Available online at www.sciencedirect.com
ScienceDirect
The Journal of Systems and Software 81 (2008) 961–971
The Journal of Systems and Software
www.elsevier.com/locate/jss

A survey study of critical success factors in agile software projects

Tsun Chow, Dac-Buu Cao *

School of Business and Technology, Capella University, Minneapolis, MN 55402, USA

Received 20 February 2007; received in revised form 12 August 2007; accepted 17 August 2007
Available online 26 August 2007

Abstract

While software is so important for all facets of the modern world, software development itself is not a perfect process. Agile software engineering methods have recently emerged as a new and different way of developing software as compared to the traditional methodologies. However, their success has mostly been anecdotal, and research in this subject is still scant in the academic circles. This research study was a survey study on the critical success factors of Agile software development projects using quantitative approach.

Based on existing literature, a preliminary list of potential critical success factors of Agile projects were identified and compiled. Subsequently, reliability analysis and factor analysis were conducted to consolidate this preliminary list into a final set of 12 possible critical success factors for each of the four project success categories – Quality, Scope, Time, and Cost.

A survey was conducted among Agile professionals, gathering survey data from 109 Agile projects from 25 countries across the world. Multiple regression techniques were used, both at the full regression model and at the optimized regression model via the stepwise screening procedure. The results revealed that only 10 out of 48 hypotheses were supported, identifying three critical success factors for Agile software development projects: (a) Delivery Strategy, (b) Agile Software Engineering Techniques, and (c) Team Capability.

Limitations of the study are discussed together with interpretations for practitioners. To ensure success of their projects, managers are urged to focus on choosing a high-caliber team, practicing Agile engineering techniques and following Agile-style delivery strategy.

Keywords: Software development; Agile methods; Critical success factors

1. Introduction

While software is so important for the all facets of the modern world, software development itself is not a perfect process. Despite the efforts to employ software engineering methodologies, software development has not been consistently successful, thus often resulting in delayed, failed, abandoned, rejected software projects. Even those software projects already implemented may need expensive on-going maintenance and corrective releases or service packs.

The above shortcomings have affected the bottom line for information technology (IT) and software development organizations in a big way. The challenge here is how software development management can be improved to avoid the above problems of waste and inefficiency? There has been a recent emergence of a new class of software development process called Agile methods, which operate rather differently from traditional methods.

The present research seeks to identify and provide insight into the critical success factors (CSF's) that help software development projects using agile methods to succeed. The study compiled the success factors reported in the agile literature, performed reliability analysis and factor analysis on those factors and consolidated them into a final 12 possible success factors for Agile projects in five different categories: Organizational, People, Process, Technical, and Project. A web-based survey was conducted to gather feedback from 109 agile software projects from 25 countries around the world, and the collected data were analyzed using the multiple regression method. The analysis addresses the following questions: (a) Are these 12 factors truly the critical success factors of Agile software development projects?; (b) If so, what is the relative importance of

* Corresponding author. Tel.: +1 714 952 5590.
E-mail address: dac-buu.cao@siemens.com (D.-B. Cao).

0164-1212/$ - see front matter
doi:10.1016/j.jss.2007.08.020

each factor when compared to other factors?; and (c) Is there a difference among those five factor categories in terms of their impact on the success of an Agile software development project?

2. Background

This section reviews briefly two key concepts, Agile method and Critical Success Factor (CSF). It is followed by a discussion of failure and success research on Agile projects. Finally, the research model is presented covering the research hypotheses.

2.1. Agile methods

The word "agile" by itself means that something is flexible and responsive, so agile methods implies its "[ability] to survive in an atmosphere of constant change and emerge with success" (Anderson, 2004, p. xxviii). This "maneuverability" in software business is a characteristic that is more important than ever these days since "deploying software to the Web has intensified software competition further than before" and "staying in business involves not only getting software out and reducing defects but tracking continually moving user and marketplace demands" (Cockburn, 2002, p. xxii). The official definition of Agile Software Development was contained in a form of "manifesto" in February 2001 by a group of 17 noted software process methodologists, who attended a summit meeting to advocate for a better way of developing software and then formed the Agile Alliance. The "Manifesto for Agile Software Development" posted on the Agile Alliance website (<http://www.agilemanifesto.org>) reads as follows:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

There are many software development methods that can be called "agile", and the list varies depending on different viewpoints, but in general the list in the literature includes Extreme Programming (XP), Scrum, Feature-Driven Development (FDD), Dynamic System Development Method (DSDM), Adaptive Software Development (ASD), Crystal, and Lean Software Development (LD).

2.2. The critical success factor approach

The Critical Success Factor (CSF) approach to identifying and measuring an organization's performance was first developed by Rockhart (1979) and later on refined and became well-established (Bullen and Rockhart, 1981; Rockhart and Crescenzi, 1984). CSF is defined by Bullen and Rockhart (1981) as "the limited number of areas in which satisfactory results will ensure successful competitive performance for the individual, department, or organization. CSF's are the few key areas where 'things must go right' for the business to flourish and for the managers goal to be attained" (p. 385).

