CAHIERS MAGELLANES-NS
Volume 06 Issue 1
2024

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

PROJECT DELAY AND COST OVERRUN IN SOFTWARE INDUSTRY IN THE KINGDOM OF SAUDI ARABIA: THE POWERFUL ROLE OF GQM

Ahmad Abdullah N Alghamdi[1] , Dr. Wajdi Al Jedaibi[2], Prof.Abdullah S. Almalaise Alghamdi[3]
1 analghamdi@stu.kau.edu.sa 2 Waljedaibi@kau.edu.sa
3 aalmalaise@kau.edu.sa

Abstract
Background: Given the enormous impact that the software industry has on so many facets of our lives today, software projects are extremely important. Digital transformation is driven across industries by software developments. They facilitate the utilisation of technology by organisations to optimise operations, augment productivity, and provide inventive products and services. To be competitive and adjust to the quickly changing market dynamics of today's world, companies must undergo digital transformation by making full use of software projects.
Objectives: The main aim of the current study is to determine the main causes of project delay and cost overrun in software projects, identify the effects of project delay and cost overrun on software projects, and define the role of cost estimation models in overcoming project delays and cost overruns in these projects.
Results and conclusions:The researcher reviews the literature on causes behind delay and cost overrun and describes the role that can be played by cost estimation models in that process. In light of the reviewed literature, the researcher reaches certain recommendations for the software industry.

KEYWORDS
Project Delay - Cost Overrun - Software Industry - Cost Estimation Models

1 INTRODUCTION

Software is generated using programming languages, which can take several forms. To be successful, a software project must be finished on schedule and under budget, with all features and functionalities as originally stated. However, some software projects fail for a variety of reasons, and some are even terminated before they are done (1). The delay in software projects is considered one of the most important causes of companies incurring high costs and losses that may cause the failure of many systems. Timely project delivery is crucial to avoid negative circumstances associated with delay (2). Project delays have negative impacts on cost overruns, project quality, and stakeholders' satisfaction (3). Accurate time, effort, and cost estimation are crucial issues for a software project's success (4). Prediction of a software project's development effort, timeline, and cost is the goal of estimation to avoid any delay (5, 6, 7). That is to say, Cost overruns and timing delays in software development are regularly occurred. Software cost estimation models and software complexity metrics measures are two prevalent methodologies used by the industry to decrease the recurrence of these difficulties.

1.1 STATEMENT OF THE PROBLEM

To determine how much time and effort software projects will take, cost estimation of software projects is a crucial activity in the software process development cycle. However, there are very few relevant studies evaluating software cost estimation in developing nations, especially in the Kingdom of Saudi Arabia. On one hand, Saudi Arabia is one of the nations that has used cost estimation techniques (8), which means that the current investigation into the causes of inaccurate cost estimations of projects in this country's software development companies is spurred by this issue. On the other hand, information

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

and communication technology (ICT) projects, especially software projects still have high failure rates in the Kingdom of Saudi Arabia (9).
Some researchers confirm that 44% of software projects cost more and take longer than anticipated (5). The problem lies in the fact that little focus has been placed on how project management techniques and instruments have affected the frequency of project delays in the Kingdom (10). In addition, there is a paucity of studies that have dealt with the use of software project management tools in Saudi Arabia (11). Despite the use of the most up-to-date tools and techniques, numerous studies show that software projects still fail (12). Software project cost estimation is a crucial management task. Despite research endeavours, the precision of estimation tools does not appear to be improved (13). Given the prevalence of software in today's products, the large proportion of software projects that fail to meet their deadlines, and the importance of product launches for the financial viability of the product, further research into the problem is needed (14).

1.2 QUESTIONS OF THE STUDY

The questions of the study can be reviewed as follows:

1. What are the main causes of delays in software projects?
2. What are the effects of project delay and cost overrun on software projects?
3. What is the role of cost estimation models in overcoming project delays and cost overruns?

1.3 OBJECTIVES OF THE STUDY

The objectives of the study can be reviewed as follows:

1. Determine the main causes of project delay and cost overrun in software projects.
2. Identify the effects of project delay and cost overrun on software projects.
3. Define the role of cost estimation models in overcoming project delays and cost overruns in software projects.

