Anticipating Success of a Business-Critical Software Project: A Comparative Case Study of Waterfall and Agile Approaches

Marko Ikonen and Pekka Abrahamsson

University of Helsinki, Finland
{marko.ikonen,pekka.abrahamsson}@cs.helsinki.fi

Abstract. A business decision to abort projects with little or no chance at succeeding should be made as early as possible. The research on success of software engineering projects is fragmented and unorganized, which makes anticipating outcomes difficult and possibly error prone. This short paper offers a preliminary insight into success factors related to project outcomes that can be found at the midpoint of the development projects. We conducted a comparative case study where eight software development projects used the waterfall development method and four projects agile software development approaches as their primary development vehicle. Due to the explorative nature of the research, we conducted these in university settings. The results reveal that signs at project failure can be seen in the middle of the projects.

Keywords: anticipation, success factor, software engineering project.

1 Introduction

Standish Group's CHAOS reports have repeatedly shown dramatic problems in running software projects successfully regardless of the type of software being developed. While strong critique has been aimed at the validity on these report results [7,11], we can argue that the results are indicative enough to warrant serious attention. In the case of a software project failure, damage is caused not only to the project itself but to the products, services, and other business which are dependent on the project. Not enough emphasis has been placed on increasing the ability to anticipate the project outcome as early as possible.

Software development of the 2010s is still very much people-oriented. At a team level, people management requires leading skills, such as motivating, empowering, supporting and truly caring for people, which affects project results [3,16]. However, even a project with unmotivated teams can be successful. In other words, a single viewpoint is likely to be insufficient when trying to determine the level of success. While the skills of a project manager and the group dynamics are critical, so is the direction of a project. Technical competence without support from the organization, skills in coordination, communication, or influencing possibilities is risky. A business decision to abort projects with little or no chance of succeeding should be made as early as possible.

P. Tyrväinen, S. Jansen, and M.A. Cusumano (Eds.): ICSOB 2010, LNBIP 51, pp. 187–192, 2010.

This short paper addresses this problem domain of coordination, communication and cohesion, and offers a preliminary insight into success factors related to project outcomes that can be found at the midpoint of the development projects. Research on success regarding software engineering projects is scarce and fragmented. Critical success factors (e.g. [15]) in software have, however, been identified but only *a posteriori*. The same applies to software process improvement studies (e.g. [1,5]). Our initial findings from an earlier study [10] suggest anticipating project success with high accuracy is possible. In this study, we extend our research to include modern agile software engineering projects using the Scrum method.

2 Preliminary Research Model for Anticipating Project Success

When developing a model to determine a project's success, the following two issues need to be addressed: (1) the level of subjectivity in performing the analysis and (2) which model will be selected as the backbone for the new model. As stated, the research in this area is fragmented and unorganized. Kitchenham et al. [13] illustrate this. None of the existing models fit our purposes perfectly. Several isolated relationships can be deduced from the literature between a "project success" and, for example, "actions" of the development teams. These relationships come from several sources with a wide difference in the context where they were identified. We decided to opt for a different strategy. We treat "all" existing models as equals and compile a comprehensive mega-model as our preliminary research model. To meet these needs we have built an all-inclusive questionnaire designed to include the majority of success relationships drawn from different reference disciplines. The description of the research model can be found in [9]. For the purposes of the study we present the preliminary research model in Figure 1 with the reference disciplines, theories used and their expected impact. Due to the very limited space, the references are excluded.

3 Study Setting

The study of [10] showed a possibility of predicting a software engineering project success, based on a small, qualitative sample. While understanding that the dynamics beyond traditional process models is important, we conducted a comparative experiment to find out whether anticipation is plausible for modern, Scrum-based projects. In the experiment, we used the same measurement and analysis techniques as in the study of [10], except we focused on project success only.