As for software development project area, the CSF method has also been considered in recent studies. CSF's in software projects are found to relate to fundamental project management techniques (Reel, 1999), or to relate to the combination of software engineering and business strategy (Bytheway, 1999). Another case study finds that CSF's in software projects consists of various dimensions, from the development life cycle and estimation and validation to executive management, project management, and resource- and strategic-level planning (Bosghossian, 2002). In the context of the present study, the CSF's can be defined as the factors that must be present for the Agile project to be successful.

2.3. Success factors in agile software development projects

There has not been any formal study on CSF's in the Agile software development project per se, based on recent searches in peer reviewed academic literature or practitioner literature related to this topic. However, there are case studies and research theories on successes or failures/problems in agile implementation and agile software development projects. The review of both failures and successes in the literature will be beneficial in identifying the possible success factors in agile software development projects, as failures can contribute to the understanding of how to avoid certain serious pitfalls that are critical to the success of a project.

2.3.1. Failure research

Failure or Problem research is typically based on "lessons learned" from certain types of projects, but they are mostly similar enough to be generalized. Reel (1999) focuses more on generic software development projects and compiles 10 signs of software development project failure, at least seven of which are determined even before a design is developed or a line of code is written. Cohn and Ford (2003) study problems in transitioning organizations to agile processes, while Larman (2004) discusses in detail mistakes and misunderstandings occurred in agile projects. A research by Boehm and Turner (2005) emphasizes on management challenges in implementing agile projects, whereas a study by Nerur et al. (2005) covers problems not only in management aspect but also in people, process, and technology dimensions of migrating to agile projects.

Based on the above-mentioned literature, failures/problems can be classified into four categories: organizational, people, process, and technical, summarized in Table 1.

Table 1
Failure Factors

Dimension	Factor
Organizational	1. Lack of executive sponsorship 2. Lack of management commitment 3. Organizational culture too traditional 4. Organizational culture too political 5. Organizational size too large 6. Lack of agile logistical arrangements
People	7. Lack of necessary skill-set 8. Lack of project management competence 9. Lack of team work 10. Resistance from groups or individuals 11. Bad customer relationship
Process	12. Ill-defined project scope 13. Ill-defined project requirements 14. Ill-defined project planning 15. Lack of agile progress tracking mechanism 16. Lack of customer presence 17. Ill-defined customer role
Technical	18. Lack of complete set of correct agile practices 19. Inappropriateness of technology and tools

2.3.2. Success research

Success research cited in the literature is mostly based on case studies or meta-data or compilations and observations of agile projects and practices. Specifically, Highsmith (2002) reports from direct experience with agile implementations, while Schatz and Abdelshafi (2005) provide results from the Primavera case study, and Karlstrom and Runeson (2005) give insight from the Star-Gate case study. Other success researches which have a comparative flavor between traditional and agile methods include Boehm and Turner (2003), Augustine et al. (2005), and Ceschi et al. (2005). Some studies focus on agile implementation on large organizations or scaling of agile methods to large projects, such as Reifer et al. (2003), Lindvall et al. (2004), and Ambler (2006). Finally, Koch (2005) makes research compilation of a wide range of success factors of agile implementations.

Based on the above-mentioned literature, agile project success factors can be classified into five categories: organizational, people, process, technical, and project, summarized in Table 2.

Table 2
Success Factors

Dimension	Factor
Organizational	1. Strong executive support 2. Committed sponsor or manager 3. Cooperative organizational culture instead of hierarchal 4. Oral culture placing high value on face-to-face communication 5. Organizations where agile methodology is universally accepted 6. Collocation of the whole team 7. Facility with proper agile-style work environment 8. Reward system appropriate for agile
People	9. Team members with high competence and expertise 10. Team members with great motivation 11. Managers knowledgeable in agile process 12. Managers who have light-touch or adaptive management style 13. Coherent, self-organizing teamwork 14. Good customer relationship
Process	15. Following agile-oriented requirement management process 16. Following agile-oriented project management process 17. Following agile-oriented configuration management process 18. Strong communication focus with daily face-to-face meetings 19. Honoring regular working schedule – no overtime 20. Strong customer commitment and presence 21. Customer having full authority
Technical	22. Well-defined coding standards up front 23. Pursuing simple design 24. Rigorous refactoring activities 25. Right amount of documentation 26. Regular delivery of software 27. Delivering most important features first 28. Correct integration testing 29. Appropriate technical training to team
Project	30. Project nature being non-life-critical 31. Project type being of variable scope with emergent requirement 32. Projects with dynamic, accelerated schedule 33. Projects with small team 34. Projects with no multiple independent teams 35. Projects with up-front cost evaluation done 36. Projects with up-front risk analysis done

2.3.3. Success attributes

In terms of attributes of success, which depict the overall perception of success of a particular project, Cohn and Ford (2003) and Lindvall et al. (2004) suggest Quality (i.e. delivering a good working product), Scope (meeting all requirements by the customer), Timeliness (delivering on time), and Cost (within estimated cost and effort). These attributes can be summarized in Table 3 below.

