

Full Length Research Paper

Critical success factors for software projects: A comparative study

Mohd Hairul Nizam Nasir^{1*} and Shamsul Sahibuddin²

¹Faculty of Computer Science and Information Technology, University of Malaya, Malaysia.

²Advanced Informatics School, Universiti Teknologi Malaysia, Malaysia.

Accepted 16 February, 2011

Although there have been studies completed on the critical success factors of software projects, these studies all have been specific to one particular country. There has been no comprehensive study reporting on different project sizes in various domains and in multiple countries. We present our extensive literature survey of critical success factors that impact software projects. Forty-three articles from the years 1990 to 2010 were found to be significant contributions that could be analysed in order to develop a list of critical factors that specifically affect the success of software projects. The method of content analysis and frequency analysis was adopted. Twenty-six critical success factors were found to be related to software project success. We suggest that organisation or project manager is attentive to control the top five critical factors to drive towards project success since the percentage of frequency of occurrences for each is more than 50%. Also, it appears that non-technical factors (94%) dominated over technical factors (6%). In a result unique to our study compared with previous one, we found that the factors of clear and frozen requirements, realistic estimation of the schedule and budget, along with a competent project manager are the five most critical success factors of software projects.

Key words: Critical success factors, software project management, comparative study, project failure, project success.

INTRODUCTION

The term 'Software Engineering' was first coined at the first NATO Software Engineering Conference in Germany in 1968 (Naur and Randell, 1969), and there was widespread consensus that there were problems with software development and maintenance. These problems were later discussed by Brooks (1975, 1987, 1995), and he concluded that there is no silver bullet to overcome this problem. The term 'Software Crisis' emerged to describe the software industry's inability to provide customers with high quality products within schedule and under budget. Hardware costs were dropping while software costs were rising rapidly. Major computer system projects were sometimes years late, and the resulting software was unreliable, hard to maintain and performed poorly.

Since the 1980s, in the medical field for example, computers have been designed to help people, and most

of the time they do. However, in the case of Therac-25, computer errors could be fatal. Between 1985 and 1987, two people died and four others were seriously injured after they received massive radiation beamed via Therac-25 radiation therapy machines. Successful investigations revealed that defective software was among the various factors leading to this accident (Leveson and Turner, 1993). Another example is the delay of over 16 months in the opening of Denver International Airport and the over 100 million dollars in excess of the budget in the airport's construction cost (Swartz, 1996). "One main reason for the delay and overrun was the presence of major bugs in the baggage handling control software" (Glass, 1998). The explosion of the European Space Agency rocket, Ariane 5, 40 seconds after lift-off in June 1996, is another example of an accident caused by software defects. The rocket and the four scientific satellites on board, which total cost about 2.5 billion dollars, were completely destroyed. In-depth investigation uncovered that "specification and design errors in the software, and poor analysis and testing of the failed subsystems, were to

*Corresponding author. E-mail: hairulnizam@um.edu.my.

Table 1. Standish software project performance over a decade.

Benchmark/year	1994	1996	1998	2000	2004	2006	2008
Succeeded (%)	16	27	26	28	29	35	32
Challenged (%)	53	33	46	49	53	46	44
Failed (%)	31	40	28	23	18	19	24

blame" (Ariane 501 Inquiry Board, 1996).

Even today, this situation has not changed much. Software development projects are known for being completed far over budget and behind schedule (Gray and Larson, 2008). In the United States (US), a survey conducted by The Standish Group (1995) in 1994, which reported data from several thousand IT projects, revealed a success rate of only 16% of software projects. Meanwhile, 31% of projects failed while the remaining 53% had cost overruns, time overruns and impaired functionality. Of these, the average cost overrun was 189%, and the average time overrun was 222%. A recent report by The Standish Group showed a slight improvement, yet the figure remained troublesome, with a success rate of less than 40%. Table 1 tracks the progress of Standish software project performance over a decade. These figures were published by a number of different authors in various journals and white papers (Standish Group, 1995, 1999, 2001, 2010; Othman et al., 2010; Hass, 2007; Eveleens and Verhoef, 2010; Humphrey, 2005; Hartmann, 2006). It is important to note, however, that some researchers have argued and expressed concerns regarding the reliability of the figures (Jorgensen and Molokken, 1994), some felt that the figures do not reflect reality (Glass, 2005, 2006), and some questioned their validity (Eveleens and Verhoef, 2010).

Another study, conducted in the United Kingdom (UK) by Oxford University in collaboration with Computer Weekly in 2003, reported that only 16% of the 421 IT projects examined were finished on time within the estimated budget and with the agreed functionalities, only 55% of projects were completed on time, and 41% were completed within the agreed budget (Sauer and Cuthbertson, 2003). Similarly, another UK survey published by the Royal Academy of Engineering and British Computer Society (Taylor, 2000) reported that only three out of the more than 500 development projects assessed met the survey's criteria for success. Although the success rate of software projects is improving each year, the general trend still suggests that software projects are hard to manage. This is also supported by the research conducted by Simons et al. (2003), where they found that the issues surrounding software engineering identified over 40 years ago, during the NATO Software Engineering Conference, remain unresolved today.

Critical success factors are factors that, if addressed appropriately, will significantly improve the chances of

project success (Pinto and Rouhiainen, 2001). Over the past several decades, numerous research studies (Pinto and Mantel, 1990; Belassi and Tukel, 1996; Tukel and Rom, 2001; White and Fortune, 2002) have been performed in the area of project management to identify critical factors that influence the success and/or failure of projects. However, the critical factors are usually identified in general for projects in various industries, such as engineering, manufacturing, construction and training rather than being focused on software development or IT projects. Software has certain unique characteristics (Brooks, 1995; Galin, 2004; Jain, 2008; Fairley, 2009) that make software development projects differ from other typical engineering projects. Also, most researchers agree that there are differences in project management among different industry types (Cooke-Davies and Arzymanow, 2002; Ibbs and Kwak, 2000; Zwikael and Globerson, 2006), Dvir et al. (1998) suggest that project success factors are not universal to all projects. Thus, the critical success factors identified in other industries cannot be used as valid critical factors for software projects. In this research study, we did not differentiate between IT projects and software development projects, because IT projects involve software (Royal Academy of Engineering and the British Computer Society, 2004).