Our goal is to reveal the key questions (i.e. the most expressive power to reveal driving and restraining elements) regarding Scrum-based projects. Scoring for this was done by aggregating the answers of each member within each project. Then, these project-specific answers were scored by comparing them with the

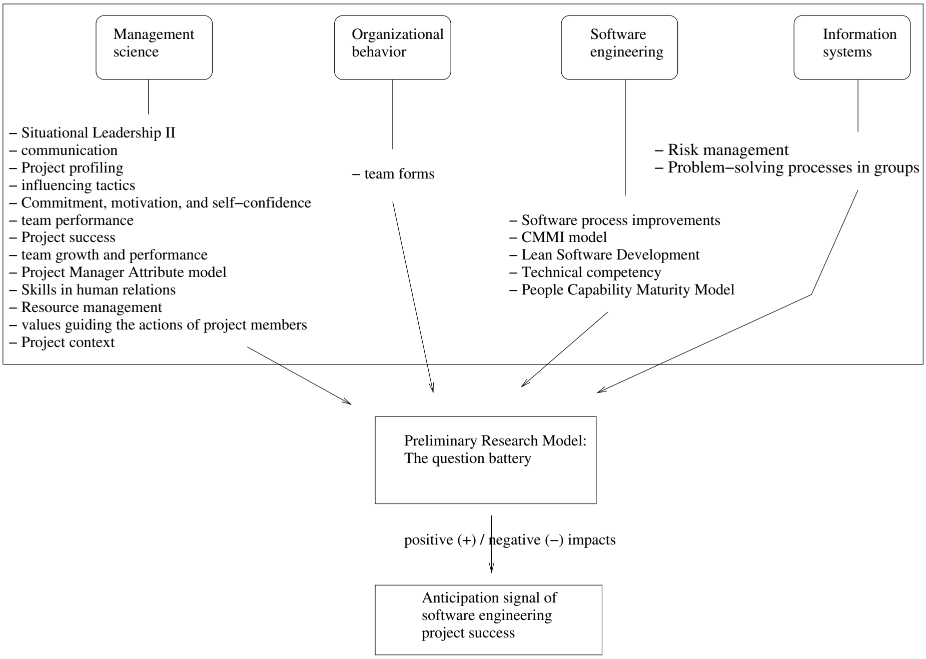

Fig. 1. The preliminary research model with the reference disciplines and theories

answers of other projects, question by question. If we perceived something to have a positive effect on the project success, it increased the project's *driving score* (DRI) by 1 to 7 points. Similarly, issues perceived as a negative effect increased the *restraining score* (RES) by 1 to 7 points. An answer can increase both driving and restraining scores.

The data gathering started in autumn 2007 and spanned eight quartiles. Totally, 61 project members were interviewed from twelve projects. In order to determine whether a particular project was successful, after a project ended, the customers of each project estimated the success in cooperation with the supervisors. Such a determination for success covers Shenhar's first and second dimensions: project efficiency and impact on customer (see [17]). Due to a possible evaluation bias, an extra evaluation, called an expert evaluation described in [10] and performed similarly, was used to compensate them.

4 Results

We explored how accurately we can anticipate the success of Scrum development projects. In this and subsequent analyses we keep continuously comparing the waterfall method results equally to enable continuous reflection on the differences. Table 1 presents the scaled scores from the interview sessions and their correlation to the success. Success, in its turn, is presented in Table 2a (see the

Table 1. The scaled scores of driving and restraining elements for the projects: the waterfall-based projects (marked with *w*) of the original study and the Scrum-based projects (marked with *s*) of our study. The correlations of these scores to the project success scores (i.e. *avg* in Table 2a) are presented.

	project												
element	w1	w2	w3	w4	w5	w6	w7	w8	s9	s10	s11	s12	**correlation**
DRIw	.53	.22	.16	.42	.03	.45	.42	.36					.87
DRIs									.63	.48	.42	.39	.13
RESw	.86	.66	.75	.92	.39	.80	.86	.86					.93
RESs									.78	.68	.61	.82	.93