Table 3
Success attributes

Dimension	Attribute
Overall perceived level of success	1. Quality (delivering good product or project outcome) 2. Scope (meeting all requirements and objectives) 3. Time (delivering on time) 4. Cost (delivering within estimated cost and effort)

2.3.4. Factor consolidation

From the two lists of possible factors (Tables 1 and 2) which may affect the success or failure of an agile software development project, a number of factors that share similar characteristics were consolidated into a reduced list of factors which cover 39 attributes.

Since this research is of exploratory nature, a reliability analysis is necessary so that each and every factor is ensured of a high level of reliability. Using reliability analysis, the researcher can determine the extent to which the items in each factor are related to each other. This can provide an overall index of internal consistency of the variables, and also can help single out problematic items that should be excluded from the variable and/or included in another variable. According to Rubin and Babbie (1997), "the most common and powerful method used today for calculating internal consistency reliability is coefficient alpha." Cronbach α is a coefficient alpha which is a direct function of both the number of items and their magnitude of inter-correlation, and is the lower bound to the test variance attributable to common factors among the items within each variable (Cronbach, 1951). For exploratory studies, it is agreed that a coefficient alpha level of 0.5 could be deemed acceptable (Nunally, 1967).

A reliability analysis was performed on all multi-item factors using the Cronbach's alpha method. After two rounds of reliability analysis, the number of factors that had Cronbach's alpha value below the acceptable level was reduced from 5 to 1.

One way to see if this factor could be reduced any further is to perform factor analysis on it (Williams and Monge, 2001). A principal component factor analysis with Varimax rotation was performed on this factor.

The final results revealed 12 factors were identified, which were translated into 12 main hypotheses, each linking its existence as a critical success factor to the success of the Agile software development project in terms of four success dimensions: Quality, Scope, Time, and Cost.

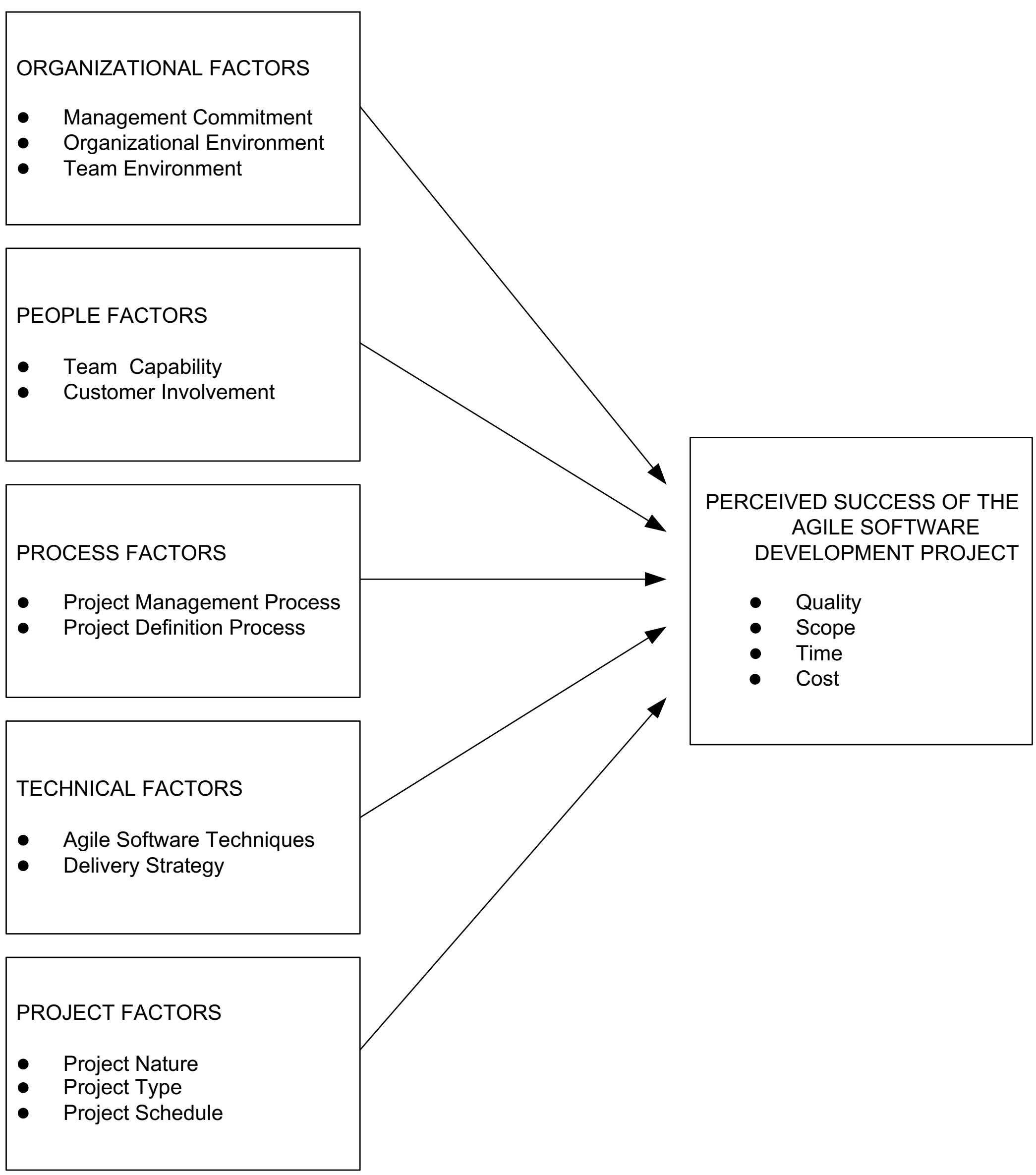

Fig. 1. The research model.

