Received 23 June 2015
Revised 19 July 2016
Accepted 25 July 2016

Empirical comparison of traditional plan-based and agile methodologies

Critical success factors for outsourced software development projects from vendors' perspective

Arthur Ahimbisibwe
Makerere University Business School, Makerere University Kampala, Kampala, Uganda, and
Urs Daellenbach and Robert Y. Cavana
School of Management, Victoria University of Wellington, Wellington, New Zealand

Abstract

Purpose – Aligning the project management methodology (PMM) to a particular project is considered to be essential for project success. Many outsourced software projects fail to deliver on time, budget or do not give value to the client due to inappropriate choice of a PMM. Despite the increasing range of available choices, project managers frequently fail to seriously consider their alternatives. They tend to narrowly tailor project categorization systems and categorization criterion is often not logically linked with project objectives. The purpose of this paper is to develop and test a contingency fit model comparing the differences between critical success factors (CSFs) for outsourced software development projects in the current context of traditional plan-based and agile methodologies.

Design/methodology/approach – A theoretical model and 54 hypotheses were developed from a literature review. An online Qualtrics survey was used to collect data to test the proposed model. The survey was administered to a large sample of senior software project managers and practitioners who were involved in international outsourced software development projects across the globe with 984 valid responses.

Findings – Results indicate that various CSFs differ significantly across agile and traditional plan-based methodologies, and in different ways for various project success measures.

Research limitations/implications – This study is cross-sectional in nature and data for all variables were obtained from the same sources, meaning that common method bias remains a potential threat. Further refinement of the instrument using different sources of data for variables and future replication using longitudinal approach is highly recommended.

Practical implications – Practical implications of these results suggest project managers should tailor PMMs according to various organizational, team, customer and project factors to reduce project failure rates.

Originality/value – Unlike previous studies this paper develops and empirically validates a contingency fit model comparing the differences between CSFs for outsourced software development projects in the context of PMMs.

Keywords Critical success factors, Agile methodology, Outsourced software development projects, Traditional plan-based methodology, Vendors' perspective

Paper type Research paper

1. Introduction

Projects are a frequent activity in various organizations which invest substantial resources to drive innovation and change (Sauser *et al.*, 2009; Shenhar and Dvir, 2007). However, evidence indicates that many software projects fail to deliver on time or budget and do not give value (PMI, 2013a; KPMG, 2013). According to Shenhar (2008, p. 15) nearly two-thirds of software projects do not meet their time and budget goals, and often do not meet their business objectives. While there are many reasons proposed for why projects are not successful, numerous studies indicate that software projects continue to fail due to

Journal of Enterprise Information Management
Vol. 30 No. 3, 2017
pp. 400-453

1741-0398
DOI 10.1108/JEIM-06-2015-0056

inappropriate choice of a project management methodology (PMM) (Sauser *et al.*, 2009). According to Sheffield and Lemétayer (2013), there are currently several alternative PMMs that often make it difficult to determine the best option. Boehm and Turner (2004) maintain that users and software developers tend to stick to what they are good at and favour the project management methods with which they have had most experience. Consequently, although available methodology choices have increased substantially, project managers appear to rarely consider their alternatives available, simply adopting categorization criteria that are not rationally linked with project objectives (Howell *et al.*, 2010).

Despite the existence of communities of methodology practice such as PRINCE2 (Office of Government Commerce (OGC), 2009), PMI (2013b) and Agile Alliance (2001) that promote best practice, software development projects continue to fail (Standish Group International Inc., 2012). These communities pursue different and contradictory goals; however the conditions and critical success factors (CSFs) favouring a PMM to achieve project success are not well understood (Ahimbisibwe *et al.*, 2015). Yet, it is argued that aligning the PMM to a particular project is considered to be essential for project success (Wysocki, 2009, 2011; Sauser *et al.*, 2009; Sheffield and Lemétayer, 2013).

Tiwana and Keil's (2004, p. 74) study of 720 software projects found that the use of an inappropriate methodology is actually the most critical risk driver for project failure. Therefore, matching the project type and the software development approach would be expected to increase the chances of project success. Howell *et al.* (2010, p. 256) further suggest that the lack of a decision support tool and theory connecting project types and project methodology discourages project managers from considering alternative methodologies. To address this gap, this paper develops and tests a contingency fit model comparing the differences between CSFs for outsourced software development projects in the current context of PMMs, from the perspective of vendors. This study contributes to a body of knowledge which seeks to understand why software development projects succeed or fail, and how project success might be improved.

This paper is organized as follows. The next section consists of a literature review and development of hypotheses. The methodology used in the study is outlined next. This is followed by the results of the structural equation modeling (SEM) analysis and a discussion of hypothesis tests. The paper concludes by discussing the contributions of the research, limitations and future research directions.

