VOL 3 (2019) NO 1

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Core Factors for Software Projects Success

Mohammad Ibraigheeth[#], Syed Abdullah Fadzli[#]

[#] *Faculty of Informatics and Computing, Universiti Sultan ZainalAbidin, 21300 Kuala Terengganu, Malaysia*
E-mail: mayyash2010@gmail.com; fadzlihasan@unisza.edu.my

***Abstract*— The successful software project can be defined as the project that meets the planned quality, cost, schedule, and effort. It has been suggested that there are many reasons which might lead to software project success. Several success and failure factors for particular projects were discussed in the literature. This paper aims to identify the most common success factors through reviewing a set of software project reports and case studies. In order to fully understand reasons behind the project success, the reasons behind project failure and the definition of failure itself are discussed. Furthermore, case studies of successful and failed software projects are described. This paper also investigates techniques that have been developed to increase software project success rate and decrease probable failures.**

***Keywords*— Software Project Success, Success Factors, Software Project Failure, Project Performance, Success Rate**

I. INTRODUCTION

According to Standish study [1], the successful software project can be defined as the project that meets its planned objectives, delivered on time, and developed within the estimated budget. However, it is not easy to provide an accurate definition of the project success [2]. Linberg K.R defined four levels to describe the project success: low, successful, high, and exceptionally [3]. A low success means that the project performance (quality, cost, schedule, and effort) is below average compared to the other similar projects in general. A project is successful when it has average performance, and it is high success if it exceeds the average. The project is exceptionally successful if it meets the required quality, budget, schedule and effort.

Fig.1 shows studies from different organizations illustrating the percentage of success software projects in the last two decades [5-7]. With an average of only 35 percent, the software projects were successful.

Many success factors have been identified in the literature. CHAOS report that was released by Standish group organization [4] has described ten software project success factors which are:

- Executive Sponsorship
- Emotional Maturity
- User Involvement
- Optimization
- Skilled Resources
- Standard Architecture Management Environment
- Agile Process
- Modest Execution
- Project Management Expertise
- Clear Business Objectives

II. SOFTWARE PROJECT FAILURE

In order to fully understand and identify software project success core factors, software project failures have to be discussed.

Analyzing the studies which were carried out regarding failed software projects in the literature [10-11] as well as related reports from The Standish Group [12], and the GAO Reports 12-7 and 14-705T [13-14] enabled us to identify that any failed software project almost has one or more of the following characteristics:

- Cancelled or the software product never released into public.
- Desired quality not achieved.
- Lack of fulfilling the user requirements led to giving up the product completely even it is already released.
- Exceeds Budgets: Project is significantly over-budget which exceeds 30% of the estimated cost.
- Missed Deadlines: Project is significantly over-scheduled which exceed 30% of estimated project completion time.

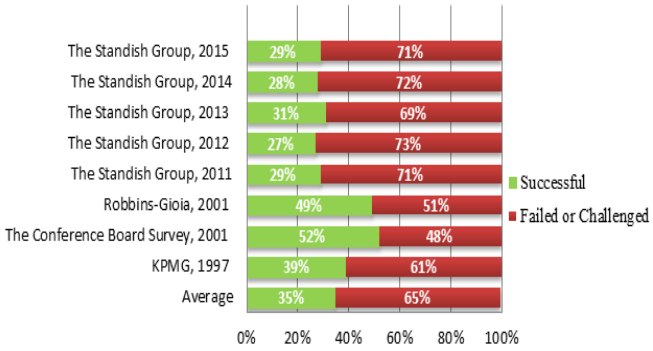

Fig.1 Software project success statistics

There are many factors might affect the success of a software project [8-9]; they are usually related to business, project management and technical concerns. This paper aims to identify the most common factors for software project success.

III. Software Project Failure

In order to fully understand and identify software project success core factors, software project failures have to be discussed.