The success factors used in the hypotheses included: (a) Strong management commitment, (b) Agile-friendly organizational environment, (c) Agile-friendly project team environment, (d) High-caliber team capability, (e) Strong customer involvement, (f) Agile-style project management process, (g) Methodical project definition process, (h) Agile-style software engineering techniques, (i) Correct delivery strategy, (j) Non-life-critical project nature, (k) Variable-scope project type, and (l) Dynamic, accelerated project schedule.

The hypotheses were numbered 1 to 12, and since there were four success dimensions for each factor, the corresponding success dimensions were identified by the letters *a, b, c, d*. As a result, there were a total of 48 hypotheses, starting at 1*a* and ending at 12*d* (see Appendix A).The final list of 12 factors is depicted by the research model in Fig. 1.

3. Data collection

This study employed the web survey method to gather data. The target population was members of the Agile Alliance and its user groups. A web survey with Likert scale questionnaires and demographic information collection was distributed to the target population. There were four sections in the survey. The first section was on demographic data, which included both the respondent's demographic information as well as the agile project information. The second section was on success factors. To measure importance of success factors, a 7-point Likert scale was used to reflect the level of perception of the question by the respondent. The third section was on perception of success, and again, to measure perception of success of agile projects, a 7-point Likert scale was used to reflect the level of perception of the question by the respondent. In order to avoid ambiguity in terms of perception of success on the part of the respondent, the questions focused on one particular project of the respondent's choice in case he/she had been involved in multiple agile projects. The last section was for additional comments, where respondents were invited to enter any feedback or thoughts on a free-form text area, which might be used for follow-up for clarification if necessary.

As part of the pilot process to test content validity and readability, five members of the Agile Alliance supplied feedback on improving the survey. The feedback was incorporated into the survey before the survey invitation was emailed to the group coordinators of all 83 Agile Alliance user groups (42 in the Americas, 28 in Europe, 12 in Asia/Pacific, and one in Africa), as well as to the contact persons of all 60 corporate members of the Agile Alliance (29 from the Americas, 30 from Europe, and one from Asia/Pacific).

In all, after a six-week survey period, a total of 408 people responded by accessing the online survey and 109 projects were submitted with complete data. Table 4 displays the breakdown of agile methods used in the 109 projects submitted, while Tables 5–7 display the size, length, and location of the projects, respectively.

Table 4
Project profile – Agile methods used

Method	Frequency	Percent
Extreme programming	58	53.2
Scrum	23	21.1
Other	21	19.3
Feature driven development	5	4.6
Crystal	2	1.8
Total	109	100.0

Table 5
Project profile – Number of team members

Team members	Frequency	Percent
Less than 10	64	58.7
10–19	23	21.1
20–29	11	10.1
30–39	3	2.8
40–49	2	1.8
50–100	6	5.5
Total	109	100.0

Table 6
Project profile – Length of project in months

Months	Frequency	Percent
1–3	8	7.3
4–6	23	21.1
7–9	13	11.9
10–12	17	15.6
13–24	15	13.8
Over 24	33	30.3
Total	109	100.0

Table 7
Project profile – Project location (zone)

Location	Frequency	Percent
Europe	61	56.0
America	36	33.0
Asia/Pacific	10	9.2
Africa	2	1.8
Total	109	100.0

4. Data analysis and results

4.1. Multiple regression models

Since this research is an exploratory study to find out which factors can positively impact the success of an agile project, it is appropriate for a multiple regression analysis, where the relationship between multiple independent variables (success factors) and the dependent variable (agile project success) is determined, and where the relative predictive importance of the independent variables is established (Williams and Monge, 2001).

According to McClave and Benson (1988), the general multiple regression model, assuming that there are k independent variables, can be written as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots. + \beta_k x_k + \varepsilon$$

where y is the dependent variable and $x_1, x_2, \ldots, x_k$ are the independent variables, and β_i is the regression coefficient, and ε is the random error component. The value of the coefficient β_i determines the contribution of the independent variable x_i, given that the other x variables are held constant and β_0 is the y-intercept.

In the case of this study, the above translates to the following general equation:

$$Y(Q,S,T,C) = \beta_1 SF_1 + \beta_2 SF_2 + \beta_3 SF_3 + \beta_4 SF_4 + \ldots + \beta_{12} SF_{12}$$

where Y is Agile Project Success dependent variable, Q is the Quality dimension, S is the Scope dimension, T is the Time dimension, C is the Cost dimension, B_i is the Partial regression coefficient for the ith Success Factor (SF).