2. Literature review and hypothesis development

2.1 PMMs – two major approaches

Scholars distinguish traditional (heavy) and agile (light) approaches to project management as two broad categories for choice (e.g. Boehm and Turner, 2003; Charvat, 2003; Highsmith, 2010; Wysocki, 2009, 2011). Traditional methodologies rely on a linear or incremental life cycle (OGC, 2009; PMI, 2013b). These methodologies are plan driven and are characterized by a requirement/design/build approach to development (Boehm and Turner, 2004). In this kind of project, the requirements are clearly specified and little change is expected. Thus, the environment is predictable and planning tools can be used to optimize the management of the project (Vinekar *et al.*, 2006). These approaches are usually change-resistant and focus on compliance to plan as a measure of success (Wysocki, 2009). Consequently, they are somewhat prescriptive, and heavy on process and documentation (Sheffield and Lemétayer, 2013). According to testimonial data gathered from 10,000 project managers (Wysocki, 2009, p. 328), no more than 20 per cent of all projects have the characteristics of traditional projects, but project managers continue to apply these traditional methods on projects for which they are not suited.

On the other hand, agile methodologies that appear to respond to the dynamic aspects of the environment have emerged. Agile methodologies promise increased customer satisfaction with lower defect rates, faster development times for solutions to rapidly changing requirements but are not clearly developed and are poorly understood (Vinekar *et al.*, 2006). Agile methodologies have gained popularity because organizations need short delivery cycles to cope with uncertainty and rapid change in requirements. Agile methodologies are based on an iterative or adaptive life cycle and are designed to accept and embrace change (Wysocki, 2011). They are value driven rather than plan driven and use tacit knowledge between team members in place of heavy documentation (Ramesh *et al.*, 2012). This principle is enshrined in the Agile Alliance (2001). In agile methods, the major, upfront, one-time planning task is replaced by an iterative and adaptive series of just-in-time tasks each of which is executed only when needed. This provides flexibility and adaptability to the project, enabling it to cope more readily with change requests. Despite exhortation to move away from old practices, it has also been cautioned that the new methodologies are not silver bullets that guarantee success every time (Boehm and Turner, 2004).

PMM can be used for many different types of projects including software development, and can be used in conjunction with different software development methodologies (SDMs). Prince2 and PMI are PMMs and not SDMs as such, although they may be used to manage and control software projects. However, agile methodologies by contrast are firmly aimed only at software development. According to Dalcher and Brodie (2007), it is important to ensure that the PMMs and SDMs are compatible or fit to achieve project success. Since SDMs must be well aligned with PMM to achieve project success (Sheffield and Lemétayer, 2013), PMM is used as a proxy for methodology in this study (Charvat, 2003, p. 3; Cockburn, 2007, p. 149).

The debate of superiority of one PMM over the other continues; however, neither appears to be a perfect fit for all types of software development projects (Shenhar, 2001; Wysocki, 2009). According to Shenhar (2001), "one size does not fit all". Instead, project characteristics define the extent to which a particular PMM may be suitably applied. Wysocki (2009) further explains that the best development methodology is based on both the project characteristics and the business and organizational environment in which the project is conducted. Potentially, in analysing empirical data, the detection of fit/misfit may help better explain project success/failure. More important, understanding the elements of such fit/misfit may provide recommendation for a preferred managerial approach before a project is launched, or for bringing a troubled project back on track.

2.2 Contingency theory and "fit" in software development

Project contingency theory argues that projects have different characteristics, and therefore they should not all be structured and managed the same way (Howell *et al.*, 2010, p. 256). The contingency theory is used in the current study to relate development methodologies and CSFs; and the "contingency" here is to do with the "fit" between the CSFs and the method. According to Sauser *et al.* (2009, p. 666), a contingency approach to project management necessarily investigates the extent of fit or misfit between project characteristics and PMM. This is consistent with the research examining enduring organizations and contingency theory that suggests that organizational effectiveness is dependent upon the organization's ability to adapt to the environment, and that there is a need for congruence between the environment and structure (Miles and Snow, 1978). It has often been suggested that more turbulent environments should be addressed by organic structures because coping with uncertainty is a core problem for complex organizations (Thompson, 1967).

2.3 The concept of project success

This paper focusses on process and product performance as two key dimensions of project success (Jun *et al.*, 2011; Nidumolu, 1996; Wallace *et al.*, 2004). Process success refers to the extent to which a project is delivered on schedule, scope and within budget. Product success refers to the quality of resulting system. It is important to study both aspects of project success because there is a potential conflict between the efficiency of the process and its quality.