Table 2. (a) The evaluations of project success after the projects made by expert evaluator and the projects' boards (including the customer and the supervisor). *w* refers to waterfall, *s* to Scrum-based project. *avg* is the average of the two success scores and is used for the correlation calculations in Table 1. *stdev* represents standard deviation. **(b)** The top 10 lists of the scored questions with the scores (scr). The codes relate to the preliminary research model (see Chapter 4 in [9]).

project	expert	board	**avg**	stdev
w1	.83	.90	**.87**	.05
w2	.63	.61	**.62**	.01
w3	.57	.95	**.76**	.27
w4	.90	.84	**.87**	.04
w5	.37	.27	**.32**	.07
w6	.83	.96	**.90**	.09
w7	.87	.82	**.85**	.04
w8	.60	.82	**.71**	.16
s9	.76	.75	**.76**	.01
s10	.70	.80	**.75**	.07
s11	.62	.76	**.69**	.10
s12	.73	.84	**.79**	.08

(a)

DRIs		*RESs*		*DRIw*		*RESw*	
code	score	code	score	code	score	code	score
CO-03	24	PP-05	26	CO-03	22	CO-03	21
CO-04	24	CM-03	14	RS-05	17	CM-02	11
PG-13	22	PI-08	14	PG-13	16	PP-05	11
PI-10	22	CO-03	12	CM-02	14	MF-06	9
CO-12	20	PG-30	12	MF-06	13	CO-04	8
CO-02	18	CO-05	10	PI-10	13	CO-21	8
PG-01a	18	RS-06	10	CO-13	11	RS-06	8
MF-17	16	IP-01	8	PG-11	11	CO-12	7
PI-08	16	PP-03	8	PG-14	11	RS-01	7
RI-04	16	CO-12	6	CO-02	10	PI-10	6

(b)

avg column). The driving element scores for Scrum projects (*DRIs*) do not correlate with project success since the correlation value is only .13. Nevertheless, the driving scores for waterfall projects (*DRIw*) correlate strongly (.87). Moreover, a strong correlation between the restraining elements for Scrum projects (*RESs*) and project success (Table 2a) exists (.93). In addition, restraining scores for waterfall projects (*RESw*) have as high correlation (.93) as *RESs*.

The average score *avg* of the two success scores in Table 2a was used in calculating the correlations (Table 1) between success and the scores conducted from the interview.

The strong value of correlation between the restraining scores and success of Scrum projects (the correlation for *RESs* in Table 1) is considered a promising signal of an accurate anticipation ability regarding project success. The questions with their codes are presented in [9]. Table 2b shows the ten most accumulated scores for positive drivers (*DRIs*, *DRIw*) and for negative restraining elements (*RESs*, *RESw*). Each of the lists is ordered based on the number of scores they received. The *RESs* and *RESw* columns in Table 2b are the most appealing due to their strong correlation (Table 1).

5 Summary

We maintain that anticipation provides a valuable tool for business critical software projects. Success, in terms of project efficiency and impact on the customer, can be seen after a project ends. However, an economical tension requires a business decision to abort projects with little or no chance to succeed as early as possible. Without anticipation, these decisions cannot be made until the end.

We found key questions which revealed the most restraining elements in the projects evaluated in the study. For the waterfall projects, both the drivers (*DRIw*) and restrainers (*RESw*) equally explained the project success whilst only the restraining elements worked to anticipate success in Scrum projects (*RESs*). We found that the less restraining elements there are, the more successful the project will be. This is the case with both waterfall and Scrum-based projects since the correlations for both (*RESw* and *RESs*) were .93. Hence, the key questions for anticipating success in Scrum projects are found in the *RESs* column in Table 2b. The five most powerful ones are PP-05, CM-03, PI-08, CO-03, and PG-30. In waterfall projects, the most useful questions in addition to CO-03 and PP-05 are CM-02, MF-06, CO-04, CO-21, and RS-06 (see the *RESw* column in Table 2b.). These findings are very similar to the literature, as shown, for example, in [2,3,4,6,12]. Lack of communication skills, oral as well as written, of experienced engineers in business has been found to be a commonplace problem [14].