The multiple regression analysis was done on both levels – the full model and the optimized model. First, at the full model level, all 12 independent variables were entered into a regression model at the same time, with the expectation that the calculation of the coefficients would take into account the interaction of all other variables being present. In this case, the relative importance of each and every independent variable would be accounted for, and those variables that got top scores would be considered to be a critical success factor. Second, at the optimized model level, a stepwise regression screening procedure was carried out in order to come up with as few variables as possible while still predicting well the results of Agile projects. In this case, those variables that made it to the model would be considered to be a critical success factor, as they alone could account for the outcome of the dependent variables.

At each level described above, the multiple correlation coefficient (R) and the coefficient of determination (R^2) of the model were calculated, and for each independent variable, the coefficients B and β as well as the t-value were computed. Additionally, the significance level of each independent variable and the distribution normality of the dependent variable were checked. Those variables with top coefficient values that reached certain thresholds (i.e. significance level $p \leqslant 0.10$ for the full model and $p \leqslant 0.06$ for the optimized model) would be recognized as candidates for being critical success factors. Finally, the list of candidates from the two model approaches were compared and analyzed for inclusion in the list of critical success factors.

4.2. Summary of hypothesis testing results

From the above analyses, we finally arrived at the list of the most significant independent variables as shown in Table 8.

The results show that for Time and Cost, both model approaches arrived at the same conclusions, namely in each case, Team Capability and Delivery Strategy were selected as the most significant factors. For Scope, factors Agile Software Engineering Techniques and Delivery Strategy showed up in both approaches, but the stepwise optimized model approach yielded another factor, namely Customer Involvement. Finally, for Quality, each approach yielded one similar factor, which was Agile Software Engineering Techniques, while the second factor was different for each: Team Environment in the Full model case and Project Management Process in the Optimized model case.

With the above observations, the results of the hypothesis testing can be finalized as follows: out of 48 research hypotheses, a total of 10 hypotheses were supported, while the remaining 38 hypotheses were rejected. Those hypotheses were rejected due to their low coefficient values and high probability level for their corresponding null hypotheses, meaning the presence of those factors did not make a significant difference to the value of the success dimensions.

Table 9 summarizes the results of the hypothesis testing. The 10 supported hypotheses are labeled with a check mark (✔). Those without the check mark are rejected hypotheses.

Table 8
Summary of outcome from the two modeling approaches

	Quality		Scope		Time		Cost	
	Selected variable	β value	Selected variable	β value	Selected variable	β value	Selected variable	β value
Full model	Agile SW engineering techniques	0.39	Agile SW engineering techniques	0.24	Team capability	0.30	Delivery strategy	0.37
	Team environment	0.20	Delivery strategy	0.19	Delivery strategy	0.24	Team capability	0.23
Optimized model	Agile SW engineering techniques	0.46	Agile SW engineering techniques	0.27	Team capability	0.32	Delivery strategy	0.36
	Project management process	0.24	Delivery strategy	0.20	Delivery strategy	0.31	Team capability	0.17
			Customer involvement	0.20				

Table 9
Summary of hypothesis testing results

	Quality	Scope	Timeliness	Cost
1. Management commitment	H1*a*	H1*b*	H1*c*	H1*d*
2. Organizational environment	H2*a*	H2*b*	H2*c*	H2*d*
3. Team environment	**H3*a*✓**	H3*b*	H3*c*	H3*d*
4. Team capability	H4*a*	H4*b*	**H4*c*✓**	**H4*d*✓**
5. Customer involvement	H5*a*	**H5*b*✓**	H5*c*	H5*d*
6. Project management process	**H6*a*✓**	H6*b*	H6*c*	H6*d*
7. Project definition process	H7*a*	H7*b*	H7*c*	H7*d*
8. Agile software engineering techniques	**H8*a*✓**	**H8*b*✓**	H8*c*	H8*d*
9. Delivery strategy	H9*a*	**H9*b*✓**	**H9*c*✓**	**H9*d*✓**
10. Project nature	H10*a*	H10*b*	H10*c*	H10*d*
11. Project type	H11*a*	H11*b*	H11*c*	H11*d*
12. Project schedule	H12*a*	H12*b*	H12*c*	H12*d*

4.3. Answers to research questions

Based on the outcome of the regression analysis and the hypothesis testing results above, we now are in a position to answer the research questions posed at the beginning of the study. Although the success level was calculated for each of the four success attributes as opposed to the overall project success level, with each attribute carrying a distinct aspect to the perception of success, we can use the frequency of supported hypotheses to generalize the overall perception of success. Each of the research questions is answered in each sub-section below.

4.3.1. Research question 1

The first research question was, "Are these 12 factors truly the critical success factors of Agile software development projects?" From the findings discussed above, the answer is clearly No. Indeed, out of those 12 factors, only half of them were represented in the list of supported hypotheses. The factors that were candidates to be considered as critical success factors were:

1. Team Environment (in terms of Quality).
2. Team Capability (in terms of Timeliness and Cost).
3. Customer Involvement (in terms of Scope).
4. Project Management Process (in terms of Quality).
5. Agile Software Engineering Techniques (in terms of Quality and Scope).
6. Delivery Strategy (in terms of Scope, Timeliness, and Cost).