Recent empirical studies by Zwikael *et al.* (2014) and Zwikael and Smyrk (2012) indicate that the influence of CSFs differs over different success measures. This categorization is aligned with the approach undertaken by Pinto and Prescott (1988), who made a distinction between the implementation process (efficiency) and the perceived "value" of the project (effectiveness).

2.4 CSFs for software development projects

CSFs are issues that if addressed appropriately, substantially increase the likelihood of project success (Nasir and Sahibuddin, 2011, p. 2175). However, the project management literature remains unclear about which CSFs affect software project success in the context of methodologies, and for various success measures. No empirical study compares CSFs according to PMMs. Most of the research has focussed on the outcomes of software development projects rather than the process of developing software itself. Yet, the process is argued to influence the outcomes (Nasir and Sahibuddin, 2011). Zwikael and Globerson (2006) propose a dynamic shift from CSFs to critical success processes but the debate remains unresolved.

Unlike other engineering industries, software projects have unique characteristics. Prior research has largely focussed on in-house software development projects where developers and users are members of the same organization, and may be more constrained in their choice of PMM due to available skills and resources. However, the trend over the last decade shows an increasing tendency for firms to outsource their software development projects activities (Sabherwal, 2003; Taylor, 2007; Jun *et al.*, 2011), which could help to mitigate the mismatch between project characteristics and CSFs.

Table I presents a summarized list of CSFs sourced from the subset of the critical factors in project success literature that explicitly addresses PMMs and process design issues in software development projects. Consistent with previous studies, published research was reviewed consisting of case studies, surveys and theoretical studies, covering different project sizes in various domains and multiple countries. The only major difference is that this current research only considered and opted for software development projects contrasting traditional plan driven and agile approaches. Most of the previous studies had included all IT projects with both non-software and software project activities in varying proportions, respectively.

While many studies that have been carried out for more than 30 years to establish CSFs for software development projects, there is only limited agreement on what CSFs are. Based on the examination of literature, it appears that the most CSFs cited across 148 publications are dominated by factors related to top level management strategic decisions and organizational culture. This is followed by factors that seem to belong to issues of project teams and then factors that relate to the customer (client). The last portion of factors ranked seems to belong to project uncertainty (risk)/technical factors. Some studies have also mixed CSFs with key performance indicators (success criteria) thereby creating more confusion.

In this study we will be empirically testing hypotheses related to the CSFs outlined in Table I, which draws on the research contained in Ahimbisibwe *et al.* (2015).

2.5 Hypotheses development

This study focusses on the selection of PMMs that can be adopted to fit various CSFs to manage outsourced software development projects to success from the perspective of vendors. Because of different and conflicting results on the impact of each CSF for various

Category	Critical success factor[a]	Ranking[b]	Hypotheses[c]
Organizational factors	Top level management support	1	*H1*
	Organizational culture	4	*H2*
	Level of project planning	5	*H3*
	Monitoring and controlling	8	*H3*
	Vision and mission	7	*H4*
	Change management skills	9	*H5*
	Leadership characteristics	6	*H6*
Team factors	Internal project communication	10	*H7*
	Project team commitment	3	*H8*
	Project team's general expertise	20	*H9*
	Project team's expertise with the task	19	*H9*
	Project team's experience with SDM	25	*H9*
	Project team's composition	16	*H10*
Customer factors	User participation	2	*H11*
	User support	11	*H12*
	Customer training and education	17	*H13*
	Client experience	18	*H13*
Project uncertainty factors	Technological uncertainty	12	*H14*
	Project complexity	14	*H15*
	Relative project size	23	*H16*
	Specifications changes	24	*H17*
	Project criticality	26	*H18*

Notes: [a]Success criteria mixed with CSFs is excluded while other omitted CSFs are represented through constituent factors; [b]rankings are based on number of occurrences in the considered literature; [c]the hypotheses are outlined in Section 2.5
Source: Based on Table IV in Ahimbisibwe *et al.* (2015, p. 19)

Table I. Critical success factors for software development projects' success

PM methodologies, two direct hypotheses and one comparative hypothesis are formulated for each CSF:

(1) the first hypothesis, e.g. *H1(T)*, relates the effect of the CSF on the two major project success measures (i.e. project success) for traditional (T) plan-based methodologies;

(2) the second hypothesis, e.g. *H1(A)*, relates the effect of the CSF on the two major project success measures (project success) for agile (A) methodologies; and

(3) the third hypothesis, e.g. *H1(TA)*, compares the contribution or relationship between a CSF and project success measures depending on whether it is assumed to be relatively greater or less important, or not significantly different, depending on the methodology that is implemented.

Note that in the following hypotheses, in the first two hypotheses for each CSF, "project success" contains two sub-hypotheses related to "(i) process success and (ii) product success".

