) A guide to the project management body of knowledge (PMBOK guide).
[11] Tamer M.A, Luiz F.C and Danny H. (2019) 'Automatic recall of software lessons learned for software project managers', Info and Softw. Techno, 115, 44–57
[12] Audebert P, Weber R, Aha D.W and Becerra-Fernandez I. (2001) 'Intelligent lessons learned systems', Expert Syst. Appl., 20(1), 17–34 .
[13] Weber R and Aha D. (2002) 'Intelligent delivery of military lessons learned', Decis. Support System, 34 (3), 287–304.
[14] Dülgerler M and Negri M. (2016) Lessons (Really) learned? How to retain project knowledge and avoid recurring nightmares: knowledge management and lessons learned, PMI Global Congress 2016 —EMEA, Project Management Institute.
[15] Li A, (2002) NASA: Better Mechanisms Needed for Sharing Lessons Learned, Government Accountability Office, Washington, DC: U.S.
[16] Chen T.H, Thomas S.W and Hassan A.E. (2016) 'A survey on the use of topic models when mining software repositories', Empirical Software Engineering, 21 (5), 1843–1919
[17] Pinar E and Onur D. (2019) 'A change management model and its application in software development projects', Computer Standards & Interfaces, 66, 103353
[18] Project Management Institute, (2017) A Guide to the Project Management Body of Knowl. (PMBOK® Guide), sixth ed., Project Management Institute, Newtown Square, PA, USA
[19] Tarhan A and Demirors O. (2012) Apply Quantitative Management Now. In: IEEE Software 29 (3) (May-June 2012) pp. 77–85.
[20] Uskarci A and Demirors O. (2017) 'Do staged maturity models result in organization-wide continuous process improvt? Insight from employees', Computer Standards Interfaces, 52, 25–40
[21] Tanriover O and Demiros O. (2015) 'A process capability based assessment model for softw. Workforce.inemerg. softw.organizations', Computer Standards Interfaces, 37, 29–40
[22] Shikha G and Anuradha C. (2020) 'Assessing Cross-Project Technique for Software Maintainability Prediction', Procedia Computer Science, 167, 656–665
[23] Kumar L, Naik DK and Rath SK (2015) 'Validating the effectiveness of object-oriented metrics for predictin maintainability', Procedia Computer Science, 57, 798–806
[24] Shafiabady A, Mahrin MN and Samadi M (2016) Investigation of software maintainability prediction models. In: 2016 18th Inter. Conf. on Adv.Communi. Techno.(ICACT). pp 783–786
[25] Alsolai H (2018) Predicting Software Maintainability in Object-Oriented Systems Using Ensemble Techniques. In: 2018 IEEE Inter. Conf. on Softw. Maintenance and Evolu.(ICSME). pp 716–721
[26] Zighed N, Bounour N and Seriai A-D (2018) 'Comparative Analysis of Object-Oriented Software Maintainability Prediction Models', Found Computer DecisSci, 43, 359–374
[27] Malhotra R and Chug A (2014) 'Application of Group Method of Data Handlin.model for software maintainability prediction using object. Oriented syst', Int J Syst Assur EngManag, 5, 165–173
[28] Jhon M, Francisco J, César P, Félix G and Mario P. (2020) 'Risk management in the software life cycle: A systematic literature review', Computer Standards & Interfaces, 71, 103431
[29] Menezes J, Gusmão C and Moura H. (2019) 'Risk factors in software development projects: a systematic literature review', Software Quality Journal, 27, 1149–117
[30] PMI, (2017) A guide to the project management body of knowledge (PMBOK® guide), sixth ed., Project Management Institute, Inc., Newtown Square, PA USA
[31] AXELOS, (2017) Managing Successful Projects With PRINCE2®, sixth ed., AXELOS
[32] ISO, (2018) ISO 31000: risk management – Guidelines, Geneva, Switzerland
[33] Carlos T, Eduardo J, Tiago O and João V. (2020) 'The factors influencing the success of on-going agile software development projects', International Journal of Project Management, 38, 165–176
[34] Ahimbisibwe, A, Cavana, R and Daellenbach, U. (2015) 'A contingency fit model of critical success factors for software dvlopmnt projects: A comparison of agile and traditional plan-based methodologies', Journal of Enterprise Info Mgmnt, 28 (1), 7–33
[35] Geoghegan, L and Dulewicz, V. (2008) 'Do project managers' leadership competencies con- tribute to project success?', Project MgmntJournal, 39 (4), 58–67 .