Some research studies and articles have reported on the critical success factors specific to software and IT projects (Standish Group, 2010; Sauer and Cuthbertson, 2003; Taylor, 2000) however, these studies are specific to one particular country. There has been no comprehensive study reported on different project sizes in various domains and in multiple countries. Such a thorough analysis is important in order to identify critical factors that are applicable for software projects. Three research questions have motivated the investigation reported here:

1. Research question 1 (RQ1): What factors, as identified in the study, have a critical impact in determining software project success?

Based on the findings from RQ1, we formulated other research questions as follows:

1. Research question 2 (RQ2): After over 20 years, are non-technical factors still a major contributor in determining software project success, as claimed by DeMarco (1988)?

2. Research question 3 (RQ3): Are there any clear and

significant differences between the critical success factors of software projects identified in this study and with those for general projects identified in other, related study?

MATERIALS AND METHODS

In this research study, the four well-known online journal databases: www.Sciencedirect.com, www.ieeexplore.ieee.org, www.springerlink.com and www.Emeraldinsight.com were extensively searched. As a result, we found 76 articles consisting of case studies, surveys and views of practitioners and experts. Of these 76 articles, only 43 articles were related to software projects, and 33 non-software projects were excluded. The remaining articles were analysed to develop a list of critical factors that specifically affect the success of software projects. The occurrences of each factor in the literature have been identified to determine the relative importance of each factor.

In this research study, each article was carefully reviewed, and a list of factors was compiled. There were three types of articles. First, in the articles describing the results of empirical study (that is, surveys and case studies), it was easy to identify the factors because the authors often provided a summary of success or failure factors in the form of a list or a separate paragraph. The well-known surveys, such as reports published by The Standish Group and the British Computer Society, belong to this category. Second, in the articles in which the authors (that is, experts and practitioners) have described the success and/or failure factors based on their wide range of experiences, it was also quite easy to identify the factors. Third, there were a few articles in which software or IT project failure was discussed, but the authors did not provide a summary of success or failure factors; in this case, each article was read carefully to avoid misunderstandings and misinterpretations. This research study covers 29 published sets of empirical data from case studies, 9 published empirical data sets from surveys and 5 articles written by experts and practitioners from 1990 to 2010. Only 2 publications were found from before 1990, namely those published by Schmitt and Kozar (1978) and Wingrove (1986), which are not covered in this analysis.

In order to analyse and produce a list of critical success factors from the extensive literature, the content analysis method was adopted. Content analysis is an approach to the quantification of qualitative data (Holsti, 1969). Although it was originally developed for the analysis of human communication in social sciences, several empirical software engineering studies, such as (Rainer et al., 2003; Rainer and Hall, 2003), have adopted this method as part of their research methodology. Babbie (2010) defined content analysis as the "*study of recorded human communications*" including various forms of communications such as books, magazine, web-pages and letters. In this research study, the communications to be analysed were the published articles. Seaman (1999) describes the method as follows: an "analysis method based on counting the frequency of occurrence of some meaningful lexical phenomenon in a textual data set." Meanwhile, Weber (1990) described the method of measurement in content analysis by "counting the occurrences or calculating percentages of meaning units such as specific words, phrases, content categories and themes, and later transfer to control document." This method later enables the application of frequency analysis by extracting quantitative data from qualitative data in the article and further recording it in frequency tables for the purposes of analysis.

Prior to performing the frequency analysis, the articles were read to generate appropriate categories for responses. Different factors that contributed to the same meaning were grouped into one category. For example, focused and hardworking staff, team commitment, team morale and motivated personnel were grouped

together in the 'committed and motivated team' category. This process was repeated until distinct sets of categories were obtained. Each category represents a critical success factor for software projects.

The method of content analysis was adopted in this study, rather than the data extraction method or the frequency analysis method alone, because some of the factors described by the authors in the articles were not explicitly clear and required careful reading, understanding and interpretation to produce accurate findings. Because different authors use different terms to identify the same factors, it is quite complicated to determine to which category a given factor belongs. It is not enough to simply count the occurrence of words (factors) as is done in the data extraction or frequency analysis methods. It is revealing that no study has used content analysis in the articles published from 1978 to 2010. One article by Wateridge (1998) produces a list of success criteria using only frequency analysis. Meanwhile, White and Fortune (2006), produce a list of success factors using only frequency analysis.

The inter-rater reliability was verified to ensure that there was no substantial bias or subjectivity in the identification and grouping process. Another researcher, who was not familiar with the current issues being discussed, was asked to identify factors that appeared in all the articles. The results were compared to those of the previous lists, and no major disagreements were found among the results.

In order to perform the frequency analysis, the occurrence of each success factor in each article in the literature search was recorded. The numbers and percentages of occurrences of each factor were then transferred and tabulated in a frequency table. By comparing the occurrences of a critical success factor in a number of articles against the occurrences of other factors in the same articles, the relative importance of each critical success factor could be calculated, and furthermore, the success factors could be compared and ranked. Finally, analysis and discussions were performed regarding the factors that were found to be critical in the literature.

Validity

The findings of the critical success factors in this research study were extracted from multiple empirical data and expert views reported in eight countries (that is, Finland, the United States, the United Kingdom, Hong Kong, Singapore, Belgium, Australia, and Canada), concerning small to large software projects in various domains; however, the findings are applicable only to software projects. Although a sample of eight countries is small and generalisability to the whole software engineering community worldwide is problematic, we have high confidence in our research findings because most of the articles included were taken from established scientific research journals and had a minimum of 11 citations, and a few of them were from well-known survey reports and articles written by experts and practitioners who had wide experience in software-related industries from 1990 until 2010. Often, the factors reported in books are based on the previous work of others and do not cover the latest research findings, so we did not consider books. We also decided not to include conference and workshop proceedings because it is hard to determine the quality of the articles in such publications. Note that this research study was not intended to localise the findings; thus, we considered it irrelevant to conduct an empirical study in any particular country.

RESULTS

Here, we present the findings with regard to the research questions presented at the end of the introduction.

Critical success factors

To answer RQ1, Table 2 shows the list of critical success factors of software projects identified throughout 43 publications. Based on the analysis of our extensive literature search, 26 critical success factors were found to be related to project success. The total frequency of occurrences was 372. Although some of the factors have a low frequency, we still decided to treat them as critical factors because the criticality depends not on the frequency but on the literature in which the critical success factors were highlighted by the articles.