2.5.1 Organizational factors and software development project success. Top management support (TMS). Amongst all organizational factors, TMS has been proposed to be the primary success factor for software development projects. This is probably because TMS drives and influences other organizational factors. Jung *et al.* (2008) found top level management commitment significantly and positively influences project performance. Wan and Wang (2010) found TMS significantly contributed to agile software development process improvement. Similarly, Nah and Delgado's (2006) case-based research of CSFs for ERP planning implementation and upgrade in USA found TMS positively influenced project success.

In contrast, however, Chow and Cao (2008) did not find a strong management commitment as a significant CSF that contributes to the successful agile software development projects. Although there is unanimous agreement in the PM literature regarding the great effectiveness of

TMS, some authors underplay its role in projects. Young and Jordan (2008) argue that what begins as support easily turns into interference, which is harmful especially in highly innovative environments. Swink (2000) found that TMS did not matter in really new product innovations. Although the relationship of TMS on project success for each methodology is not clear from the literature, the use and success of agile methodologies appears to be a major initiative in most companies that requires more TMS. Thus, the following hypotheses can be derived:

H1(T). A significant positive relationship between TMS and project success for traditional plan-based methodologies ($T+$).

H1(A). A significant positive relationship between TMS and project success for agile methodologies ($A+$).

H1(TA). TMS is more important for agile compared with traditional plan-based methodologies ($A+ > T$).

Organizational culture (OC). The literature suggests that OC, in particular risk taking attitude, has a positive effect on the extent to which agile projects succeed (Misra *et al.*, 2009; Sheffield and Lemétayer, 2013). However, some conflicting results have been reported about the effect of agile OC on software development project success. For instance, while Misra *et al.* (2009) and Wan and Wang (2010) found agile corporate culture influences project success, Chow and Cao (2008) did not find a significant effect of agile OC on the perceived project success. Strode *et al.* (2009) found a positive and significant relationship between low formality organizations and the use of agile methodologies. Iivari and Huisman (2007) found a significant positive relationship between hierarchical rational organizations and the deployment of traditional methodologies. Since agile is considered to be a cultural issue and flexible or risk taking cultures can more easily accommodate adventurous methodologies like agile. Thus:

H2(T). No significant relationship between flexible OC and project success for traditional plan-based methodologies ($T\#$).

H2(A). A significant positive relationship between flexible OC and project success for agile methodologies ($A+$).

H2(TA). Flexible OC is more important for agile compared with traditional plan-based methodologies ($A+ > T$).

Planning and controlling (P&C). Project P&C refer to the extent to which P&C practices are used in a project. Previous research has demonstrated a positive relationship between planning and process performance (Yetton *et al.*, 2000; Jun *et al.*, 2011). Poor planning is likely to be associated with inefficiencies in development and, thus, lead to large budget and time variances. Rigorously tracking and monitoring a project according to a project plan can ensure that the final product is delivered within budget and on schedule. However, surprisingly, Misra *et al.* (2009) did not find a significant relationship between in-formalized plans and agile projects success. Yet, it argued that plans and controls easily become obsolete for agile projects since change usually occurs faster than they can be updated (Ramesh *et al.*, 2012; Sheffield and Lemétayer, 2013). Thus:

H3(T). A significant positive relationship between project P&C and project success for traditional plan-based methodologies ($T+$).

H3(A). No significant relationship between project P&C and project success for agile methodologies ($A\#$).

H3(TA). Project P&C is more important for traditional plan-based compared with agile methodologies ($T+ > A$).

Vision and mission (V&M). Jung *et al.* (2008) found that clear V&M directly impacted on the continuous improvement of international project management. Similarly, Nah and Delgado (2006) found clear business plans and vision that were well aligned with project objectives led to project success. Again, since V&M are closely associated with rigorous long-term P&C, agile projects might not require emphasis on the V&M due to dynamic changes in the software project environment. Thus:

H4(T). A significant positive relationship between V&M and project success for traditional plan-based methodologies (T+).

H4(A). No significant relationship between V&M and project success for agile methodologies (A#).

H4(TA). V&M is more important for traditional plan-based compared with agile methodologies ($T+ > A$).

Change management (CM). Jung *et al.* (2008) found that effective CM process positively and significantly impacted on project performance. Similarly, Nah and Delgado (2006) found that CM was essential for successful software implementation throughout the project life cycle. Since agile methodologies deal with dynamic changes and uncertainty management than traditional projects, it can be theorized that:

H5(T). A significant positive relationship between CM and project success for traditional plan-based methodologies (T+).

H5(A). A significant positive relationship between CM and project success for agile methodologies (A+).

H5(TA). CM skills are more important for agile compared with traditional plan-based methodologies ($A+ > T$).