1.4 SIGNIFICANCE OF THE STUDY

Cost estimation is still a difficult problem that motivates scholars to look into and try out different strategies and approaches. To efficiently manage project budgets, foresee risks, make educated decisions, and control costs, companies can benefit greatly from the insights and guidance offered by cost estimation models. They provide the cornerstone for proactive cost management, which helps businesses better handle project delays and overruns. In the current study, the researcher will try to shed light on the causes of project delay and cost overrun and how to overcome these challenges using cost estimation models.

1.5 DEFINITIONS OF THE STUDY

Delay
A delay is a time overrun or an expansion of the project's completion timeline (15). The researcher procedurally defines it as the inability to deliver a software project on time. It is a critical issue to be discussed in software projects because of its negative effects on cost estimation.
GQM
The Goal/Question/Metric (GQM) approach is a measurement model that allows quantitative evaluation of goal achievement within projects (16). The researcher procedurally defines it as one of the best tools

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

to manage the measurement process, especially when dealing with complex or overlapping variables.

2. LITERATURE REVIEW

The IT sector is complicated, which naturally increases the complexity of initiatives undertaken in this field. Such big initiatives require a high level of efficiency and activity. These projects are prone to delays owing to a variety of difficult scenarios and challenges. Despite considerable scientific research and better management practices, software projects continue to struggle with a variety of challenges such as delays, disputes, productivity loss, poor performance, and coordination. Poor software project performance is a worldwide problem (17).

Software development is divided into stages, each of which necessitates some work (18). That is to say, certain factors affecting development processes in software projects can be reviewed as follows (19):

i. **Organizational factors**: They refer to all factors inside a workplace or entity that have an impact on the people, purpose, work patterns, culture, leadership, and structure of the specific institution.
ii. **Business environment factors:** They refer to internal or external factors that impact a company's or organization's capacity to create and sustain effective customer relationships and are outside the organization's control. For example, competitors, customers, societal economic trends, government operations, and so forth.
iii. **Governance factors:** They refer to the application of all acts involved in an organization's management as well as the effort to regulate the environment in which the organisation operates via continual monitoring and policy. Examples include strategic management, operations, quality control, and corporate strategy.
iv. **Technical factors:** They are elements that have an impact on how an organisation is operated and connected. Tools, processes, skill levels, experiences, and business knowledge are employed within the context of an organisation.

Developing software, which is one of the IT industry's subgroups, is a complex process that is difficult to foresee and estimate. Delays are a key issue in software development. As a result, when these initiatives run late, clients become dissatisfied. Delay is a typical occurrence in software projects; thus, it is critical to understand the causes of delays in these projects and their influence on meeting the objectives (17). Software projects' high risk of cost and schedule overruns has long been a source of concern for the software engineering community. One of the challenges in software project management is making reliable predictions of delays in the context of the constant and rapid changes inherent in software projects (20).

Delays are defined as circumstances that result in an extension of the period scheduled to complete a project (10). A project delay is an unforeseen and unexpected postponement of a project due to an incident or occurrence that prevents the project from starting or continuing. It is the amount of time that causes the project duration to be extended and the delivery of project goals and objectives to be disrupted. Project delays can result in a host of issues, some of which may not be apparent at first. Unexpected delays have a substantial financial impact on projects. When there is a delay, resources are squandered, and the company incurs additional costs. Project delays can also harm your company's reputation among stakeholders and clients. Delays in one client project might induce delays in other projects by tying up resources that could be used elsewhere. Delays in projects can also disrupt the budget, lead the project to deviate from its original path, and, worst of all, result in missed deadlines (17).