4.3.2. Research question 2

The second research question was, "What is the relative importance of each factor when compared to other factors?" Based on the hypothesis testing result, we can see that Delivery Strategy had the most hypotheses supported (three), followed by Agile Software Engineering Techniques and Team Capability (two each), and finally followed by Team Environment, Customer Involvement, and Project Management Process (one each). However, between Agile Software Engineering Techniques and Team Capability, the former was more important than the latter on the account of its higher Beta value. Likewise, among the last three factors, Project Management Process was more important than Team Environment and Customer Involvement by virtue of its higher Beta value. Table 10 provides the details below (for a list of attributes of these 6 success factors, please see Appendix B).

4.3.3. Research question 3

The third research question was, "Is there a difference among those five factor categories in terms of their impact on the success of an Agile software development project?" Again, based on the regression analysis and the hypothesis testing results, it was found that there was a marked difference among those five factor categories, namely Organizational, People, Process, Technical, and Project. It was evident the Technical dimension, which included Agile Software Engineering Techniques and Delivery Strategy, was the most critical in impacting the success of Agile projects, as it covered all four success dimensions. It was followed by the People dimension, which included Team Capability and Customer Involvement, as this dimension covered three success dimensions. The Organizational dimension and the Process dimension each touched one success dimension (Quality, in both cases). The only dimension which failed to make any impact at all was the Project dimension.

Table 10
Ranking of critical success factors

Rank	Factor	Hypotheses supported	Selection in the full model		Selection in the optimized model	
			Frequency	β value	Frequency	β value
1	Delivery strategy	H9*b*, H9*c*, H9*d*	3	0.37	3	0.36
				0.24		0.31
				0.19		0.20
2	Agile software engineering techniques	H8*a*, H8*b*	2	0.39	2	0.46
				0.24		0.27
3	Team capability	H4*c*, H4*d*	2	0.30	2	0.32
				0.23		0.17
4	Project management process	H6*a*	0	–	1	0.24
5	Team environment	H3*a*	1	0.20	0	–
6	Customer involvement	H5*b*	0	–	1	0.20

Table 11 summarizes the level of impact of the five factor dimensions on the perceived success of Agile software development projects based on the above discussion.

4.4. Research limitations

Based on the data collected from the survey, there are five limitations that need to be recognized in this research. First of all, the data did not represent all methods that were considered Agile. Indeed, only four out of seven methods were reported in the returned surveys, namely Extreme Programming (XP), Scrum, Feature-Driven Development, and Crystal. The other three (Dynamic System Development Method, Adaptive Software Development, and Lean Software Development) were not represented in the projects reported.

The second limitation is the possible bias toward XP in the data reported. As Table 4 demonstrates, XP projects occupied 53.2% of all projects reported, thus the findings might have been rather influenced by the way XP projects worked, with such practices as pair programming, refactoring, continuous integration, 40 h work week, etc.

The third limitation is the possibility of survey participants' subjective biases such as Agile proponents trying to claim Agile success in introductory projects (in order to promote the adoption of their methodology), and the lack of independent, non-Agile advocates in the survey.

The fourth limitation is the relatively low US-based projects representation in the sample population. The US user community had been the first to spread the Agile movement, and it is the most experienced as well as the most populous among Agile user communities worldwide. While in this study the US was the top country with a 20% representation of the projects, it may still be considered under-represented, as according to the Agile Alliance web site 35% of all Agile user groups (29 out of 83) are US-based, and its site demographic statistics shows over 58% of Agile Alliance users (1783 out of 3057) are from the US.

Lastly, the sample size was still small, considering the large Agile community population. A larger sample size could provide more robust and accurate statistical calculation and analysis, and also could include other Agile methods that were missing in this sample size. Although as many user groups as possible were contacted during the two email campaigns, the response rate of 2.40% was rather low.

Table 11
Impact level of factor dimensions on project success

Rank	Factor dimension	Hypotheses supported	Success dimension
1	Technical	H8*a*, H8*b*, H9*b*,H9*c*, H9*d*	Quality, Scope, Timeliness, Cost
2	People	H4*c*, H4*d*, H5*b*	Scope, Timeliness, Cost
3	Process	H6*a*	Quality
4	Organizational	H3*a*	Quality
5	Project	none	–

When interpreting the findings, the reader should be aware this study was done based on data that reflected a relatively immature state of the Agile development methods. As more and more organizations adopt Agile methods in their software development, it is anticipated that critical success factors may involve. It may be worthwhile to repeat such a study again in five and ten years to see whether any new factors may emerge or current key success factors become no longer critical.

4.5. Possible interpretations for Practitioners

Despite its limitations, this research has provided some interesting interpretations for practitioners. While the empirical analysis validates some long-held beliefs, it also provides some surprises. First of all, the findings agree with many of the 12 principles of agile practices laid down in the Agile Manifesto (Martin, 2003) that practitioners follow: the three top critical success factors identified by the study (Delivery Strategy, Agile Software Engineering Practices, and Team Capability) consist of many attributes covered in the Manifesto. Specifically, Delivery Strategy corresponds to the first and the third Manifesto practices – continuous delivery of valuable, working software in short time scales. Likewise, Agile Software Engineering Practices are in line with the ninth and tenth practices – continuous attention to technical excellence and simple design. Finally, Team Capability is parallel to the fifth practice, namely building projects around motivated individuals.