Leadership characteristics (LC). Sheffield and Lemétayer (2013) and Strode *et al.* (2009) found that LC positively influences agile software development project success. Similarly, Wan and Wang (2010) indicate that top level leadership which supports an agile culture with a clear vision for education and training significantly and positively impact agile software project success. Agile methodology studies appear to emphasize that better LC (e.g. innovative and entrepreneurial) positively impact more on agile project success. Thus:

H6(T). A significant positive relationship between client's LC and project success for traditional plan-based methodologies (T+).

H6(A). A significant positive relationship between client's LC and project success for agile methodologies (A+).

H6(TA). LC are more important for agile compared with traditional plan-based methodologies ($A+ > T$).

2.5.2 Team factors and software development project success. Internal project communication (IPC). IPC is defined as the practices that increase information exchange and cohesion among development team members. IPC enhances the levels of information sharing and collaboration between the members of the project team which decreases the amount of team conflict and keeps the team stable. Along similar lines, Jun *et al.* (2011) and Sudhakar (2012) found that IPC had a significant positive effect on both process and product performance. Similarly, Yetton *et al.* (2000) demonstrated that project team conflict leads to instability in a project team and, thus, result in a project being delayed and exceeding budget. This is consistent with Jun *et al.* (2011), Misra *et al.* (2009) and Chow and Cao (2008) who found that those workers with a positive attitude about project tasks carry out certain role behaviours well beyond the basic minimum levels required of them. Agile methodologies tend to require more face-to-face communication and short meetings within teams and continuous interactions with management than traditional plan-based methodologies do. Thus:

H7(T). A significant positive relationship between IPC and project success for traditional plan-based methodologies ($T+$).

H7(A). A significant positive relationship between IPC and project success for agile methodologies ($A+$).

H7(TA). IPC is more important for agile compared with traditional plan-based methodologies ($A+ > T$).

Project team commitment (PTC). PTC is the willingness by a team to devote energy and loyalty to a project as expressed in three forms: affective, continuance and normative (Meyer and Allen, 1997). Affective commitment is a team's emotional attachment with the project. Continuance commitment refers to the team's recognition of the benefits of continued association with the project compared to the perceived cost of leaving the project. Normative commitment refers to the team's feeling of obligation to remain in the project. All these three forms of commitment affect the team members' willingness to remain with a project and their work-related behaviour. Chow and Cao (2008) found that team members with great motivation positively influenced the perceived success of the agile software development projects. Correspondingly, Wan and Wang (2010) found significant positive relationships between team commitment and agile project success. This suggests that committed project team members more often do not have intentions to quit, which saves the project the costs of recruiting and orienting new members in terms of both time and money. Similarly, costs of supervision are mitigated if the project team members are committed to their project tasks. However, agile projects success might require a higher level of team commitment than traditional plan-based projects because of changes that must be done under various tighter constraints. Thus:

H8(T). A significant positive relationship between PTC and project success for traditional plan-based methodologies ($T+$).

H8(A). A significant positive relationship between PTC and project success for agile methodologies ($A+$).

H8(TA). PTC is more important for agile compared with traditional plan-based methodologies ($A+ > T$).

Development team's expertise (DTE). DTE has been found to influence software project success (Wan and Wang, 2010; Misra *et al.*, 2009; Chow and Cao, 2008; Jiang and Klein, 2000; Boehm and Turner, 2003; Ratbe *et al.*, 2000; Lindavall *et al.*, 2002; Little, 2005; Sudhakar, 2012). Agile methodologies require more senior and skilled staff to continuously adapt to change (Cockburn and Highsmith, 2001). If a team lacks expertise, it is more appropriate to choose a traditional plan-based methodology with more structure or plan and pre-identified processes to correctly guide the project team members. Thus:

H9(T). A significant positive relationship between DTE and project success for traditional plan-based methodologies ($T+$).

H9(A). A significant positive relationship between DTE and project success for agile methodologies ($A+$).

H9(TA). DTE is more important for agile compared with traditional plan-based methodologies ($A+ > T$).

Project team composition (PTC). Chow and Cao (2008) found that cross-functional teams which involved customers contributed to successful agile software development projects in terms of scope but not time, cost and quality. Similarly, Nah and Delgado (2006) found that software projects with best people on the team, made up of balanced cross-functional teams and working on a full-time basis were successfully implemented. Equally, Nah and Delgado (2006) established that teams composed of business, technical knowledge and consultants that were empowered in decision making had completed projects on time, budget, scope and quality. Agile methodologies require small teams that are autonomous and self-organizing with best skilled expertise and experience than traditional plan-based methodologies (Sheffield and Lemétayer, 2013). Thus:

H10(T). A significant positive relationship between PTC of best people and project success for traditional plan-based methodologies (T+).