In software development projects, delays frequently happen during some operations. Many software development projects miss their dates because project delays are not handled effectively (21). In addition to organisational and managerial issues, several technological factors contribute to software development delays (12). Software development projects frequently experience delays due to a variety

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

of causes, including rework, abandonment, and inaccurate initial estimates (21).
Delays in software projects are typically caused by earlier phases, such as inadequate requirement analysis, bad system design, frequent changes in requirements, or faulty project estimation. Similarly, factors for integration testing delays include inadequate unit testing, system architecture, and requirement analysis (22). A software project delay may result in increased costs as well as a reduction in the scope and quality of product launch activities. Top management, software managers, and sales and marketing managers must recognise the extensive consequences of a software project delay and manage the problem holistically from the perspective of the entire firm, rather than from the views of particular departments (14).
The delay in software projects is considered one of the most important causes of companies incurring high costs and losses that may cause the failure of many systems. There is a real need to reduce the effect of delay causes using comprehensive mitigation measures and advanced tools (23). That is to say, more accurate time and expense estimates for software projects need to be provided by project managers and leaders (24). Worthy here to mention is that forecasting the total costs of a software project is extremely challenging due to the rapid rate of software change (25). In addition to the lack of accuracy associated with this process. Without specific objectives and the relationship between goals and measurements, as well as the interpretation of those measures, an organisation will not be able to regard the measures as meeting its aims (22).
The evaluation point and the ability to quantify the quantitative measurements are the most important part of solving this problem. Understanding the relationships between cause and effect and linking them to delay times will help to understand the delay and discover its aspects, causes, and size of its impact in a clear quantitative manner that can be measured, compared, and followed up on its performance during the stages of the project. It is essential to note that a key problem with delay analysis and the costs associated with it emerges from the absence of a measurement tool or software that is especially suitable for analyzing, registering, categorizing, and identifying the sources of delays and their effects on the different aspects of the project. In addition, using a single software instrument makes it more difficult and inaccurate (26).
The last 40 years have witnessed the introduction of numerous cost estimation models that have been suggested by many researchers. Algorithm-based models and non-algorithm-based models are the two major categories (27). Although it is common for people to use software measurement standards, they are not always simple to implement in a specific environment where the data and context are different. Without explicit goals and a connection between goals and measures, an organisation will not have the opportunity to see the measures as fulfilling its goals (22).
In 1984, the Goal-driven measurement, also known as the Goal-Question-Metric (GQM) was introduced by Dr Victor Basili and his colleagues (28). The GQM method is used to explain the connection between metrics and measurement purposes, which is frequently the same as the project objective. To ascertain whether the project objective (measurement purpose) has been attained, questions are evaluated (29). There are four main phases of the GQM identified by (30, 31) as follows:

i. **The planning phase:** The main goals of the planning phase are to gather all the data needed for an effective introduction and to get team members ready for a measurement programme. A project plan is a crucial product of the planning process.
ii. **The definition phase:** The measurement programme is outlined (goal, questions, metrics, and hypotheses are defined) and documented during this phase.
iii. **The data collection phase:** This phase occurs when actual data collection takes place to identify the data that is essential for analysis and interpretation took place in the following phase.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

iv. **The interpretation phase:** The interpretation phase is when the gathered data is processed in light of the established metrics to produce measurement outcomes, which then enable the evaluation of target achievement. The four phases can be shown in the following figure:

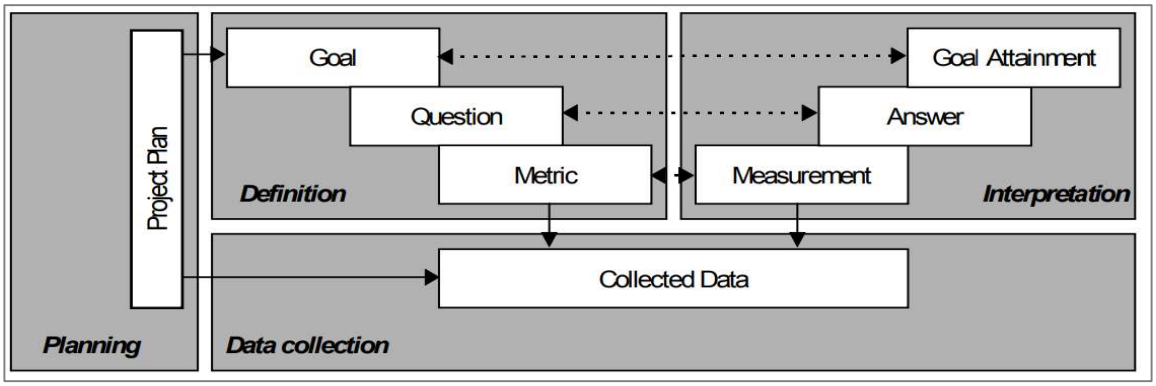

FIGURE 1: Phases of GQM
Source: (31).