Analyzing the studies which were carried out regarding failed software projects in the literature [10-11] as well as related reports from The Standish Group [12], and the GAO Reports 12-7 and 14-705T [13-14] enabled us to identify that any failed software project almost has one or more of the following characteristics:

- Cancelled or the software product never released into public.
- Desired quality not achieved.
- Lack of fulfilling the user requirements led to giving up the product completely even it is already released.
- Exceeds Budgets: Project is significantly over-budget which exceeds 30% of the estimated cost.
- Missed Deadlines: Project is significantly over-scheduled which exceed 30% of estimated project completion time.

As seen in different studies such as the Standish Group [12, 14-16], the failing of software projects caused by several factors. Verner J et al [15] analyzes data from 70 failed projects, and specifies the following common failure factors:

- 93 percent of failed projects did not meet the delivery deadlines.
- 81 percent of failed projects were underestimated.
- 76 percent of failed projects had poor risk analysis.
- 73 percent of failed projects had not rewarded the staff for working long hours.
- In 73 percent of failed projects, the project delivered without sufficient requirements information.
- In 73 percent of failed projects, the staff did not have the required experience.

TableI presents failure factors identified in three studies: Ewusi-Mensah[16], The Standish Group [12], and GAO [14].

TABLE I
Failure Factors Identified Throughout Literature Research

Ewusi-Mensah, 2003	**The Standish Group, 2014**	**GAO, 2014**
Unrealistic project objectives.	Incomplete Requirements	Lack of stakeholder's engagement
Inappropriate project-team composition.	Users are not Involved	Project team did not have related knowledge and skills
Problems in Project management and control	Insufficient Resources	Lack of department support
Team technical Problems	Unrealistic Expectations	Lack of stakeholder and end user involvement in requirements phase,
Problematic technology base/infrastructure	Lack of Executive Support	End user did not participate in project testing.
Changing requirements	Requirements and Specifications changing	Inconsistency and instability of Government and contractor staff
Insufficient executive support	Lack of Planning	Poor requirements prioritization.
Users are uninvolved	Didn't Need It Any Longer	Lack of communication between program officer and contractor
Cost and schedule overruns	Poor IT Management	Insufficient funding.

IV. Examples of Successful and Failed Software Projects

Occupational Health Record-keeping System (OHRS) is one example of successful software projects [13]. OHRS was developed to serve as electronic recording system specifically for Veteran's Affairs (VA) employees. OHRS enables the end users to monitor and collect clinical data about its employees and generate reports. Additionally, OHRS provides physicians the ability to document some health issues. Furthermore, OHRS can be used to allocate appropriate staff to patient care assignments. Among other things, OHRS can identify whether a patient has received a certain vaccine.

OHRS project officials stated that there were representatives of the end user involved in requirement development, project status meetings, daily functionality testing, and other activities, and they recognize a set of system requirements and the system features were approved prior to final release. Therefore, the representatives were treated as part of the project development team rather than customers only involved at the beginning and the end of the project. They also noted that the involvement of leadership was critical to project success.

OHRS critical success factors include:

- Partnership between project development team and representatives of end users
- Stakeholders engagements
- Project development staff had sufficient skills.

An example of failed software project is the automated airport baggage handling system of Denver international airport (DIA) [31]. According to Calleam Consulting [32], this project was planned to be the largest automated baggage system. The underestimated complexity was one of the main reasons behind the failure of the DIA project. The airport contains about 100 baggage pickup points, and an enough number of empty carts need to be distributed for each point by using "line balancing" algorithms. The inadequate procedure to determine the suitable number of carts for each pickup point could result in high delay in bags picking up, and therefore, it will affect the overall performance. The following additional reasons were reported:

- Poor project planning, and requirement instability.
- High schedule pressure.
- Poor working environment.
- Problems in project managements.
- Poor project progress tracking.
- Problems in risk management

V. Research Method and Findings

For developing the list of success factors, we analyzed study carried out by Ewusi-Mensah [16] regarding software project failure and success as well as two reports from The Standish Group [4,12], and two reports from the U.S. Government Accountability Office (GAO) [13,14]. The six common factors reported in these five studies are selected to construct our success factors list. The list is presented in Table II. Furthermore, data related to software project failure/success was gathered for 236 projects to find the frequency of each factor among these projects in order to identify which factors have the more influence on the project success. The data was collected using available project reports, surveys, case studies, and IT organization websites. The collected data could be used in further research in developing and training models for project success/failure prediction.