There is a good opportunity, if an organisation or project manager is attentive, to control the top 5 critical factors to drive towards project success since the percentage of frequency of occurrences for each is more than 50%. The extensive literature shows that most of the practitioners consider clear requirements and specifications; clear objectives and goals; and a realistic schedule to be the three most critical success factors that can contribute to project success. These three critical success factors can be considered to be pre-project execution aspects that need to be made clear and solidified before commencing and executing software projects. Although 88% of the publications include at least one of these three factors, only 26% cite all three. This finding is very much in line with the research studies conducted by Wateridge (1995) and Fortune and White (2006), which reflect that there is no broad consensus among researchers and practitioners in determining project critical success factors.

The estimations of schedule and budget, which are the third and eighth highest factors, respectively, show that arbitrary and illogical estimation due to schedule pressure, or sometimes to a business strategy in order to win a tender, can contribute to project failure. The common problem of early estimation, without fully defined requirements, has troubled the software industries for more than fifty years (Boehm, 1981; Jones, 1998). Accurate estimation based on proven methods is needed to address these factors. Several methods, such as the Constructive Cost Model (COCOMO) and Constructive Cost Model-II (COCOMO-II), proposed by Boehm (1981) and Boehm et al. (1996), respectively; the PROxy-Based Estimation (PROBE) method through the adoption and utilisation of the Personal Software Process (PSP) by Humphrey (1995); Function Point Analysis; expert judgment method; Delphi technique; and also Wide Band Delphi technique by Boehm (1981), have been proposed and developed to support estimation procedures.

The results also suggest that, in the opinion of many practitioners, the top three most important human factors are effective project management skills and methodologies applied by the project manager, support from top management and user involvement. The critical success factor 'effective communication and feedback' (between stakeholders) was cited in 46.6% of the

literature, and was ranked at seventh place.

In our previous analysis, we combined skilled and sufficient staff with good performance vendors into one category, that is, 'skilled and sufficient staff/contractors/vendors', and this category falls into the fourth rank. However, we split this category into two, that is, 'skilled and sufficient staff' and 'good performance by vendors/contractors/consultants', once we realised that both are acquired and handled by different units in organisations, as outlined in the Project Management Body of Knowledge (PMBOK) (2008), and Projects In Control Environment 2 (PRINCE2) (2009). Staff-members are handled by the human resource department, while the vendors and consultants are acquired by the procurement department. Both departments handle, manage, acquire and assess these resources differently. Splitting this into two makes 'skilled and sufficient staff' become ranking number nine, while 'good performance by vendors/contractors/consultants' becomes ranking number twenty-five.

We also considered frozen requirements to be another group of factors, different from clear requirements and specifications, because sometimes the sponsors or end-users are very clear about what they want in the first stage, but as the project progresses through subsequent phases or stages, they might see the need to introduce new or changing requirements (or sometimes features that are not required). This also affects the project schedule and budget.

DISCUSSION

Here, we discuss the findings with regard to the research questions presented at the end of introduction particularly for RQ2 and RQ3.

Non-technical factors versus technical factors

According to Zahran (1998), software development has three main aspects, which are people, process and technology. Thus, in order to answer RQ2, the critical success factors have been grouped into these three general categories, that is, people factor, process factor and technical factor, as shown in Table 3. Based on our examination, from the 26 critical project success factors listed in Table 2, it appears that 7 critical success factors (27%) belong to the people factor category, 16 success factors (62%) belong to the process factor category and the remaining 3 critical factors (12%) belong to the technical factor category. This is not surprising, as software projects almost never fail because of technical reasons, despite the fact that people and process problems may manifest technically. Yet, these technical factors can be alleviated with proper management of people and process practise. Based on our findings, the

Table 2. Critical success factors identified across 43 publications.

Critical success factors	Literature citation	Citation count in the literature (n = 43)	
		Frequency	%
1	Clear requirements and specifications	26	60.5
2	Clear objectives and goals	24	55.8
3	Realistic schedule	23	53.5
4	Effective project management skills/ methodologies (project manager)	23	53.5
5	Support from top management	22	51.2
6	User/client involvement	20	46.5
7	Effective communication and feedback	20	46.5

Table 2. Cont'd.