To make the business goals and strategies clear, the GQM Strategies add several extensions to the GQM model. Business goals like increasing customer satisfaction, gaining market share, cutting production costs, considering the context, and making assumptions are dealt with by developing strategies. The organisation defines and incorporates sequences of actions required to complete the goals into strategies to accomplish a particular aim. Links are made between every objective and the strategy it promotes. Models that are used to interpret data to identify whether the predetermined goals at all levels are met are called interpretation models. Assumptions must be taken into account while estimating unknowns as they may have impacts on how the results are interpreted (32, 33).

Previous Studies

Bilgaiyan et al. (34) carried out a systematic survey of the literature on effort and cost estimation techniques for software development. The researchers concentrated on studying several development models to reach an accurate measurement of the effort and cost of different software projects based on different development models that can accurately predict delay and calculate the percentage of misbehaviour in total effort, cost estimation, delivery time forecast, and budget. This helped projects adapt to changes in a software project and reach a development model where the customer becomes an active participant in the development thus changes can occur at any stage of development and can be dynamic. Accurate measurements of the effort and cost were able to solve the problems and make the project more flexible through the application of the genetic algorithm (GA), the improvement of particle swarming (PSO), the artificial neural network (ANN), and the fuzzy inference systems (FIS).

Zadeh & Kashef (24) aimed to help the company's executives and management team provide more realistic time and cost estimates for future software projects by examining the link between project complexity and cost/time overrun. The Changepoint database was used to collect sample data for around 50 projects. Statistical approaches were used to define and test the two study hypotheses. Descriptive analysis and regression modelling are part of the quantitative analysis. The findings of the experiments revealed that there was a strong positive linear link between project complexity and cost/time overrun.

Yoon et al. (35) proposed a new effort estimation methodology aimed at agile and iterative development environments that were not suitable for description by traditional prediction methods. The study proposed a detailed development methodology that discussed several structures of these models and also a large group of augmented regression models and neural networks based on machine learning and included a comprehensive case study for Extreme Programming (XP) in two semi-industrial projects.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

The results showed that the proposed stepwise model outperformed traditional estimation techniques significantly in the early stages of development.

Sharma & Singh (36) presented a systematic review of software effort estimation techniques using machine learning. For software effort estimation, the most common machine learning algorithms are Artificial Neural Networks, Fuzzy Logic, Genetic Algorithms, and Regression Trees. The results revealed that the most common methods for effort estimation, are Line of Code (LOC) and Function Point (FP) for software metrics.

Choetkiertikul et al. (37) presented a prediction model for estimating and evaluating projects based on a novel combination of two powerful deep-learning architectures: long-term memory and recurrent highway network. The researchers proposed a comprehensive dataset for effort-based estimation and evaluation that studied 313 releases from 16 open-source projects. It also proposed a comprehensive prediction system based on deep learning for effort estimation. The results revealed that a large part of these improvements has been made possible by the use of deep learning architecture LSTM for modelling textual descriptions.

Kula et al. (38) reviewed the problems of late delivery of projects and the resulting increase in costs and showed that it was a problem resulting from the failure to estimate effort during project planning. Despite the complex systems and technologies involved in software projects, they can be affected by many factors that affect effort estimation and timely delivery. The researchers also identified the factors that affected schedule deviations using a multi-method case study in ING that revealed many organizational, personnel, process, and technical factors. The researchers then structured the findings into a conceptual framework representing the influencing factors and their relationships to on-time delivery. The proposed framework identified and managed delays and risks and designed automated tools to predict schedule overruns and manage a software project.

Agrawal & Chari (39) focused on multitasking projects to examine the effects of very large processes on effort, quality, and cycle time. Using a linear regression model based on data collected from 37 projects and focusing on program size to determine effort, cycle time, and quality on average, a set of developed models was designed that predicted effort and cycle time to be about 12 percent and defects to about 49 percent of the actual values across organizations. It compared favourably with widely used estimation software models such as FPs and COCOMO. The results showed a sharp decrease in variance in the effort, quality, and cycle time which led to relative uniformity in the effort, cycle time, and quality.