TABLE II
IDENTIFIED SUCCESS FACTORS

Success Factor ID	Success Factor
F1	Realistic Project Objective
F2	Requirement Stability
F3	Adequate Team knowledge and experience
F4	Efficient Technology
F5	User Involvement
F6	Efficient Management

The factors that affect the project success or failure are identified for each project through content reviewing and analysis. Multiple reports and sources were reviewed for some projects to assure the accurate identification. Table III contains a sample of seven successful software projects data. We note that each of the identified success factors is common in at least four projects of this sample, and there is no project success for single factor.

TABLE. 3
COMMONLY IDENTIFIED CRITICAL SUCCESS FACTORS ACROSS SEVEN SUCCESSFUL IT PROJECTS

Critical Success Factors	Project ID						
	P1	P2	P3	P4	P5	P6	P7
F1	X	X	X	X	X	X	X
F2	X	X		X	X		
F3	X		X	X	X	X	X
F4	X	X	X	X	X		
F5		X	X	X	X		X
F6	X		X	X			X

Fig.2 provides a graph showing the frequency of success factors for our data which contains 236 software projects. The median number of projects for each factor is 103; with minimum of 72 and maximum of 131. For the investigated projects, the better project had significantly more success factors. We note that there is no successful project had just one success factor; all projects had multiple factors that led to their success.

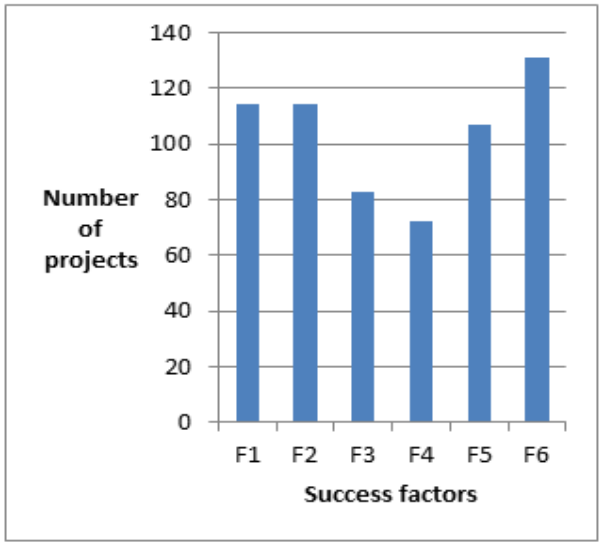

Fig.2 Number of projects versus success factors

The frequent of success factors for all 236 software projects are:

1) The efficient project management and support (F6); 56% of all projects.

2) The realistic project scope and objectives (F1); 48% of all projects.

3) The stability of project requirements (F2); 48% of all projects.

4) The user involvement (F5); 45% of all projects.

5) The adequate team knowledge and experience; 35% of all projects.

6) The efficient technology; 31% of all projects.

VI. DISCUSSION

Reviewing and analyzing different studies in the literature [10-14] as well as related case studies and reports for successful and failed software projects [17-25] enabled us to describe the identified success factors as a following:

A. Realistic Project Objectives

One of the major goals of the software engineering process is to deliver a product that meets the required quality on time and within a budget. Inability to specify the project goals and objectives is one of reasons that affect the success of software project [26]. The Standish group lists "incomplete requirements" and "unrealistic expectations" as a failure causes [12]. However this factor contributes to failure in most of the projects studied in this research, and has a critical effect across all stages of project development process.