H10(A). A significant positive relationship between PTC of best people and project success for agile methodologies (A+).

H10(TA). PTC is more important for agile compared with traditional plan-based methodologies ($A+ > T$).

2.5.3 Customer factors and software development project success. User participation (UP). UP and support consists of the behaviours and activities of the customer in relation to product development (Jun *et al.*, 2011). The literature reveals that UP significantly increases the likelihood of the chances of software development projects success. Empirical studies by Chow and Cao (2008), Misra *et al.* (2009) and Sheffield and Lemétayer (2013) indicate significant and positive relationship between UP and support and agile software development project success. Although it can be argued that UP tends to increase budget variance by encouraging suggestions for changes to specification, Yetton *et al.* (2000) found that UP also decreases budget variance by managing expectations and quickly resolving potential problems. Likewise, Jun *et al.* (2011) also demonstrated that resolving potential conflicts early arising from greater UP plays a vital role in the perceived system satisfaction of software developers and users. Agile projects require onsite and full time UP from specification phase up to the end. In contrast, however, traditional plan-driven projects do not necessarily require UP once requirements are clearly specified. Thus:

H11(T). No significant relationship between UP and project success for traditional plan-based methodologies (T#).

H11(A). A significant positive relationship between UP and project success for agile methodologies (A+).

H11(TA). UP is more important for agile compared with traditional plan-based methodologies ($A+ > T$).

User support (US). Yetton *et al.* (2000) found that customer support increases the possibility that the project is completed on time and not redefined or abandoned but can also increase budget variances. Equally, Chow and Cao (2008) found that having a strong customer support contributes to the successful agile software development projects in terms of scope but not time, cost and quality. Likewise, Misra *et al.* (2009) found a statistically significant and positive relationship between customer collaboration and software development project success. Jun *et al.* (2011) also found a significant positive relationship between US and product performance. However, traditional plan-driven projects may not necessarily require US once specifications are clearly done compared to agile projects. Thus:

H12(T). No significant relationship between US and project success for traditional plan-based methodologies ($T\#$).

H12(A). A significant positive relationship between US and project success for agile methodologies ($A+$).

H12(TA). US is more important for agile compared with traditional plan-based methodologies ($A+ > T$).

User experience (UE). Previous scholars have shown that end user training, experience and education play a positive role in project success (Jun *et al.*, 2011; Livermore, 2008; Misra *et al.*, 2009; Charvat, 2003; Jiang and Klein, 2000). Based on these studies, it is apparent that, if software customers are highly educated, trained or experienced in IT, and are willing to participate in the development process and are supportive of the new software product, agile methodologies are more likely to be used rather than traditional approaches and vice versa. Thus:

H13(T). No significant relationship between UE and project success for traditional plan-based methodologies ($T\#$).

H13(A). A significant positive relationship between UE and project success for agile methodologies ($A+$).

H13(TA). UE is more important for agile compared with traditional plan-based methodologies ($A+ > T$).

2.5.4 Project factors and software development project success. Technological uncertainty (TU). Technical complexity (TC) and project uncertainty are frequently regarded as independent (e.g. Ratbe *et al.*, 2000; Shenhar and Dvir, 2007). However, authors such as Petit (2012) and Hass (2008) consider complexity and uncertainty to be aspects of the same variable. The use of new technologies increases uncertainty (Howell *et al.*, 2010). Consistent with Nidumolu's (1996) study, Jun *et al.* (2011) found that the use of unfamiliar technologies can lead to software problems that reduce the performance of the software product and delay the project. Unlike traditional plan-based, agile development methodologies were developed to respond to change and uncertainty in requirements and to reduce the cost of change throughout the project (Cockburn and Highsmith, 2001). New technologies may not go as planned and require a team to adapt as they learn. Gradual learning is more suitable for agile methodologies as they use short iterations and constant feedback. Thus:

H14(T). A significant negative relationship between TU and project success for traditional plan-based methodologies ($T-$).

H14(A). A significant negative relationship between TU and project success for agile methodologies ($A-$).

H14(TA). TU has a greater negative effect for traditional plan-based compared with agile methodologies ($\mathrm{T}- > A$).

TC. Jun *et al.* (2011) reported that TC adversely and negatively affected software project performance in terms of both process and product performance. In terms of methodologies, whereas Wysocki (2009, p. 312) indicates that as the level of TC increases so does the importance of process flexibility and the need to be creative and adaptive delivered by agile methodologies. Jiang and Klein (2000, p. 77) argue that complexity requires structure and discipline that comes from traditional methodologies. Likewise, Highsmith (2010) argues that dealing with complexity requires structure and discipline provided by traditional plan-based methodologies. Thus:

H15(T). A significant negative relationship between TC and project success for traditional plan-based methodologies ($T-$).