Bhattacharya & Neamtiu (40) proposed models for predicting the time to fix errors that used different features to report errors such as the number of developers who participated in fixing the error, the severity of the error, and the number of corrections. The time it would take to fix was estimated, and from it, the study was able to build more accurate and more general models to predict the time to fix errors by using multivariate and univariate regression testing to test the pre-interpretation significance of the existing models. The researcher presented a case study of a bug report from five open-source projects: Eclipse, Chrome, and three Mozilla project products (Firefox, Seamonkey, and Thunderbird). The results revealed three unresolved research issues: (1) determining if prioritising bugs based on bug-opener reputation is advantageous, (2) defining features that are useful in forecasting bug-fix time, and (3) developing bug-fix time prediction models that can be evaluated on real-world data.

Grimstad (41) addressed the issue of improving program cost estimates based on expert judgments and cost uncertainty assessments through better processes, support processes, and better learning/training processes. The emphasis was on expert judgment in most software costing exercises and improving understanding of the causes of estimation inaccuracies in software development projects. It included understanding the impact of the activities and phenomena that occurred before, during, and after the actual development project and examining how personal experiences affected the estimator when deciding to grade.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

Trendowicz et al. (33) proposed an integrated approach to selecting relevant factors affecting software development productivity. Evaluation of Effort and Delay Factors was used to identify the most relevant factors affecting software development productivity by incorporating data analysis and an expert judgment approach via a multi-criteria decision support technique. The study was conducted using a process where the researchers presented a different set of factors compared to individual data and expert-based factor selection methods. The results showed an improvement in the performance of effort estimation in terms of accuracy and an improvement in the estimation performance on the groups of factors that were reduced by the data-based selection method. It was conducted that expert and data-based selection methods identified different (only partially overlapping) combinations of relevant factors.

Abdalkareem et al. (42) addressed the impact of commits on project delays, where it examined the commits of 58 Java projects and identified the commits that were explicitly skipped by developers through manual investigation of 1813 explicit commits and proposed a prototype of the commit that can be Skip CI. The model used a rule-based technology that automatically identified which commits to skip by focusing on unseen datasets extracted from ten projects and demonstrated that the technology can detect and label skip commits CI. The number of commits that are needed to run the CI process can be reduced by 18.16%. A publicly available prototype tool called CI-SKIPPER has been developed that can integrate with any git repository and automatically flag commits that can be skipped.

Lebedeva & Guseva (43) dealt with the factors of late delivery of software projects and cost overruns in the software industry due to deficiencies in estimating effort during project planning as it affected the estimation of effort and delivery on time, which affected schedule deviations in software development. A multi-method case study in ING revealed many organizational, personnel, process, project, and technical factors that were then quantified and statistically modelled using software repository data from 185 teams. Focusing on agent metrics such as project size, number of dependencies, historical delivery performance, and team knowledge, it also addressed hierarchical interactions between factors, which in turn influence technical factors. From it, a conceptual framework was reached that represented the influencing factors and their relationships to delivery on time. From it, it was possible to access the identification and management of delay risks. The researchers also designed automated tools to predict schedule overruns and developed a relational theory for software project management.

Chang et al. (44) proposed an automatic code review tool using static analysis with quality evaluation metrics designed systematically under the GQM methodology and evaluating the three software quality characteristics of the ISO/IEC52060 standard for intelligent applications. It was an automatic code review tool for applications built on an open platform for personal use or public distribution, as it reviewed the code responsible for detecting violations of coding standards and ensuring that best practices were followed. This tool developed a code review model and automatic quality analysis without the need for human intervention to monitor delay elements and interpret their cause and proposal for treatment.