8	Realistic budget		Schmidt et al., 2001; Jones,1996; Jones,1995; Whittaker, 1999; May,1998; OGC2005; Beynon-Davies, 1999; Baccharini et al., 2004; Drummond, 1998; Ewusi-Mensah, 1997; Ewusi-Mensah and Prazasnyski, 1994; Boehm, 1991; Charette, 2005; Standish Group, 2001, 2006; Clegg et al., 1997, Oz and Sosik, 2000; Oz, 1994; Sauer and Cuthbertson, 2003;	19	44.2
9	Skilled and sufficient staff		Schmidt et al., 2001; Keil et al., 2002; Sauer and Cuthbertson, 2003; Kappelman et al., 2006; Standish Group, 1995, 1999, 2009; May,1998; Jiang and Klein, 2000; Beynon-Davies, 1999; Baccharini et al., 2004; Ewusi-Mensah, 1997; Ewusi-Mensah and Prazasnyski, 1994; Boehm, 1991; Standish Group, 2001; Milis and Mercken, 2002; Reel, 1999; Oz and Sosik, 2000;	18	41.9
10	Frozen requirement		Schmidt et al., 2001; Jones,1996, 1995; Yeo, 2002; Jiang and Klein, 2000; Beynon-Davies, 1999; Drummond, 1998; Oz, 1994; Jiang et al., 2001; Nuseibeh, 1997; Taylor, 2000; Oz and Sosik, 2000; Taylor, 2006, Kappelman et al., 2006; Baccharini et al., 2004; Boehm, 1991; Sauer and Cuthbertson, 2003;	17	39.5
11	Familiarity with technology/development methodology		Schmidt et al., 2001; Standish Group, 1995; Whittaker, 1999; Jiang and Klein, 2000; Beynon-Davies, 1999; Jiang et al., 1999; Baccharini et al., 2004; Drummond, 1998; Oz, 1994; Ewusi-Mensah and Prazasnyski, 1994; Jiang et al., 2001; Charette, 2005; BCS, 2004; Oz and Sosik, 2000; Sauer and Cuthbertson, 2003;	15	34.9
12	Proper planning		Schmidt et al., 2001; Jones, 2006,1995; Kappelman et al., 2006; Standish Group, 1995; Whittaker, 1999; May,1998; Humphrey, 2005; Oz, 1994; Standish Group, 2001, 1999; Taylor, 2000; Milis and Mercken, 2002; Standing et al., 2006; Sauer and Cuthbertson, 2003;	15	34.9
13	Appropriate development processes/methodologies (process)		Schmidt et al., 2001; Jones,1995; OGC,2005; Beynon-Davies, 1999; Jiang et al., 1999; Drummond, 1998; Mahaney and Lederer, 2003; Jiang et al., 2001; Nuseibeh, 1997; Charette, 2005; Standish Group, 2009; Milis and Mercken, 2002; Oz and Sosik, 2000; Sauer and Cuthbertson, 2003;	14	32.6
14	Up-to-date reporting	progress	Jones, 2006; Jones, 1995; Whittaker, 1999; May, 1998; Baccharini et al., 2004; Humphrey, 2005; Ewusi-Mensah, 1997; Oz, 1994; Charette, 2005; Reel, 1999; Oz and Sosik, 2000; BCS, 2004;	12	27.9
15	Effective monitoring and control		Schmidt et al., 2001; Jones,1996; OGC,2005; Beynon-Davies, 1999; Humphrey, 2005; Mahaney and Lederer, 2003; Ewusi-Mensah, 1997; BCS, 2004; Reel, 1999; Oz and Sosik, 2000; Baccharini et al., 2004; Sauer and Cuthbertson, 2003;	12	27.9
16	Adequate resources		Kappelman et al., 2006; Standish Group, 1995, 2006; Jiang and Klein, 2000; Baccharini et al., 2004; Ewusi-Mensah and Prazasnyski, 1994; Milis and Mercken, 2002; Oz and Sosik, 2000; Jones, 2006; Beynon-Davies, 1999; Leveson, 2004	11	25.6
17	Good leadership		Schmidt et al., 2001; OGC,2005; Baccharini et al., 2004; Glaser, 2004; Humphrey, 2005; Drummond, 1998; Ewusi-Mensah, 1997; Standing et al., 2006; Reel, 1999; Clegg et al., 1997; Oz and Sosik, 2000;	11	25.6
18	Risk management		Whittaker, 1999; OGC, 2005; Yeo, 2002; Jiang et al., 1999; Ewusi-Mensah, 1997; Leveson, 2004; Nuseibeh, 1997; Charette, 2005; BCS, 2004; Oz and Sosik, 2000	10	23.3

Table 2. Contnd

19	Complexity, project duration, number of organisations involved	size, of	Schmidt et al., 2001; Sauer and Cuthbertson, 2003; Yeo, 2002; Jiang and Klein, 2000; Beynon-Davies, 1999; Glaser, 2004; Humphrey, 2005; Drummond, 1998; Jiang et al., 2001; Charette, 2005;	10	23.3
20	Effective change configuration management	and	Schmidt et al., 2001; Taylor, 2006; Kappelman et al., 2006; Jones, 1995; Whittaker, 1999; Baccarini et al., 2004; BCS, 2004; Taylor, 2000; Oz and Sosik, 2000; Sauer and Cuthbertson, 2003	10	20.9
21	Supporting tools and infrastructure	good	Jones, 1996, 1995; Jiang et al., 1999; Ewusi-Mensah, 1997; Ewusi-Mensah and Prazasnyski, 1994; Leveson, 2004; Standish Group, 2006, 2001, 2009	9	23.3
22	Committed and motivated team		Standish Group, 1995; Beynon-Davies, 1999; Jiang et al., 1999; Mahaney and Lederer, 2003; Ewusi-Mensah, 1997; Oz, 1994; Standing et al., 2006; Reel, 1999; Milis and Mercken, 2002	9	20.9
23	Good quality management		Jones, 2006, 1996, 1995; Baccarini et al., 2004; Boehm, 1991; Leveson, 2004; Ariane 501 Inquiry Board, 1996; Nuseibeh, 1997; Reel, 1999;	9	20.9
24	Clear assignment of roles and responsibilities		Schmidt et al., 2001; Keil et al., 2002; Jiang and Klein, 2000; Baccarini et al., 2004; Leveson, 2004; Milis and Mercken, 2002; Sauer and Cuthbertson, 2003;	7	16.3
25	Good performance by vendors/ contractors/ consultants		Schmidt et al., 2001; Taylor, 2006; Whittaker, 1999; Baccarini et al., 2004;	4	9.3
26	End-user training provision		Beynon-Davies, 1999; Jiang et al., 1999;	2	4.7

claim made by DeMarco (1988) over 20 years ago is still applicable today: "The success or failure of a project is seldom due to technical issues".

Comparison with previous study

For comparison, we have chosen a research study, conducted by Fortune and White (2006), which took a quite similar approach to that of our study in deriving project critical success factors. In their study, 63 publications were reviewed, consisting of case studies, surveys and theoretical studies. The method used was quite similar to ours; it covers different project sizes, various domains and multiple countries. We also found that this research study is the latest publication, reported up to September 2010, to derive critical success factors based on comprehensive literature. The only major difference is that our research study only considered and opted for software projects, but they included both: 36 non-software project articles and 27 software project articles,

with the proportions of 57 and 43%, respectively.

In order to answer RQ3, Table 4 highlights a number of significant differences between the findings (that is, p value highlighted for significant differences). 18 critical success factors (51%) have significant differences while 17 critical success factors (49%) have no significant differences between the studies. For example, 'realistic schedule', 'effective project management skills/methodologies (competent project manager)' and 'appropriate development processes/ methodologies (process)' were cited by both; however, the analysis shows that statistical differences exist between the studies. Another example: 'clear requirements and specifications', 'frozen requirements' and 'realistic budget' were not cited in (Fortune and White, 2006) but these critical factors were present in our study. Similarly, 'strong business case/sound basis for project', 'project sponsor/champion' and 'adequate budget' were present in (Fortune and White, 2006) but were not identified in our research study.

Figure 1 illustrates and indicates that there were some

Table 3. Critical success factors based on categories.

