International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

© Global Society of Scientific Research and Researchers

http://ijcjournal.org/

A Study of the Identification of the Factors that Lead to Time Delays in Software Development

Abdool Qaiyum Mohabuth*

University of Mauritius, Reduit, and 80835, Mauritius

Email: a.mohabuth@uom.ac.mu

Abstract

Completing software development projects on time has always been very challenging for software developers. Despite proper planning and use of tools and techniques, project managers often complained that they miss their targets. Rarely software development projects are delivered on time to the satisfaction of clients. Time slips, deadline missed and schedule overruns resulting in serious delays for software delivery. This study aims at identifying the factors which prevent software project to be completed on time. It extracts where software developers experience most of the time delays and assess whether overrunning schedule is distinct to different size of organisation. A survey questionnaire was designed for identifying the causes and frequency of occurrences of delays on software projects. The questionnaire was administered to software developers at small, medium and large software development house. Semi-structured interviews were also conducted with software developers who have experienced on managing software development projects. Findings reveal that software delays are mostly due to managerial problems rather than technical complexities. Coordination, communication, conflicting relationship, unrealistic planning, team complexity are among the factors found that contribute most to time delays. Technical factors are found to be less prominent with the most frequent ones being in accommodating changes, determining requirements and in the testing phase. Besides, it has been observed that there is uniformity across organisations i.e. the small, medium and large development firms face the same issues as regards to the time delays affecting software project.

Keywords: delays; schedule overruns; deadline; delivery; target.

1. Introduction

The execution of software development projects are not always successful, their development is a challenging issue. Software is developed to perform a specific set of functions.

----

* Corresponding author.

Unless it is successful in performing these functionalities effectively, the purpose of the software development will be defeated and questioned. The software should be delivered to the customer on time as scheduled. Today, most of the software industries are concerned with failure and escalation of original budget due to time delay in project implementation. Quite often, a penalty clause is associated with the delivery of the software, as the client is expected to face substantial business setback if they cannot bring change in their system on time. Software delivery delays not only incur cost due to penalty, but also there is always the chance of increased prices of materials and services with time (price escalation), loss of image, and incurring opportunity cost. Software development is complex and software developers are intelligent and deal daily with complex problems which in turn makes the management formula in the organisation more complex [1]. Coping with delays which accrued up down the development stages is very stressful. Software developers are concerned with requirement specification, design, coding and implementation of software and they make use of latest tools, techniques and practices at each stage to successfully deliver project milestones. Yet delivery dates often slip at the end. Numerous studies indicate that software projects continue to fail despite the use of latest tools and techniques. There is adequate literature about the successful factors to complete software project on time, within budget and within the quality requirement standard, but there exist a dearth of literature on the identification of the factors that contribute to most delays on software projects. This paper aims at identifying these time delay factors which prevent software to be delivered on time.

2. Literature Review

Evidence indicates that many software projects fail to deliver on time or budget and do not give value [2,3]. According to [4] nearly two-thirds of software projects do not meet their time and budget goals, and often do not meet their business objectives. A study of 720 software development projects conducted by [5] found that the use of an inappropriate methodology is actually the most critical factor leading to project delivery failure. Reference [6] argued on their sides that the lack of a decision support tool contribute to delays along the stages of development. Over 31.1 per cent of software projects are cancelled before they get completed and 52.7 per cent of the projects are escalated by 198 per cent of their original estimates [7]. Reference [8] found that coordination issues can cause a substantial loss on the developmental speed of project. When problems emege during the course of development, the difficulties of knowing who to contact about what, of initiating contact, and of communicating effectively, led to a number of serious coordination problems. Ineffective communication among team members is also recognised as a factor that can lead to delays. Reference [9] reported that "one of the most salient impressions conveyed by observation was the sheer amount of time each developer spent in informal communication" (p. 41). The developers in their study spent an average of 75 minutes each day in "unplanned interpersonal interaction." In an 8-month study of a medium-sized telecommunications software project, an analysis of time sheets indicated that about 50% of time was spent in "group work" (meetings and unplanned work-related discussions) during the first month, and this level dropped fairly steadily until only about 10% of time was spent in group work during the last month. Conflicting relationship among team members is also a potential source of the cause of time delays. Personal attacks and interpersonal disagreements within groups may cause dissatisfaction and hence decrease the amount of individual effort for completing group tasks [10,11]. Design activities, in particular, seemed to require a very large proportion of collaborative work (over 50% in all but one 4-week period), in contrast to the relatively solitary activities of coding and testing. On the other hand, according to [12] the leading source of time delays in software development is the amount of rework to be carried out in the redesign and re-coding due to the need to accommodate changes in requirements, changes in the interfaces. Team size in software development is also a sensitive issue which may speed or relents effort on software project. Adding more human resources on project may not necessarily speed up project but may expand the time delays. Reference [13] made observations that high-performance work teams may be surprised by the "more is less" effect that is adding people relents project speed instead of catching up lateness. [14] further found that the larger the team of development is, the more unproductive they are resulting in slower completion of task. They reported that larger teams diminish productivity because of inefficiencies created by the difficulty of communicating within a large number of people. This equates to Brooks law who argued that communication demands increase in proportion to the square of the size of the team [15].