B. Requirements Stability

The success of the software project begins with clear requirements definition. Requirement instability or the continuous changing in requirements and specifications can lead to project failure. Unstable requirements can increase the schedules delay and the project cost. Each time the requirement changes, it may need to revise the original project objectives and goals. Each revision could critically affect the project outcome, and can increase the project development complexity. Requirement stability is one of the core success factors identified in this research.

C. Adequate Team Knowledge and Experience

This factor is concerned with the level of experience of the project team with the application domain. This factor leads to increase the schedule delay as the project team requires additional time to understand and be able to use technology. Insufficient team knowledge and experience can result in more risks, as the project team lacks the ability to solve the problems that they face during software project development. This factor also affects the requirements understanding, and therefore it will affect the final product quality.

D. Efficient Technology

This factor is concerned with the ability of the available technology infrastructure to satisfy the project development needs. However, inefficient technology is problematic, because it can be incompatible with developing high quality product. Alternatively, the "new technology" will add another delay to the project schedule, as the project team needs more time to learn and apply this technology in software project development.

E. User Involvement

Insufficient user involvement and commitment in requirements phase can lead to lack of proper understanding of project requirements, and therefore can lead to incomplete and inconsistent requirements that do not meet the user expectations. The sufficient feedback from the users throughout the software project development process is also identified as a major success factor.

F. Efficient Management

Efficient project management is identified in this study as one of the major success factors because it can lead to efficient use of resources, and efficient financial and time control. It enables the team to achieve expected performance goals on time and within estimated budget.

The department support and involvement during project development process is a major factor of project success. The tracking of project progress by corporate leaderships and the sufficient participation in making the critical decisions are common reasons behind project success.

The lack of the above factors or some of them could lead to additional project cost and delay, and thus the failure of the software project. For example, each time the requirements changed, the system design must be changed, and therefore, all subsequent work will be affected, and this will increase the project estimated cost and could add additional delay to the estimated schedule.

VII. IMPROVING SOFTWARE PROJECT SUCCESS RATE

There are number of techniques have been conducted to improve software project success rate and avoid a probable failures. In this paper, some of these techniques are presented. A framework presented by Ewusi-Mensah [16] aimed to identify the impact of different failure factors on the SDLC stages. The empirically based study defines the reasons behind these factors and how they can be prevented. Case studies used to determine these factors such as the IRS's tax system and the automated baggage handling system of Denver international airport. This study claims that identifying and understanding factors that lead to software failure can help in improving software project success rate, and in avoiding the project failures and abandonments.

A questionnaire based approach which was developed by Takagi et al. [27] through investigating different risk management models to determine core risk factors. The questionnaire classified these risk factors into five viewpoints:

- Requirements viewpoint which include factors related to requirement understanding and consistency.
- Estimation viewpoint which include factors related to estimation awareness and methods.
- Planning viewpoint which include factors related to project scheduling, activities, and the staff commitment.
- Team organization viewpoint which include factors related to staff skills, and project resources allocation.

- Management activities which include factors related to project monitoring, and resource managements.

Using resulting data, the software projects are classified into "confused" or "not confused". A logistic regression model is used to characterize the confused projects, and to predict if the software project is risky or not risky.

Verner et al. [15] investigated a number of failed projects to determine the factors behind project failing. A data from 70 failed projects was analyzed to identify the software project failure factors. They also developed questionnaire based on literature, and discussions with software developers to investigate the developers' opinions to identify the factors that affect project outcomes.

Another model proposed by Reyes et al. to identify and analyze the risk factors that have more influence on software project outcomes in order to calculate the probability of project success, taking into consideration the project cost in order to develop an effective project resources allocation which can increase the probability of the project success [28]. The model classified the risk into seven categories, and it used 88 questions based on these categories. A genetic algorithm (GA) is applied on six predictive models developed with past project data. These models are developed using Bayesian classifiers. A unique example is defines as a case for each model based on the projects used to develop these models. According to literature or to expertise knowledge, initial costs that associated with variables that affect the project success probability are selected. GA is executed to find a satisfactory solution to the problem given by the chromosome with highest population fitness value. The chromosome information is used as indication to the most risk factors that affect the project success. An improved success probability at reasonable cost can be obtained by controlling and managing these factors. The authors suggested a strategy for effective resources allocation to get high success rate with minimum cost. The developed model is to identify and control the risks that affect the project success.