Despite the promising results presented in this paper, they cannot validate the findings comprehensively. An obvious limitation to the validity of the proposed findings is the use of students in this study as study subjects. However, it is quite well established that when one seeks to establish a trend, the use of students is quite acceptable [18]. Höst et al. [8] concluded that students are indeed relevant when considering experimentation in software engineering. We do not maintain that our findings are one-to-one with industry but rather that, given the specific circumstances, we indicate that there may be a trend explaining project success or failure when a particular set of indicators are searched for.

While the fragmented area of success in SE projects still lacks a holistic model capable to explain project success comprehensively, the results above are encouraging for pursuing the development of such a model. Once successful, this is a significant cost-saving opportunity for software businesses when applying the model in practice. The new and revised model visible in Table 2b serves as a mini-model for practitioner use already at its current state.

References

1. Abrahamsson, P.: Measuring the success of software process improvement: the dimensions. In: Proceedings of EuroSPI 2000 -conference (2000)
2. Addison, T., Vallabh, S.: Controlling software project risks: an empirical study of methods used by experienced project managers. In: SAICSIT 2002: Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, pp. 128–140. South African Institute for Computer Scientists and Information Technologists (2002)
3. Situational Leadership® II - The Article. The Ken Blanchard Companies (2001)
4. Boehm, B.: Software engineering economics. Prentice-Hall, New Jersey (1981)
5. Conradi, R., Fuggetta, A.: Improving software process improvement. IEEE Software 19(4), 92–99 (2002)
6. Curtis, B., Krasner, H., Iscoe, N.: A field study of the software design process for large systems. Communications of the ACM 31(11), 1268–1287 (1988)
7. Glass, R.L.: The Standish report: does it really describe a software crisis? Communications of the ACM 49(8), 15–16 (2006)
8. Höst, M., Regnell, B., Wohlin, C.: Using students as subjects - a comparative study of students and professionals in lead-time impact assessments. Journal of Empirical Software Engineering 5(3), 201–214 (2000)
9. Ikonen, M.: Working Toward Success Factors in Software Development Projects. Number 2009-19 in Series of Publications C. Department of Computer Science. University of Helsinki, Helsinki (2009)
10. Ikonen, M., Kurhila, J.: High-impact success factors in capstone software projects. In: SIGITE 2009: Proceedings of the 10th ACM SIGITE conference on Information technology education. ACM, New York (2009)
11. Jørgensen, M., Moløkken-Østvold, K.: How large are software cost overruns? critical comments on the standish group's chaos reports. Information and Software Technology 48(4), 297–301 (2006)
12. Keil, M., Cule, P.E., Lyytinen, K., Schmidt, R.C.: A framework for identifying software project risks. Communications of the ACM 41(11), 76–83 (1998)
13. Kitchenham, B., Budgen, D., Brereton, P., Turner, M., Charters, S., Linkman, S.: Large-scale software engineering questions – expert opinion or empirical evidence? IET Software 1(5), 161–171 (2007)
14. Norback, J., Hardin, J.: Integrating workforce communication into senior design. IEEE Transactions on Professional Communication 48(4), 413–426 (2005)
15. Reel, J.: Critical success factors in software projects. IEEE Software 16(3), 18–23 (1999)
16. Robbins, S.P.: Essentials of organizational behavior. San Diego State University, Prentice Hall (2002)
17. Shenhar, A.J., Levy, O., Dvir, D.: Mapping the dimensions of project success. Project Management Journal 28(2), 5–13 (1997)
18. Tichy, W.: Hints for reviewing empirical work in software engineering. Journal of Empirical Software Engineering 5(4), 309–312 (2000)