3. Research Methodology

To extract data about the causes of delays on software projects and where they are more prominent during software development, semi-structured interviews were first conducted with a three experienced software practitioners. A list of the most probable causes that make software projects fall behind schedule was identified. Discussions were also held about the stages where most of the time delays were encountered during development. Following which, a set of questionnaires were developed which was administered to a set of 40 software developers involving the small, medium and large size software houses. The questionnaire contained section that included Likert scale questions to rate the degree of occurrences of the causes of delays with rating scale 1-5 (1: Very Rare (VR), 2: Rare (R), 3:Neutral (N), 4:Often (O), 5:Very Often (VO)). The extent at which these delays affected completion time were also measured by rating scale 1-5 (1: Not at all, 2: A little, 3: Neutral, 4: Much, 5: Very Much). The stages at which most delays were experienced were ranked. Questions were also set about the consequences of delays in one phase to the subsequent phases. The frequency software developers have been able to meet targets and deadlines were also investigated. Open-ended questions were also provided for enabling respondents to share their experiences and insights about time delays. The questionnaire was stratified over the three types of software development firms Small (S), Medium (M) and Large (L) for ensuring a good representation of practitioners across the different organisations.

4. Results and Discussion

The quantitative data were analysed using SPSS version 21. 70% software developers confirmed that they have very rarely or rarely been able to meet the timeline scheduled for project completion, while 22.5% attested that they have never been able to complete their project in time despite the fact that 82.5% rated the importance of on time completion as high and very high. The task of delivering software products is therefore seen to remain challenging. These findings have close similarity to previous research works as detailed in the literature review above. Table 1 shows the mean rank of the degree of occurrences of identified items that causes most delays during system development. Factors such as ‘coordination problem’, ‘conflict among team members, ‘poor communication’, ‘unrealistic plan’, ‘team size complexity’, and ‘accommodating changes’ have been more highly rated (mean > 4) than ‘inappropriate supporting tools’, ‘inadequate reuse of codes’ or ‘poor testing strategies and methods' (mean < 4). These reveal that the causes of delays involved more managerial factors as compared to technical factors. Management is seen to focus more on the technical issues rather than on the managerial factors to gain control over project time. Coordination, conflicting relationships and communication problems were often underestimated and many times they were discarded and yet these are found to be the most obvious factors contributing to delays. In fact, [16] found that personal attacks among group members causes a decrease in concentration, and a waste of effort on quarrelling which relents the outcomes of software development. This was also supported by [17] who found that cooperation, coordination and integration are critical factors for software projects to succeed.

Table 1: Degree of occurrences on projects

Factors	Mean Rank	Standard Deviation
conflict arising among team members	4.42	0.133
poor communication among team members	4.10	0.147
coordination problem	4.52	0.101
unrealistic project plan	4.35	0.122
team size complexity	4.12	0.158
requirement specification poorly defined	4.10	0.167
client resistance in signing off contract	3.00	0.189
priority shift	2.80	0.193
poor design structure-poor system specification	2.32	0.121
inappropriate development platform chosen	2.08	0.090
inappropriate supporting tools used	1.88	0.089
inadequate reuse of codes	2.65	0.177
skipping phases of development	2.22	0.162
poor testing strategies & methods	3.62	0.171
accommodating changes	4.38	0.132
lateness in software acquisition	1.92	0.121
lateness in hardware device acquisition	2.08	0.191