A few gaps can be found in the extensive corpus of research on cost estimate models and their relevance to project delays and cost overruns. These gaps point to areas where more investigation and learning can advance our comprehension of the subject. Although cost estimation models can be used in a variety of businesses, certain features and difficulties may exist in other areas that call for cost estimation techniques unique to that industry. Despite the abundance of cost estimation models available, a more thorough quantitative assessment and comparison of these models concerning their precision, dependability, and suitability is required.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

3. METHODOLOGY OF THE STUDY

To achieve the study objectives and answer its questions, the researcher adopts the quantitative descriptive approach. The researcher has tried to describe the nature of delays in software projects and the different reasons behind software project delays and cost overruns to provide a comprehensive overview of the different factors and methods to overcome them.

4. DISCUSSION

In the current study, the researcher has tried to shed light on the main causes of project delay and cost overrun in software projects. The results revealed that one of the main causes of delays in projects is poor planning (Esteves & Pastor, 2001; Solingen &Berghout, 1999). Scope creep, frequent revisions, and delays in providing the intended software solution can be caused by inadequately stated project scope, ambiguous requirements, and unattainable schedules. Software initiatives may be delayed by a lack of resources, including trained labour, technical infrastructure, and tools. Project progress may be slowed by poor resource allocation, the absence of skilled team members, or reliance on outside sources. Delays can result from setting unrealistic timelines or commitments without taking the project's complexity (Kakaei, 2022), possible hazards, or dependencies into account. Delays may be caused by complicated technical specifications, technological limitations, or a lack of experience with new technology (Tuape & Ayalew, 2019).

Concerning the effects of project delay and cost overrun in software projects, the results showed that delays and cost overruns lead to higher project expenses. Both the project budget and the organization's overall financial resources are strained by these challenges. Project delays may cause deadlines to be missed, which could affect the anticipated dates of delivery or deployment. This may lead to unmet company goals, lost market opportunities, or dissatisfied clients. Consistent cost overruns and project delays can damage an organization's reputation in the marketplace.

In software projects, cost estimation models are essential for preventing delays and cost overruns. These models assist in precisely forecasting and controlling project costs, empowering businesses to decide wisely and take preventative action to reduce risks. Organizations can create budgets that are more realistic and precise by employing these models. Precise budgeting lessens the possibility of cost overruns and the requirement for additional funding by ensuring that sufficient financial resources are allocated to the project (Kakaei, 2022).

5. CONCLUSION

In the software sector worldwide, particularly in the Kingdom of Saudi Arabia, project delays and cost overruns are frequent problems. Rapid technological improvements are defining features of the software industry. It can be difficult for enterprises to keep up with these improvements, which can cause delays in project timeframes and higher expenses. There is frequently a talent shortage as a result of the demand for qualified workers exceeding the supply. Due to the possibility that businesses would need to employ more staff or finance training initiatives, this could cause delays in the completion of projects and raise expenses. Software initiatives in Saudi Arabia may be impacted by the accessibility and dependability of connection and infrastructure. Project delays may result from infrastructure constraints or problems with internet connectivity that impede development and testing.

Organizations can use strong project management techniques, such as careful planning, risk identification and mitigation, and frequent monitoring and control, to avoid delays and cost overruns in the software sector in Saudi Arabia. Investing in programs that improve the capabilities of the software workforce through training and development can help expedite project execution and lower the risk of delays. Organizations can prevent project delays and cost overruns by proactively addressing possible difficulties through the implementation of mitigation techniques and thorough risk assessments.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

Through the use of these tactics and the resolution of the unique obstacles encountered by the software industry in Saudi Arabia, businesses can optimize software project outcomes, curtail expenses, and augment total project success.