H15(A). A significant negative relationship between TC and project success for agile methodologies ($A-$).

H15(TA). TC has a greater negative effect for agile compared with traditional plan-based methodologies ($A- > T$).

Relative project size (RPS). Some empirical evidence reveals that project size negatively affects project performance since extra communication and coordination may be required (Sauer and Cuthbertson, 2003). Jun *et al.* (2011) found that the project size was negatively and significantly associated with project performance in terms of both process and product performance. Boehm and Turner (2003) argue that it is very difficult to apply agile methodologies to large projects, whereas it is more rigid and cumbersome to apply traditional plan-based methodologies to small projects. Thus:

H16(T). A significant negative relationship between RPS and project success for traditional plan-based methodologies ($T-$).

H16(A). A significant negative relationship between RPS and project success for agile methodologies ($A-$).

H16(TA). RPS has a greater negative effect for agile compared with traditional plan-based methodologies ($A- > T$).

Specification changes (SC). Boehm and Turner (2003) suggest that SC can negatively affect project performance. Ratbe *et al.* (2000) found that changes in specifications negatively affect project success. Equally, Nidumolu (1995) found that requirements instability had a negative effect on project performance. Too much user SC may have a negative effect on project success and in particular, variations in delivery time, scope and budget (Nidumolu, 1995). However, agile projects are likely to be successful if the requirements change rate is high since they have been specifically developed to address the problems of rapid change. Iterative delivery helps reduce uncertainty and leads the project through uncertainty while basing on customer feedback for continuous quality improvements. This is unlike traditional approaches that plan for every task in advance and many of them will not be executed if changes occur. In the case of a low SC, the traditional methodology and its big, up-front design works best. As there is no or little change expected during the project and the plan does not need to be modified. Future features are prepared in the design and all the pieces are designed to fit well together. Thus:

H17(T). A significant negative relationship between SC and project success for traditional plan-based methodologies ($T-$).

H17(A). A significant negative relationship between SC and project success for agile methodologies ($A-$).

H17(TA). SC has a greater negative effect for traditional plan-based compared with agile methodologies ($T- > A$).

Hypotheses related to project criticality (*H18*) were removed due to poor psychometric properties of the collected data invalidating hypothesis testing.

The research hypotheses are summarized and illustrated in Figure 1.

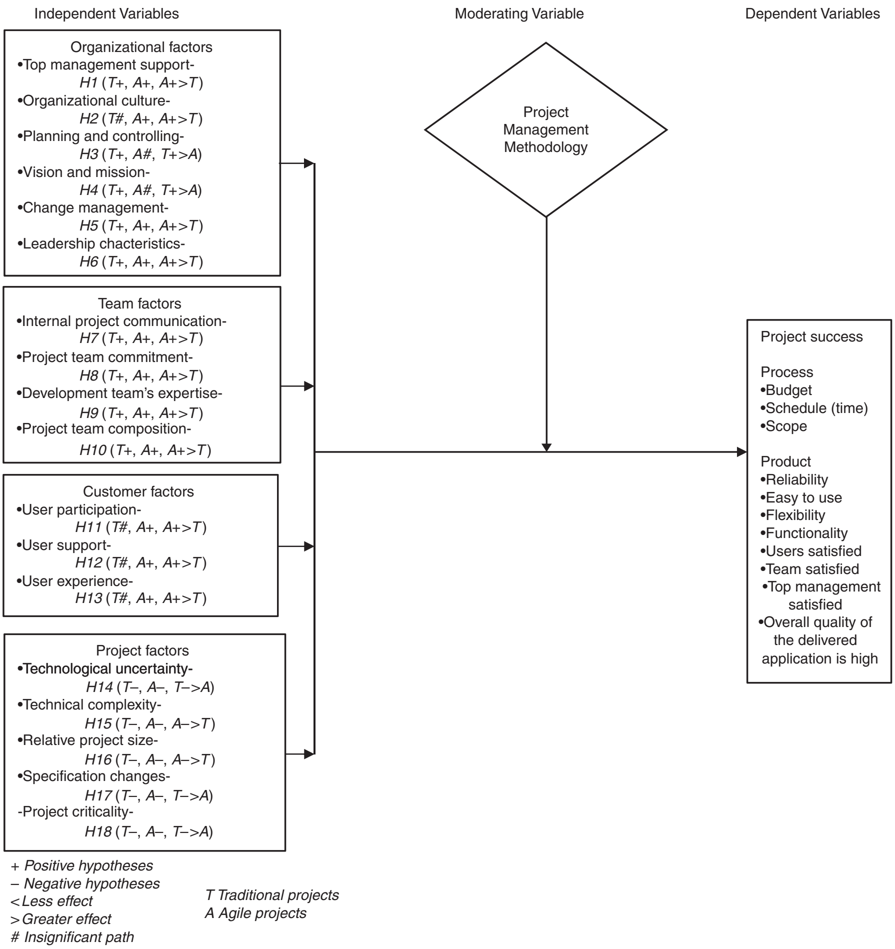


Figure 1.
A contingency fit model of CSFs for outsourced software development projects (hypotheses included)