Almost similar observations were made as regards to the extent these factors affected project completion, 'coordination problem', 'poor communication' 'accommodating changes' and 'conflict among team members' were found to be more pronounced as compared to factors such as 'inappropriate support tools' or 'lateness in software and hardware acquisition' as shown in Table 2. The means and standard deviation of the different factors under test are illustrated in the table. Again it seen that the extent managerial factors affect project completion exceed the technical ones. Among the technical factors 'accommodating changes' is found to carry a mean of 4.42. In fact, almost all practitioners agree that accommodating changes impacted much on project completion time. Even in the initial semi-structured interviews, the experienced developers mentioned that accommodating changes causes quite a lot of disruption in the phases of development. The leading source of primary time delays in software development after the managerial factors is found to be the changes that need to be accommodated. Changes particularly in requirements have a ripple effect in redesign, recoding and retesting. It was also observed that most of delays are experienced during the stages where customer requirements are determined and the testing and debugging phase. Almost 68% ranked 'determining customer requirements' as the stage of highest time delays. Most practitioners claimed that very often they had to return to this initial stage to accommodate changes which require lot of rework to get back on track.

Table 2: Extent affecting completion time

Factors	Mean Rank	Standard Deviation
conflict arising among team members	4.52	0.080
poor communication among team members	4.42	0.107
coordination problem	4.28	0.095
unrealistic project plan	4.15	0.137
team size complexity	3.90	0.178
requirement specification poorly defined	3.48	0.203
client resistance in signing off contract	2.40	0.100
priority shift	2.30	0.114
poor design structure-poor system specification	2.10	0.106
inappropriate development platform chosen	1.70	0.089
inappropriate supporting tools used	1.68	0.900
inadequate reuse of codes	2.28	0.164
skipping phases of development	1.85	0.116
poor testing strategies & methods	3.85	0.160
accommodating changes	4.42	0.154
lateness in software acquisition	1.65	0.092
lateness in hardware device acquisition	2.28	0.139

In addition, there was also need to assess the variability in the time delay factors identified by different categories of software houses. This was achieved by considering the following hypothesis which was tested to investigate whether the delays experienced differ from the small, medium and large software houses.

Hypothesis

H_o: There was no significant difference in the time delay factors affecting software development experienced by the small, medium and large enterprises

H_1: There was a significant difference in the time delay factors affecting software development experienced by the small, medium and large enterprises

Kruskal Wallis which is a non-parametric test was used as the data was not found to be normal. An index named DI was created to measure the delaying indicators.

$H_{o:}\ \mu_{small} = \mu_{medium=}\ \mu_{large}$

$H_{1:}\ \mu_{small} \neq \mu_{medium} \neq \mu_{large}$

where μ_{small} = median value of DI for developers from the small enterprise; μ_{medium} = median value of DI for developers from the medium enterprise and μ_{large} = median value of DI for developers from the large enterprise

Table 3 shows that the mean ranks for the different size of enterprises do not differ much ranging from 18.89 to 22.7. Kruskal Wallis result revealed that p-value = 0.701 > 0.05 as illustrated in Table 4, indicates that there was no significant difference in the time delay factors in software development at the different types of enterprises. This provides further evidence that the software houses irrespective of size experience similar occurrences of time delays for software development. The managerial factors identified above (staffing, coordination, communication etc) as the common sources of time delays are therefore independent factors irrespective of sizes of organisation.

Table 3: Ranks

	Company	N	Mean Rank
DI	Small size	14	20.21
	Medium Size	14	18.89
	Large Size	12	22.71
	Total	40	