REFERENCES

1. Kamuni SK. Study of Factors that Induce Software Project Overrun Time [Ph.D. Thesis]: Cloud State University; 2015.
2. Lopes L, Mañas AV. Delays In IT Projects Due To Failures In The Stakeholders Management. Future Studies Research Journal. 2013;5(2):158 – 85.
3. Khalid M, Khan RA, Khushnood M, Aslam S, Khattak ZZ, Abbas S. An Empirical Analysis of the Influence of Project Governance and Information Technology Governance on Project Delay. Global Business Review. 2022:1–20.
4. Chirra SMR, Reza H. A Survey on Software Cost Estimation Techniques. Journal of Software Engineering and Applications. 2019;12:226–48.
5. Laqrichi S, Marmier F, Gourc D, Nevoux J. Integrating uncertainty in software effort estimation using Bootstrap based Neural Networks. IFAC-PapersOnLine. 2015;48(3):954-9.
6. Leung H, Fan Z. Software Cost Estimation. Handbook of Software Engineering and Knowledge Engineering. USA: University of Pittsburgh Publishing; 2005.
7. Mahboob T, Gull S, Ehsan S, Sikandar B. Predictive Approach towards Software Effort Estimation using Evolutionary Support Vector Machine. IJACSA) International Journal of Advanced Computer Science and Applications. 2017;8(5):446–54.
8. Gumaei A, Almaslukh B, Tagoug N. An empirical study of software cost estimation in Saudi Arabia software industry. International Journal of Soft Computing and Engineering (IJSCE), ISSN. 2015:2231-307.
9. Ebad SA. An exploratory study of ICT projects failure in emerging markets. Journal Of Global Information Technology Management. 2018;21(2):139–60.
10. Alotaibi N, Chong HY, Sutrisna M. Managing Critical Factors Causing Delays in Public Construction Projects in Kingdom of Saudi Arabia. The 6th International Conference on Engineering, Project, and Production Management EPPM2015; 20152015. p. 109–19.
11. AlMobarak N, AlAbdulrahman R, AlHarbi S. The use of software project management tools in Saudi Arabia: an exploratory survey. International Journal of Advanced Computer Science and Applications. 2013;4(7).
12. Mohabuth AQ. A Study of the Identification of the Factors that Lead to Time Delays in Software Development. International Journal of Computer (IJC. 2017;25(1):141–8.
13. Magazinius A, Feldt R. Confirming Distortional Behaviors in Software Cost Estimation Practice. 37th EUROMICRO Conference on Software Engineering and Advanced Applications; 20112011.
14. Rahikkala J, Hyrynsalmi S, Seppanen M, Leppanen V. The Impact of a Delayed Software Project on Product Launch Coordination: A Case Study. 2016 International Conference on Engineering, Technology and Innovation/IEEE International Technology Management Conference (ICE/ITMC; 20162016.
15. Hamzah N, Khoiry MA, Arshad I, Tawil NM, Ani AIC. Cause of Construction Delay - Theoretical Framework. Procedia Engineering. 2011;20(2011):490 – 5.
16. Khamis S, Aljedaibi W. A framework for measuring critical success factors of large-scale software systems. International Journal of Computer Engineering and Technology. 2016;7(6):71-82.
17. Kakaei S. A Hybrid Method Of Cognitive Mapping And Fuzzy Data Envelopment Analysis To Analyze The Delays In Software Projects. Journal Of Applied Intelligent Systems & Information