Critical success factors of software project		Rankings from Table 2
People-related factors (7 critical success factors)	Effective project management skills/methodologies (project manager)	4
	Support from top management	5
	User/client involvement	6
	Skilled and sufficient staffs	9
	Good leadership	17
	Committed and motivated team	22
	Good performance by vendors/contractors/ consultants	25
Process-related factors (16 critical success factors)	Clear requirements and specifications	1
	Clear objective/goal/scope	2
	Realistic schedule	3
	Effective communication and feedback	7
	Realistic budget	8
	Frozen requirement	10
	Proper planning	12
	Appropriate development processes/methodologies (process)	13
	Up-to-date progress reporting	14
	Effective monitoring and control	15
	Adequate resources	16
	Risk management	18
	Effective change and configuration management	20
Good quality management	23	
Clear assignment of roles and responsibilities	24	
End-user training provision	26	
Technical-related factors (3 critical success factors)	Familiar with technology/development methodology	11
	Complexity, project size, duration, number of organisations involved	19
	Supporting tools and good infrastructure	21

clear similarities and differences between the findings. It is clear that the aggregation of two data sets has cited 18 critical success factors (51%), and 17 critical success factors (49%) are only cited by individual studies.

The results suggest that the factors of clear and frozen requirements; realistic estimation of schedule and budget; and effective project management skills and methodologies applied by the project manager are the five most critical success factors of software projects, illustrating clear and significant differences between our study and the research study in (Fortune and White, 2006).

Conclusion

We have reported on our extensive literature survey of critical success factors that impact software projects. In this research study, 43 articles were found to be significant contributions that can be analysed to develop a list of critical factors that specifically affect the success of software projects. The 43 articles consist of 9

published sets of empirical data from case studies, 9 published empirical data sets from surveys and 5 articles written by experts and practitioners from 1990 to 2010. The method of content analysis was adopted in this study, rather than the data extraction method or the frequency analysis method alone, because some of the factors described by the authors in the articles were not explicitly clear and required careful reading, understanding and interpretation to produce accurate findings.

Based on the analysis of our extensive literature search, 26 critical success factors were found to be related to project success. There is a good opportunity, if an organisation or project manager is attentive, to control the top five critical factors to drive towards project success since the percentage of frequency of occurrences for each is more than 50%. All these critical success factors were then grouped into three general categories, namely, people factor, process factor and technical factor. Based on our examination, from 26 critical success factors, it appears that the non-technical factors dominated over technical factors, with 23 critical success factors and 3 critical success factors,

Table 4. Significant differences between the two studies.

	Critical success factors	Occurrence in our research study (n = 43)			Occurrence in [48] (n=63)			Chi-square test $\alpha= 0.05, df = 1$	
		Freq.	%	Rank	Freq.	%	Rank	X ²	p
1	Clear requirements and specifications	26	60.5	1	0	0	0	49.177	0.000
2	Clear objectives and goals	24	55.8	2	31	49.2	2	0.296	0.587
3	Realistic schedule	23	53.5	3	14	22.2	13	10.341	0.001
4	Effective project management methodologies (project manager) skills/	23	53.5	4	19	30.2	8	5.313	0.021
5	Support from top management	22	51.2	5	39	61.9	1	1.498	0.221
6	User/client involvement	20	46.5	6	24	38.1	5	0.580	0.447
7	Effective communication and feedback	20	44.2	7	27	42.9	4	0.071	0.790
8	Realistic budget	19	46.5	8	0	0	0	33.078	0.000
9	Skilled and sufficient staffs	18	41.9	9	20	31.7	6	0.950	0.330
10	Frozen requirement	17	39.5	10	0	0	0	28.939	0.000
11	Familiarity with technology/ development methodology	15	34.9	11	14	22.2	12	1.847	0.174
12	Proper planning	15	34.9	12	0	0	0	24.979	0.000
13	Appropriate development processes /methodologies (process)	14	32.6	13	6	9.5	23	8.473	0.004
14	Up-to-date progress reporting	12	27.9	14	29	46.0	3	3.857	0.050
15	Effective monitoring and control	12	27.9	15	12	19.0	16	1.007	0.316
16	Adequate resources	11	25.6	16	16	25.4	10	0.002	0.963
17	Good leadership	11	25.6	17	15	23.8	11	0.20	0.888
18	Risk management	10	23.3	18	13	20.6	14	0.67	0.795
19	Complexity, project size, duration, number of organisations involved	10	23.3	19	4	6.3	26	6.111	0.130
20	Effective change and configuration management	10	20.9	20	19	30.2	7	0.724	0.395
21	Supporting tools and good infrastructure	9	23.3	21	0	0	0	14.070	0.000
22	Committed and motivated team	9	20.9	22	0	0	0	14.070	0.000
23	Good quality management	9	20.9	23	0	0	0	14.070	0.000
24	Clear assignment of roles and responsibilities	7	16.3	24	0	0	0	10.724	0.001
25	Good performance by vendors/contractors/ consultants	4	9.3	25	10	15.9	19	1.048	0.306
26	End-user literacy, knowledge and skills to use the system	2	4.7	26	7	11.1	21	1.45	0.229
27	Strong business case/ sound basis for project	0	0	0	16	25.4	9	13.139	0.000
28	Project sponsor/champion	0	0	0	12	19.0	15	9.440	0.002
29	Adequate budget	0	0	0	11	17.5	17	8.563	0.003
30	Organisational adaptation/culture/ structure	0	0	0	10	15.9	18	7.704	0.006

Table 4. Contnd.

31	Planned close down/review/acceptance of possible failure	0	0	0	9	14.3	20	6.863	0.009
32	Political stability	0	0	0	6	9.5	22	4.493	0.035
33	Environmental influences	0	0	0	6	9.5	24	4.493	0.035
34	Past experience (learning from)	0	0	0	5	7.9	25	3.663	0.056
35	Different viewpoints (appreciating)	0	0	0	3	4.8	27	2.156	0.142