The other three auxiliary success factors identified by the study also correspond to several Manifesto practices: Project Management Process is related to the sixth and eight practices (face-to-face conversation within team, and sustaining a constant pace), while Team Environment has to do with the eleventh practice (self-organizing team), and Customer Involvement is comparable with the first and the fourth practices (satisfying the customer, and business people working closely with developers).

On the other hand, the study findings fail to support several common assumptions about Agile success factors. First of all, strong executive support and/or sponsor commitment are found to be a non-factor, contrary to the belief that strong executive support is critical to carry out an Agile project. Moreover, the assumption that Agile-style work facility, such as pair programming stations, communal areas, wall spaces, etc. which figure prominently in Agile books, are prerequisite for successful Agile project execution is not supported, either. This means that such work facilities while nice to have may not be critical. The project team may improvise the working environment to suit their needs without having to rigidly follow the exact physical set-up as suggested in Agile literature.

Perhaps the most interesting finding is that the Organizational Environment factor had a negative coefficient in both the regression analyses of the Timeliness and Cost dependent variables. This seems to suggest that Agile projects could be done in a timely manner and within cost despite the lack of agile-friendly organizational environment factors, which include cooperative culture, oral culture, universal acceptance of Agile, appropriate reward system, etc. This finding can be explained by the fact that Agile is relatively new and those agile-friendly organizational traits have not typically been introduced or entrenched in the organizations where Agile projects were carried out; thus, as long as the team is capable and has a correct delivery strategy, a widely-accepted agile environment is not a prerequisite for success in terms of Timeliness and Cost.

In terms of the impact of the five factor categories on Agile project success, the research findings show a rather surprising unevenness in the distribution of supported hypotheses among the categories. While the Technical dimension and the People dimension weigh heavily as the most important categories, the Process dimension and the Organizational dimension are underrepresented in their contribution to the Agile project success. Most notable is the complete lack of contribution of the Project dimension in the list of identified success factors, although there had been ample discussion of this dimension in the literature. This seems to suggest that project managers, when deciding whether to go Agile for their software projects, may not have to put too much weight on factors such as the project nature, project type, or project schedule, thus broadening the applicability of Agile development methods.

5. Conclusions

This research study set out to use survey data to explore the critical success factors of Agile software development projects using quantitative methods. The data collected from 109 Agile projects from a diverse group of organizations of various sizes, industries, and geographic locations provided enough empirical information for statistical analysis to arrive at a number of conclusions.

First of all, in spite of a large number of factors affecting Agile projects discussed in the literature, the actual number of critical success factors found here is quite small. Out of 48 research hypotheses, only 10 are supported. Through multiple regression analysis, the only factors that could be called critical success factors are found to be (a) a correct delivery strategy, (b) a proper practice of Agile software engineering techniques, and (c) a high-caliber team. Three other factors that could be critical to certain success dimensions are found to be (a) a good Agile project management process, (b) an Agile-friendly team environment, and (c) a strong customer involvement.

The study results have failed to find evidence that some assumed prerequisites for success of Agile projects such as strong executive support, strong sponsor commitment, ready availability of physical Agile facility, or Agile-appropriate project types, etc. are actually critical factors for success.

The key contribution of this research is to reduce a multitude of anecdotal success factors to three critical ones based on survey data analysis. As long as the Agile project picks a high-caliber team, practices rigorous Agile software engineering techniques and executes a correct Agile-style delivery strategy, the project could be likely to be successful. It provides a focus for the management when they embark on adopting Agile methods in their software development projects.

Appendix A

For testing purpose, the factors in the research model were used for forming the corresponding 12 main hypotheses, each addressing four success dimensions. Some of these hypotheses have been reworded to properly represent the various attributes that the literature review indicated as possible success factor items:

1. Hypotheses related to the Organization dimension:

H 1. The existence of a strong management commitment is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

H 2. The presence of agile-friendly organizational environment is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

H 3. The existence of agile-friendly project team environment is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

2. Hypotheses related to the People dimension:

H 4. Having a team of high caliber is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

H 5. Having a strong customer involvement is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

3. Hypotheses related to the Process dimension:

H 6. The practice of agile project management process is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

H 7. The practice of a methodical project definition process is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

4. Hypotheses related to the Technical dimension:

H 8. The practice of agile software engineering techniques is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

H 9. The execution of a correct delivery strategy is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

5. Hypotheses related to the Project dimension:

H 10. Limiting only to non-life-critical projects is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

H 11. Limiting only to projects of variable scope with emergent requirements is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

H 12. Limiting only to projects with dynamic, accelerated schedule is a critical success factor that contributes to the successful agile software development projects in terms of (a) Quality, (b) Scope, (c) Time, and (d) Cost.