Wang et al. developed a predictive model based on Bayesian Network in order to predict the software projects outcome through prediction and controlling the variance in estimated project schedule [29]. The model was proposed to help the project decision makers to predict the probability of variance in estimated project schedule and to provide them with guidelines to improve the project success probability. Factors that significantly affect the schedule variance were identified by developing a sensitivity analysis. The Bayesian Network approach is used because of its ability to gather similar data from previous projects and expert knowledge, the uncertainty of the project schedule, the ability to use Bayesian learning in implementing knowledge representations of the schedule factors, and the ability to perform "what-if" analysis to help the project decision makers take decisions to improve the project success probability by identifying the most factors that affect software schedule. The developed model aimed to maximize the opportunity to complete the project on time through project re-planning, resource re-allocation, and schedule variance factors identification.

Lehtinen et al. performed analysis in corporation with four software companies to identify the causes behind failures, and the relationships among them [30]. They performed their research to understand the factors that interconnected with each other in specific process and they could be used as key factors to improve development process, and control the failure causes. The causes of failures are distributed into: people, tasks, method, and environment. Root Cause Analysis (RCA) was used to identify relationships among the factors that cause project failure. The research identified why and where the problems occurred and their relationships in order to develop failure prevention mechanism. A "causal model" is developed based on four cases by analyzing data related to project failure in each case. They suggested diagrams to describe causal relationships among failure causes, and it recommended performing specific analysis for each case, and controlling the causes outside the development area to prevent the software project failure.

VIII. LIMITATIONS

The list of success factors identified in our research is based on software project reports and surveys. Some limitations are recognized for our study. Some of reports and surveys are based on limited number of software projects, and some of these reports are limited to knowledge of the software project developers.

IX. CONCLUSION AND FURTHER WORK

In this paper, a common success factors are identified through reviewing relevant project reports and surveys that described failed and successful software projects. These factors usually led to finish the project within estimated schedule and budget, and thus cause the success of the project.

Our study shows that there is no one reason for software project success; they succeed for several reasons. Different techniques that can help in improving the project success rate are investigated. However, most of these techniques focus on identifying and reducing the impact of factors that can cause the project failure.

In this study, we suggest that the early prediction of software system failures can help in making decisions about possible solutions that could improve success rate. Even though there are many tools proposed to address the software project failures, most of these tools do not perform project failure prediction..

In further research, the identified success (failure) factors will be used to develop prediction models that can be applied early through software development process.

REFERENCES

[1] Standish, G. (1994). The chaos report. *The Standish Group.*

[2] Lewis, J. P. (2000). Project planning, scheduling and control. *McGraw-Hill Companies.*

[3] Linberg, K. R. (1999). Software developer perceptions about software project failure: a case study. Journal of Systems and Software, *49(2-3), 177-192.*

[4] Hastie, S., & Wojewoda, S. (2015). Standish group 2015 chaos report-q&a with jennifer lynch. *Retrieved, 1(15), 2016.*

[5] Robbins-Gioia, L. L. C. (2002). ERP survey results point to need for higher implementation success.

[6] Chowdhury, R., Butler, R. E., & Clarke, S. (2007). *Healthcare IT* project failure: A systems perspective. Journal of Cases on Information Technology *(JCIT), 9(4), 1-15.*

[7] Whittaker, B. (1999). What went wrong? Unsuccessful information technology projects. Information Management & Computer Security, *7(1), 23-30.*

[8] Charette, R. N. (2005). Why software fails. IEEE spectrum, 42(9), 36.

[9] Dalcher, D., & Brodie, L. (2007). Successful IT projects. *Cengage Learning EMEA*.