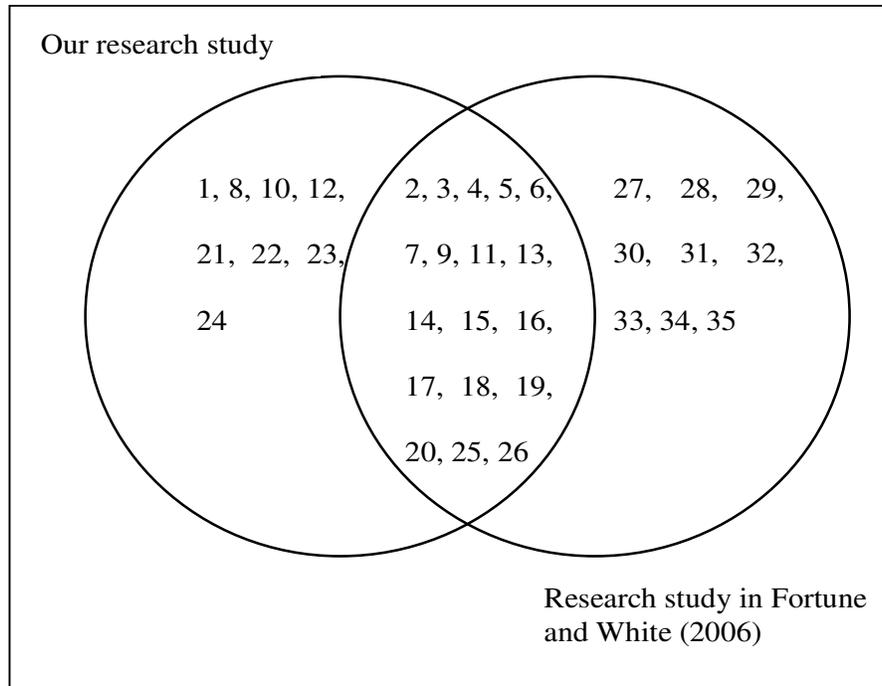


Figure 1. Comparison of two data sets.

respectively. Thus, our findings were in line with the claim made by DeMarco (1988) over 20 years ago.

We also found that the factors of clear and frozen requirements; realistic estimation of schedule and budget; and effective project management skills and methodologies applied by the project manager; are the five most critical success factors of software projects, illustrating clear and significant differences between our study and the research study by Fortune and White (2006) that focuses on general projects.

We believe that our findings will assist researchers in selecting a reliable set of critical success factors of software projects for empirical studies. One might continue to survey other countries to build a body of evidence to support the findings presented here. A similar study approach can be replicated in other fields such as construction, engineering, manufacturing, research and development, etc., in order to identify the critical success factors for that particular field.

For future work, the list of critical factors can be used as a valid factor to determine the degree to which a project management framework and methodology such as PMBOK or PRINCE2; software process improvement standards and methods such as the Capability Maturity Model Integration (CMM-I), International Standard for Software Process Improvement and Capability dEtermination (SPICE), International Organization for Standardization (ISO) 9000, or TSP; well-known software development processes such as agile processes, Rational Unified Process (RUP), etc. can address all these critical factors. Based on our initial research study, for example, PMBOK and Team Software Process (TSP) contribute a certain degree to addressing the identified critical success factors (Nasir and Sahibuddin, 2011). To highlight 'realistic schedule' as an example, PMBOK uses formal processes for estimating and scheduling that are capable of producing a realistic schedule when suitable historical data is available. However, nothing specific is

included in the process to address political pressure to produce lower cost estimates and shorter schedules during the negotiations of scope, budget, and delivery date that are private between the project manager and sponsoring management. This can be a much bigger problem than the lack of capability of the estimating process. TSP, in contrast, uses a highly capable estimation process based on historical data. The entire team produces and reviews the plan together, identifies any proposed exceptions to management goals and constraints, and participates in the out-brief to sponsoring management. This helps catch omissions and inconsistencies that could result in an unrealistic plan. Any negotiations concerning scope, budget, staffing, etc. are performed in public as part of the out brief meeting. Team and management commitments are documented in the minutes of the meeting. This is explicitly included to prevent the project manager from privately acquiescing to unrealistic goals or constraints under pressure from management or the client. We can integrate both models together to complement each other to more effectively address the critical success factors for software projects.

Each framework or methodology on its own could not perfectly address all the identified critical success factors. By blending a project management framework and software process improvement, or other excellent software development processes, we believe one can address all the critical success factors to a better degree. This is supported by several research efforts that seek integration between a few areas, such as traditional project management and agile (Hass, 2007), CMMI for Development (CMMI-Dev) and PMBOK (von Wangenheim et al., 2010), PMBOK and Rational Unified Process (Callegari and Bastos, 2007), agile and PRINCE (Nawrocki et al., 2006), CMMI and PMBOK (Jenkins, 2005) and many more in order to ensure better control in managing software projects. A study conducted by Bayo et al. (2007) showed that there has been increasing demand in the knowledge and application of quality software to improve the country socio-economic growth. Thus, a comprehensive model is needed in ensuring the way to produce high quality software.

ACKNOWLEDGEMENTS

The authors are grateful to the Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya and Advanced Informatics School, Universiti Teknologi Malaysia for supporting this research effort.

REFERENCES

- Ariane 501 Inquiry Board (1996). Ariane 5: Flight 501 Failure. Report by the Inquiry Board. Paris: Ariane 501 Inquiry Board. Available at: <http://www.di.unito.it/~damiani/ariane5rep.html>
- Babbie ER (2010). *The Practice of Social Research*, 10th ed., Thomson Learning Inc., p. 333.