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

Sciences. 2022;3(2022):63–76.
18. Verma DK. A Systematic Analysis Of Software Cost Estimation Techniques. INTELLECTUALS: An International Refereed Journal of Commerce & Management, IIRJCM. 2016;2(1):31–50.
19. Tuape M, Ayalew Y. Factors affecting development process in small software companies. 2019 IEEE/ACM Symposium on Software Engineering in Africa (SEiA; 20192019.
20. Choetkiertikul M, Dam HK, Tran T, Ghose A. Predicting delays in software projects using networked classification. 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE; 20152015.
21. Zhang W, Yang Y, Liu X, Zhang C, Li X, Xu R, et al. Decision Support for Project Rescheduling to Reduce Software Development Delays based on Ant Colony Optimization. International Journal of Computational Intelligence Systems. 2018;11(2018):894–910.
22. Monden A, Matsumura T, Barker M, Torii K. Customizing GQM Models for Software Project Monitoring. IEICE TRANS INF & SYST. 2012;E95(9):2169–82.
23. Elhusseiny HD, Nosair I, Ezeldin AS. Developing a user plug-in to assess delay causes' impact on construction projects in Egypt. Ain Shams Engineering Journal. 2021;12(2021):3553–68.
24. Zadeh MT, Kashef R. The Impact of IT Projects' Complexity on Cost Overruns and Schedule Delays. ArXiv. 2022:1–6.
25. Briciu C, Filip I, Indries I, editors. Methods for cost estimation in software project management. IOP Conference Series: Materials Science and Engineering; 2016: IOP Publishing.
26. Jalal MP, Yousef E. An Integrated Expert Model for Delay Management in Construction Projects. KICEM Journal of Construction Engineering and Project Management. 2017:1–14.
27. Armario J, Gutiérrez JJ, Alba M, García-García JA, Vitorio J, Escalona MJ. Project Estimation with NDT. Proceedings of the 7th International Conference on Software Paradigm Trends (ICSOFT-2012; 20122012. p. 120–6.
28. Asghari N. Evaluating GQM+ Strategies Framework for Planning Measurement System [Ph.D. Thesis]. Sweden: Blekinge Institute of Technology; 2012.
29. Nakai H, Honda K, Washizaki H, Asoh K, Takahashi K, Ogawa K, et al. Continuous Product-Focused Project Monitoring with Trend Patterns and GQM. 21st Asia-Pacific Software Engineering Conference, Jeju, Korea (South; 20142014.
30. Esteves JM, Pastor JA. Goals/Questions/Metrics Method and SAP Implementation Projects. Catalonia: Polytechnic University of Catalonia; 2001 2001.
31. Van Solingen R, Berghout EW. The Goal/Question/Metric Method: a practical guide for quality improvement of software development: McGraw-Hill; 1999.
32. Basili V, Heidrich J, Münch J, Regardie M, Rombach D, Seaman C, et al. GQM+Strategies: A Comprehensive Methodology for Aligning Business Strategies with Software Measurement2007.
33. Trendowicz A, Kowalczyk M, Heidrich J, Basili VR. GQM+Strategies in a Nutshell. Aligning Organizations Through Measurement. Berlin, Heidelberg: Springer; 2014. p. 9–17.
34. Bilgaiyan S, Sagnika S, Mishra S, Das M. A Systematic Review on Software Cost Estimation in Agile Software Development. Journal of Engineering Science and Technology Review. 2017;10(4):51–64.
35. Yoon KS, Duncan T, Lee SW-Y, Scarloss B, Shapley KL. Reviewing the evidence on how teacher professional development affects student achievement. issues & answers. rel 2007-no. 033. Regional Educational Laboratory Southwest (NJ1). 2007.
36. Sharma P, Singh J. Systematic Literature Review on Software Effort Estimation Using Machine Learning Approaches. 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS; 20172017.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26128036
http://magellanes.com/

37. Choetkiertikul M, Dam HK, Tran T, Pham T, Ghose A, Menzies T. A deep learning model for estimating story points. IEEE Transactions on Software Engineering. 2019;45(7):637–56.
38. Kula E, Greuter E, Deursen AV, Gousios G. Factors Affecting On-Time Delivery in Large-Scale Agile Software Development. IEEE Transactions On Software Engineering. 2022;48(9):3573– 92.
39. Agrawal M, Chari K. Software effort, quality, and cycle time: a study of CMM level 5 projects. IEEE Transactions on Software Engineering. 2007;33(3):145–56.
40. Bhattacharya P, Neamtiu I. Bug-fix time prediction models. Proceeding of the 8th Working Conference on Mining Software Repositories - MSR '11; 20112011.
41. Grimstad S, Jorgensen, M., & Molokken-Ostvold, K. . The clients' impact on effort estimation accuracy in software development projects. In 11th IEEE International Software Metrics Symposium (METRICS'05): Italy; (2005, September).
42. Abdalkareem R, Mujahid S, Shihab E, Rilling J. Which Commits Can Be CI Skipped? IEEE Transactions on Software Engineering. 2021;47(3):448 – 63.
43. Lebedeva AV, Guseva AI, editors. Cognitive maps for risk estimation in software development projects. Biologically Inspired Cognitive Architectures 2019: Proceedings of the Tenth Annual Meeting of the BICA Society 10; 2020: Springer.
44. Chang BM, Son JC, Choi K. A GQM Approach to Evaluation of the Quality of SmartThings Applications Using Static Analysis. KSII Transactions On Internet And Information Systems. 2020;14(6):2354– 76.