[10] Flowers, S. (1997). Software Failures: Management Failure: Amazing Stories and Cautionary Tales. *London: John Wiley & Sons.*

[11] Tan, S. (2011). How to Increase Your IT Project Success Rate. Gartner Research.

[12] The Standish Group. (2013). The Chaos Manifesto.*The Standish Group.*

[13] GAO Report (12-7). (2011). Information Technology: Critical Factors Underlying Successful Major Acquisitions, *GAO. Washington, D.C.: Government Accountability Office.*

[14] GAO Report *(14-705T).* (2014). Preliminary Results of Undercover Testing of Enrollment Controls for Health Care Coverage and Consumer Subsidies Provided Under the Act

[15] Verner, J., Sampson, J., &Cerpa, N. (2008, June). What factors lead to software project failure?.In Research Challenges in Information Science, *2008.RCIS 2008. Second International Conference on (pp. 71-80). IEEE.*

[16] Ewusi-Mensah, K. (2003). Software Development Failures: Anatomy of Abandoned Projects. *Cambridge: MIT Press*

[17] Devos, J., Van Landeghem, H., & Deschoolmeester, D. (2008). Outsourced information systems failures in SMEs: a multiple case study. *Electronic Journal of Information System Evaluation, 11(2), 73-84.*

[18] Nelson, R. R. (2005). Project retrospectives: Evaluating project success, failure, and everything in between. *MIS Quarterly Executive, 4(3), 361-372.*

[19] Dalal, S., & Chhillar, R. S. (2012). Case studies of most common and severe types of software system failure. *International Journal of Advanced Research in Computer Science and Software Engineering, 2(8).*

[20] Al-Yaseen, H., & Al-jaghoub, S. (2012). Success and failure of e-learning projects: alignment of vision and reality, change and culture.

[21] www.Reksoft.com

[22] www.calleam.com

[23] Stanciu, V., & Tinca, A. (2013). ERP solutions between success and failure. Accounting and Management Information Systems, *12(4), 626-649.*

[24] Devos, J., Van Landeghem, H., & Deschoolmeester, D. (2008). Outsourced information systems failures in SMEs: a multiple case study. *Electronic Journal of Information System Evaluation, 11(2), 73-84.*

[25] Wende, E., & Philip, T. (2011, January). Instant messenger in offshore outsourced software development projects: experiences from a case study. In System Sciences (HICSS), *2011 44th Hawaii International Conference on (pp. 1-10). IEEE.*

[26] Cole, A. (1995). Runaway projects: cause and effects. Software World, *26(3), 3-5.*

[27] Takagi, Y., Mizuno, O., &Kikuno, T. (2005). An empirical approach to characterizing risky software projects based on logistic regression analysis. Empirical Software Engineering, 10(4), 495-515.

[28] Reyes, F., Cerpa, N., Candia-Véjar, A., & Bardeen, M. (2011). The optimization of success probability for software projects using genetic algorithms. *Journal of Systems and Software, 84(5), 775-785.*

[29] Wang, X., Wu, C., & Ma, L. (2010, July).Software project schedule variance prediction using Bayesian Network.In Advanced Management Science (ICAMS), *2010 IEEE International Conference on (Vol. 2, pp. 26-30).IEEE.*

[30] Lehtinen, T. O., Mäntylä, M. V., Vanhanen, J., Itkonen, J., &Lassenius, C. (2014). Perceived causes of software project failures–an analysis of their relationships. *Information and Software Technology, 56(6), 623-643.*

[31] Dalal, S., & Chhillar, R. S. (2012). Case studies of most common and severe types of software system failure. *International Journal of Advanced Research in Computer Science and Software Engineering, 2(8).*

[32] http://calleam.com

[33] Reyes, F., Cerpa, N., Candia-Véjar, A., & Bardeen, M. (2011). The optimization of success probability for software projects using genetic algorithms. *Journal of Systems and Software, 84(5), 775-785.*