- Baccarini D, Salm G, Love PED (2004). Management of Risks in Information Technology Projects. *Ind. Manag. Data Syst.*, 104(4): 286-295.
- Bayo ML, Ekene NS, Kenneth AC, Idowu SF (2007). Software development: An attainable goal for sustainable economic growth in developing nations: The Nigeria experience. *Intl. J. Phys. Sci.*, 2(12): 318-323
- Belassi W, Tukel OI (1996). A New Framework for Determining Critical Success/Failure Factors in Projects. *Int. J. Proj. Manag.*, 14(3): 141 - 151
- Beynon-Davies P (1999). Human Error and Information System Failure: The Case of the London Ambulance Service Computer-Aided Dispatch System Project. *Interact. Comput.*, 11(6): 699-720.
- Boehm B (1981). *Software Engineering Economics*, Prentice-Hall, p. 117.
- Boehm BW, Clark B, Horowitz E, Westland C, Madachy R, Selby R (1996). The COCOMO 2.0 Software Cost Estimation Model – A Status Report. *American Programmer*, pp. 2-17.
- Boehm BW (1991). *Software Risk Manage. Principles and Practices*. *IEEE Softw.*, 8(1): 32-41
- Brooks FP (1975). *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, p. 13.
- Brooks FP (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *IEEE Comput.*, 20(4):10-19
- Brooks FP (1995). *The Mythical Man-Month: Essays on Software Engineering*, Anniversary ed., Addison-Wesley, p. 13.
- Callegari D, Bastos R (2007). Project Management and Software Development Processes: Integrating RUP and PMBOK. *Proceedings International Conference Systems Engineering and Modeling 2007*, Israel, IEEE, pp. 1-8.
- Charette RN (2005). Why Software Fails. *IEEE Spectrum.*, 42(9): 42-49.
- Clegg C, Axtella C, Damodaran L, Farbey B, Hull R, Llyod-Jones R, Nicholls J, Sells R, Tomlinson C (1997). *Information Technology: A Study of Performance and the Role of Organizational Factors*. *Ergonomics*. 40(9): 851-857.
- Cooke-Davies TJ, Arzymanow A (2002). The maturity of project management in different industries: An investigation into variations between project management models. *Int. J. Proj. Manage.*, 21(6): 471-478
- DeMarco T (1988). Looking for Lost Keys. *Software Magazine*, pp. 58-62.
- Drummond H (1998). Riding a Tiger: Some Lessons of Taurus. *Manage. Decis.*, 36(3): 141-146.
- Dvir D, Lipovetsky S, Shenhar A, Tishler A (1998). In Search of Project Classification: A Non-Universal Approach to Project Success Factors. *Res. Policy.*, 27(9): 915-935
- Eveleens JL, Verhoef C (2010). The Rise and Fall of the Chaos Report Figures. *IEEE Softw.*, 27(1): 30-36
- Ewusi-Mensah K (1997). Critical Issues in Abandoned Information Systems Development Projects. *Commun. ACM.*, 40(9): 74-80.
- Ewusi-Mensah K, Prazasnyski, ZH (1994). Factors Contributing to the Abandonment of Information Systems Development Projects. *J. Inform. Tech.*, 9: 185-201.
- Fairley ER (2009). *Managing and Leading Software Projects*, Wiley-IEEE Computer Society Press
- Fortune J, White D (2006). Framing of Project Critical Success Factors by a Systems Model. *Int. J. Proj. Manage.*, 24(1): 53-65
- Galin D (2004). *Software Quality Assurance-From Theory to Implementation*, Addison-Wesley, pp. 4-7.
- Glaser J (2004). Management's Role in IT Project Failures. *Healthc. Financ. Manage.*, 58(10): 90-92.
- Glass RL (2005). IT Failure Rates-70% or 10-15%. *IEEE Softw.*, 22(3): 110-112
- Glass RL (2006). The Standish Report: Does It Really Describe a Software Crisis? *Commun. ACM.*, 49(8): 15-16
- Glass RL (1998). *Software Runaways: Lessons Learned from Massive Software Project Failures*, Prentice Hall, pp. 1-288.
- Gray CF, Larson EW (2008). *Project Management: The Managerial Process*, 4th ed., Irwin/McGraw-Hill, p. 97.
- Hartmann D (2006). Interview: Jim Johnson of the Standish. Available at <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS> accessed September 2010.