Table 4: Test Statistics

	D Indicator
Chi-Square	.709
df	2
Asymp. Sig.	.701

5. Conclusion and Recommendation

The study demonstrates that the reasons for delays in software development are mostly related to organisational and managerial problem rather than being purely technical in nature. Software developers make use of tools and techniques across the phases of development which is seen to rarely lead to time delays, instead the human aspects are seen to carry more weights on the health of software projects. In fact software development has been termed as a labour and knowledge-intensive task. To date much care has been given for the knowledgeable part in terms of providing the latest skills to the practitioners with the latest available tools and products on the market to facilitate the task of software development. However, little consideration has been given about the managerial factors such as communication, coordination and conflict that are seen to contribute much to software delays. Based on the findings in this study, equal consideration should be given for the aspects of communication, coordination and conflicting relationship on software project. People on system development work in team with different background and beliefs, the chances of disagreement, misunderstanding and conflict are real and should not be underestimated. Appropriate time must be devoted to avoid the sources of conflict and formal communication plan should be established at the initial stages of project to prevent these parameters impacted on completion time. As regards to the frequency of changes that impacted much on completion time, it is recommended that there should be better customer specification initially. Time spent for learning what the customer wants and needs upfront should reduce the frequency of specification changes later in system development. The result of this study provides direction about the factors causing delays on software project. The study should be replicated by considering larger sample size with more specificity about the type of software developed by software houses. Further studies could look into ways of mitigating time delays during software development.

References

[1] J. Reel. “Critical success factors in software projects.” IEEE software, vol. 16, no. 3, pp. 18-23, 1999.

[2] PMI (2013a). Pulse of the Profession Report 2013. Project Management Institute Inc. [On-line]. Available: www.pmi.org/Knowledge-Center/Pulse/~/media/PDF/Business-Solutions/PMIPulse%20Report-2013Mar4.ashx [May 05, 2017]

[3] KPMG (2013). Project Management Survey Report 2013. KPMG. [On-line]. Available:

www.kpmg.com/NZ/en/IssuesAndInsights/ArticlesPublications/Documents/KPMG-Project-Management-Survey-2013.pdf [May 02, 2017]

[4] A. Shenhar. “Unleashing the power of project management.” Industrial Management, vol. 50 no. 1, pp. 14-18, 2008.

[5] A. Tiwana and M. Keil. “The one-minute risk assessment tool.” Communications of the ACM, vol. 47, no, 11, pp. 73-77, 2004.

[6] D. Howell. C. Windahl and R. Seidel. “A project contingency framework based on uncertainty and its consequences”, International Journal of Project Management, vol. 28 no. 3, pp. 256-264, 2010.

[7] C. Samantra. S. Datta. S. Mahapatra and B. Debata. “Interpretive structural modelling of critical risk factors in software engineering project.” Benchmarking: An International Journal, vol. 23, no. 1, pp. 2-24, 2016

[8] J. Herbsleb. “Distance, Dependencies, and Delay in a Global Collaboration,” CSCW 2000, Philadelphia, 2000

[9] D. Perry. N. Staudenmayer and L. Votta. “Understanding and improving time usage in software development.”, Software Process, vol. 5, pp. 111-135, 1995.

[10] A. Amason and D. Schweiger. “Resolving the paradox of conflict, strategic decision making, and organizational performance.” International Journal of Conflict Management, vol. 5, no. 3, pp. 239-253, 1994.

[11] K. Jehn. “A multimethod examination of the benefits and detriments of intragroup conflict.”. Administrative science quarterly, pp. 256-282, 1995.

[12] J. Blackburn. G. Scudder and L. Van Wassenhove. “Improving speed and productivity of software development: a global survey of software developers.” IEEE Transactions on Software Engineering, vol. 22, no. 12, pp. 875-885, 1996.

[13] M. Huang. L. Lee and A. Kao. “Balancing performance measures for information security management.” Industrial Management & Data Systems, vol. 106, no. 2, pp. 242-55, 2006.

[14] J. Blackburn. G. Scudder and L. Van Wassenhove. “Concurrent software development.” Communications of the ACM, vol. 43, no, 11, 2000.

[15] J. Blackburn. M. Lapré and L. Van Wassenhove. “Brooks' law revisited: improving software productivity by managing complexity”, 2006.

[16] L. Ting-Peng. L. Chih-Chung. L. Tse-Min and L Binshan. "Effect of team diversity on software project performance." Industrial Management & Data Systems, vol. 107, no. 5, pp.636-653, 2007.

[17] M. Amberg and M. Wiener. “Analysis of critical success factors for offshore software development projects-a German perspective.” ISOneWorld 2006, pp. 19-21, 2006.