Appendix B

Summary of attributes of the 6 success factors found in the study

Rank	Factor	Attributes
1	Delivery strategy	•Regular delivery of software •Delivering most important features first
2	Agile software engineering techniques	•Well-defined coding standards up front •Pursuing simple design •Rigorous refactoring activities •Right amount of documentation •Correct integration testing
3	Team capability	•Team members with high competence and expertise •Team members with great motivation •Managers knowledgeable in agile •Managers who have adaptive management style •Appropriate technical training to team
4	Project management process	•Following agile-oriented requirement management process •Following agile-oriented project management process •Following agile-oriented configuration management process •Good progress tracking mechanism •Strong communication focus with daily face-to-face meetings •Honoring regular working schedule
5	Team environment	•Collocation of the whole team •Coherent, self-organizing teamwork •Projects with small team •Projects with no multiple independent teams
6	Customer involvement	•Good customer relationship •Strong customer commitment and presence •Customer having full authority

References

Ambler, S.W., 2006. Supersize me. Software Development 14 (3), 46–48.

Anderson, D.J., 2004. Agile Management for Software Engineering. Prentice Hall, Upper Saddle River, New Jersey.

Augustine, S., Payne, B., Sencindiver, F., Woodcock, S., 2005. Agile project management: steering from the edges. Communications of the ACM 48 (12), 85–89.

Boehm, B., Turner, R., 2003. Using risk to balance agile and plan-driven methods. Computer 36 (6), 57–66.

Boehm, B., Turner, R., 2005. Management challenges to implement agile processes in traditional development organizations. IEEE Software 22 (5), 30–39.

Bosghossian, Z.J., 2002. An investigation into the critical success factors of software development process, time, and quality, Ph.D. Thesis, Pepperdine University, Malibu, California.

Bullen, C.V., Rockhart, J.F., 1981. A primer on critical success factors (Working Paper No. 69), Massachusetts Institute of Technology, Sloan School of Management, Center for Information Systems Research, Cambridge, Massachusetts.

Bytheway, A.J., 1999. Successful software projects and how to achieve them. IEEE Software 16 (3), 15–17.

Ceschi, M., Sillitti, A., Succi, G., Panfilis, S.D., 2005. Project management in plan-based and agile companies. IEEE Software 22 (3), 21–27.

Cockburn, A., 2002. Agile Software Development. Addison-Wesley, Boston, Massachusetts.
Cohn, M., Ford, D., 2003. Introducing an agile process to an organization. Computer 36 (6), 74–78.
Cronbach, L.J., 1951. Coefficient alpha and the internal structure of tests. Psychometrika 16, 297–334.
Highsmith, J., 2002. Agile Software Development Ecosystems. Addison-Wesley, Boston, Massachusetts.
Karlstrom, D., Runeson, P., 2005. Combining agile methods with Star-Gate project management. IEEE Software 22 (3), 43–49.
Koch, A.S., 2005. Agile Software Development: Evaluating the Methods for Your Organizations. Artech House, Northwood, Massachusetts.
Larman, C., 2004. Agile & Iterative Development. Addison-Wesley, Boston, Massachusetts.
Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., et al., 2004. Agile software development in large organizations. Computer 37 (12), 26–34.
Martin, R.C., 2003. Agile Software Development: Principles, Patterns, and Practices. Prentice Hall, Upper Saddle River, New Jersey.
McClave, J.T., Benson, P.G., 1988. Statistics for Business and Economics. Dellen Publishing, San Francisco, California.
Nerur, S., Mahapatra, R.K., Mangalaraj, G., 2005. Challenges of migrating to agile methodologies. Communications of the ACM 48 (5), 72–78.
Nunally, J., 1967. Psychometric Theory. McGraw Hill, New York.
Reel, J.S., 1999. Critical success factors in software projects. IEEE Software 16 (3), 18–23.
Reifer, D.J., Maurer, F., Erdogmus, H., 2003. Scaling agile methods. IEEE Software 20 (4), 12–14.
Rockhart, J.F., Crescenzi, A.D., 1984. Engaging top management in information technology. Sloan Management Review 25 (4), 3–16.
Rubin, A., Babbie, E., 1997. Research Methods for Social Work. Brooks/Cole Publishing Company, Pacific Grove, California.
Schatz, B., Abdelshafi, I., 2005. Primavera gets agile: A successful transition to agile development. IEEE Software 22 (3), 36–42.
Williams, F., Monge, P., 2001. Reasoning with Statistics. Thomson Wadsworth, Belmont, California.

Tsun Chow is an IT Management Faculty at Capella University. Dr. Chow's research interests cover a variety of IT management issues including outsourcing, information assurance and process re-engineering. Before joining Capella, Dr. Chow held many senior IT management positions in a Fortune 100 company, covering data center management, enterprise network management, application development, process re-engineering and outsourcing management. He has published research articles in software development, quality assurance and artificial intelligence. He received his PhD in Computer Science and Electrical Engineering from University of California at Berkeley.

Dac-Buu Cao is currently a Director of Product Development & Support Systems at Siemens PLM Software, a division of Siemens Automation & Drives Group. His research interests include software quality and software engineering methodologies. Dr. Cao graduated in Computer Science from University of California at Irvine and received his PhD in IT Management from Capella University. He is a member of the IEEE and the ACM.