- Hass KB (2007). The Blending of Traditional and Agile Project Management. *PM World Today*, 9(5): 1-8.
- Holsti OR (1969). *Content Analysis for the Social Sciences and Humanities*, Addison-Wesley, p. 9.
- Humphrey WS (1995). *A Discipline for Software Engineering*, Addison-Wesley, pp. 1-816.
- Humphrey WS (2005). Why Big Software Project Fail: The 12 Key Questions. *CrossTalk, J. Def. Software Eng.*, 18(3): 25-29
- lbbs CW, Kwak YH (2000). Assessing Project Management Maturity. *Project Manag. J.*, 31(1): 32-43
- In Naur P, Randell B (Eds.) (1969). *Software Engineering: Report on a Conference Sponsored by the NATO Science Committee, NATO Science Committee, Belgium*. Available at: <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>
- Jain M (2008). Delivering Successful Projects with TSP and Six Sigma: A Practical Guide to Implementing Team Software Process, Auerbach Publications, p. 4.
- Jenkins M (2005). Combining Multiple Models to Develop a Software Project Management Methodology. *Proceedings International Conference Software Engineering and Applications, USA*. pp. 467
- Jiang J, Klein G (2000). Software Development Risks to Project Effectiveness. *J. Syst. Software.*, 52(1): 3-10.
- Jiang J, Klein G, Discenza R (2001). Information System Success as Impacted by Risks and Development Strategies. *IEEE T. Eng. Manage.*, 48(1): 46-55.
- Jiang JJ, Klein G, Balloun JK, Crampton SM (1999). System Analysts' Orientations and Perceptions of System Failure. *Inform. Software Tech.*, 41(2): 101-106.
- Jones C (1998). Sizing Up Software. *Scientific American Magazine.*, 279(6): 104-111.
- Jones C (1995). Patterns of Large Software Systems: failure and Success. *Comput.*, 28(3): 86-87.
- Jones C (1996). Our Worst Current Development Practices. *IEEE Softw.*, 13(2): 102-104.
- Jones C (2006). Social and Technical Reasons for Software Project Failures. *CrossTalk, The J. Def. Software Eng.*, 19(6): 4-9.
- Jorgensen M, Molokken K (2006). How Large Are Software Cost Overruns? A Review of the 1994 Chaos Report. *Inform. Software Tech.*, 48(4): 297-301
- Kappelman LA, Mckeeman R, Zhang L (2006). Early Warning Signs of IT Project Failure, The Dominant Dozen. *Inform. Syst. Manage.*, 23(4): 31-36.
- Keil M, Tiwana A, Bush A (2002). Reconciling User and Project Manager Perceptions of IT Project Risk: A Delphi Study. *Inform. Syst. J.*, 12(2): 103-119.
- Leveson NG (2004). The Role of Software in Spacecraft Accidents. *J. Spacecraft Rockets.*, 41(4): 564-575.
- Leveson NG, Turner CS (1993). An Investigation of the Therac-25 Accidents. *IEEE Comput.*, 26(7): 18-41
- Mahaney RC, Lederer AL (2003). Information System Project Management: An Agency Theory Interpretation. *J. Syst. Software.*, 68(1): 1-9.
- May LJ (1998). Major Causes of Software Project Failure. *CrossTalk, J. Def Software Eng.*, 11(7): 1-7.
- Milis K, Mercken R (2002). Success Factors Regarding the Implementations of ICT Investment Projects. I. *J. Prod. Econ.*, 80(1): 105-117.
- Nasir MHN, Sahibuddin S (2011). Addressing a Critical Success Factor for Software Projects: A Multi-Round Delphi Study of TSP. *Intl. J. Phys. Sci.*, 6(5): 1213-1232.
- Nawrocki J, Olek L, Jasinski M, Paliswiat B, Walter B, Peirtzak B, Godek P (2006). Balancing Agility and Discipline with XPrince. *Lecture Notes in Comput. Sci.*, 3943: 266-277
- Nuseibeh B (1997). Ariane 5- Who Dunnit? *IEEE Softw.*, 14(3): 15-16.
- Office of Government of Commerce (2009). *Managing Successful Projects with PRINCE2*, 5th ed., Office of Government of Commerce
- OGC Best Practice. (2005). *Common Causes of Project Failure*, Office of Government of Commerce.
- Othman M, Zain AM, Hamdan AR (2010). Review on Project Management and Issues Surrounding Dynamic Development Environment of ICT project: Formation of Research Area. *Intl. J. Digital Content Technol. its Appl.*, 4(1) .
- Oz E (1994). When Professional Standards are Lax, The CONFIRM Failure and its Lessons. *Commun. ACM.*, 37(10): 29-36.
- Oz E, Sosik JJ (2000). Why Information Systems Projects are Abandoned: A Leadership and Communication Theory and Exploratory Study. *J. Comput. Inform. Syst.*, 41(1): 66-77.
- Perkins TK (2006). Knowledge: The Core Problem of Project Failure. *CrossTalk, The J. Def. Software Eng.*, 19(6): 13-15.
- Pinto JK, Mantel SJ (1990). The Causes of Project Failure. *IEEE T. Eng. Manage.*, 34(7): 305-327
- Pinto JK, Rouhiainen PJ (2001). *Building Customer-Based Project Organizations*, John Wiley and Sons, p. 87.
- Procaccino JD, Verner JM, Overmyer S, Darter ME (2002). Case Study: Factors for Early Prediction of Software Development Success. *Inform. Software Tech.*, 44(1): 53-62.
- Project Management Institute (2008). *PMBOK: A Guide to the Project Management Body of Knowledge*, 4th ed., Project Management Institute
- Rainer A, Hall T (2003). A quantitative and qualitative analysis of factors affecting software processes. *J. Syst. Software.*, 66(1): 7-21
- Rainer A, Hall T, Baddoo N (2003). Persuading Developers to 'Buy Into' Software Process Improvement: Local Opinion and Empirical Evidence. *Proceedings of the International Symposium on Empirical Software Engineering (ISESE'03)*, Rome, Italy. *IEEE*, pp. 326-335
- Reel JS (1999). Critical Success Factors in Software Projects. *IEEE Softw.*, 16(3): 19-23.
- Royal Academy of Engineering and the British Computer Society (2004). *The Challenges of Complex IT Projects*, The Royal Academy of Engineering
- Sauer C, Cuthbertson C (2003). The State of IT Project Management in the UK 2002-2003. *Computer Weekly*, 15 April, pp. 1-82.
- Schmidt R, Lyytinen K, Keil M, Cule P (2001). Identifying Software Project Risks: An International Delphi Study. *J. Manage. Inform. Syst.*, 17(4): 5-36
- Schmitt JW, Kozar KA (1978). Management's Role in Information System Development Failures: A Case Study. *MIS Quart.*, 2(2): 7-16
- Seaman CB (1999). Qualitative Methods in Empirical Studies of Software Engineering. *IEEE T. Software Eng.*, 25(4): 557-572
- Simons CL, Parmee IC, Coward PD (2003). 35 Years on: to what Extent has Software Engineering Design Achieved its Goal? *IEE Proc-Software.*, 150(6): 337-350
- Standing C, Guilfoyle G, Lin C, Love PED (2006). The Attribution of Success and Failure in IT Projects. *Ind. Manag. Data Syst.*, 106(8): 1148-1165.
- Standish Group International (1995). *Chaos. Technical Report*.
- Standish Group International (1996). *Chaos. Technical Report*.
- Standish Group International (1999). *Chaos: A Recipe for Success. Technical Report*.
- Standish Group International (2001). *Extreme Chaos. Technical Report*.
- Standish Group International (2009). *Chaos Summary 2009: 10 Laws of CHAOS. Technical Report*.
- Standish Group International (2010). *Chaos Summary for 2010. Technical Report*.
- Swartz AJ (1996). Airport 95: Automated Baggage system? *ACM SIGSOFT.*, 21(2): 79-83
- Taylor A (2000). IT Projects: Sink or Swim. *Comput. Bull.*, 42(1): 24-26
- Taylor H (2006). Critical Risks in Outsourced IT Projects: The Intractable and the Unforeseen. *Commun. ACM.*, 49(11): 75-59
- Tukel OI, Rom WO (2001). An Empirical Investigation of Project Evaluation Criteria. *Int. J. Oper. Prod. Man.*, 21(3): 400-416
- Von WCG, Silva DAD, Buglione L, Scheidt R, Prikladnicki R (2010). Best Practice Fusion of CMMI-DEV v1.2 (PP, PMC, SAM) and PMBOK 2008. *Inform. Software Tech.*, 52(7): 749-757
- Wateridge J (1995). IT projects: A Basis for Success. *Int. J. Proj. Manage.*, 13(3): 169-172
- Wateridge J (1998). How Can IS/IT Projects be Measured for Success. *Int. J. Proj. Manage.*, 16(1): 59-63
- Weber RP (1990). *Basic Content Analysis*, 2nd ed., SAGE Publications, pp. 1-96.
- White D, Fortune J (2002). Current Practice in Project Management – An Empirical Study. *Int. J. Proj. Manage.*, 20(1): 1-11
- Whittaker B (1999). What Went Wrong? Unsuccessful Information Technology Projects. *Information Manage. Comput. Secur.*, 7(1): 23-

- 29.
- Wingrove A (1986). The Problems of Managing Software Projects. *Software Eng. J.*, 1 (1): 3-6
- Yeo KT (2002). Critical Failure Factors in Information System Projects. *Int. J. Proj. Manage.*, 20(3): 241-246.
- Zahran SM (1998). *Software Process Improvement: Practical Guidelines for Business Success*, Addison-Wesley, p. 20.
- Zwikael O, Globerson S (2006). From Critical Success Factors to Critical Success Processes. *Int. J. Prod. Res.*, 44(17): 3433-3449.