3. Methodology

3.1 Data collection

The initial draft of the survey was pilot tested on Victoria Business School professors (VUW). Subsequently, the revised survey was pre-tested on 45 senior and experienced software development project managers from nine different industries in the capital city, Wellington. They were asked to complete and evaluate the online instrument. Their feedback contributed to the improvement of the layout of the questionnaire, clarity and its wording. Based on these pilot tests, the final instrument was developed.

An online global survey was e-mailed directly to senior software development project managers and practitioners. All these were professional members of PRINCE2, PMI (i.e. traditional plan-based project management) and agile communities with practical experience in software project management. These aforementioned communities of practice provide different guidelines for managing software projects and have a significant influence on the selection of the development methodology to be used. The mailing list had been derived from lists of these professional members. Essentially, membership to these software professional bodies is acquired based on qualifications, experience and personal choice.

The survey was expected to take about 15 minutes to complete. Following guidelines set forth by Dillman (1991), questions were to the point, addressed only a single issue at a time and avoided phrases that could elicit social or professional methodological affiliation biased responses. Each construct was measured by at least three questions that were relevant in terms of prior research or established theory. Survey questions were designed to capture the perceptions of project managers about software development practices for which they are expected to be the most knowledgeable and manage on an ongoing basis. When using single informants it is desirable to select the most experienced and knowledgeable person.

A well-designed cover letter was included that explained the purpose and intended use of survey data. To encourage participation, the respondents were informed that data would be collected and reported anonymously and a summary of study findings was offered. Each respondent was asked to fill in the survey for only the last outsourced software project he/she worked on regardless of whether the project was successful or not. This provided respondents with clear criteria for selecting the required projects and reduced the potential bias of reporting on only successful projects so that even the failed projects were collected in the database for the analysis. After two weeks reminders were sent. The Qualtrics survey remained online for eight weeks. A copy of the survey instrument is available from the online supplementary file to this paper.

3.2 Measures

Sources for research and tested instruments used to operationalize constructs were as follows: top level management support (Nah and Delgado, 2006), organizational culture (Strode *et al.*, 2009), project P&C (Jun *et al.*, 2011), LC (Misra *et al.*, 2009), CM (Nah and Delgado, 2006), V&M (Wan and Wang, 2010; Nah and Delgado, 2006), IPC (Jun *et al.*, 2011), PTC (Ahimbisibwe and Nangoli, 2012), development expertise and skill (Jiang and Klein, 2000; Jun *et al.*, 2011), team's composition (Nah and Delgado, 2006), UP (Jun *et al.*, 2011), US (Jiang and Klein, 2000), UE (Jiang and Klein, 2000; Jun *et al.*, 2011), TU (Nidumolu, 1995), TC (Jun *et al.*, 2011), RPS (Jiang and Klein, 2000; Jun *et al.*, 2011), SC (Nidumolu, 1995), project criticality (Sheffield and Lemétayer (2013), project success (Jun *et al.*, 2011; Sheffield and Lemétayer, 2013).

3.3 Survey responses

Overall, 4,500 e-mail invitations were sent out to all the intended research participants during data collection of this study. Of these e-mails invitations 1,626 automatic responses were received back confirming that either the owners had changed their original contacts or that the e-mails were no longer valid. In total, 1,880 participants started to complete the survey questionnaire. Out of these, only 1,376 respondents completed and submitted the online survey questionnaire. Some of the responses were identified as skimmers and deleted. In total only 984 usable vendor responses were received which had clearly specified the PMM used. This represents about 34.2 per cent of the initial valid e-mail invitations sent out which compares well with most other IS surveys (e.g. Rai and Hindi, 2000; Aladwani, 2002; Wallace *et al.*, 2004; Jun *et al.*, 2011; Sheffield and Lemétayer, 2013).

Table II shows the detailed demographics of the various sample characteristics for the data collected from vendors for this study (i.e. the traditional plan-based and agile methodology projects). The descriptive statistics indicate the data profiling of the respondents as well as the projects. Typically, about 95 per cent of these respondents occupied senior positions and carried out senior roles, hence are very conversant with the operations and management of outsourced software development projects. These projects were predominantly undertaken in larger client organizations with 80 per cent employing

Characteristics	Scale	Traditional plan-based methodology, $n = 513$	Agile methodology projects, $n = 471$
Experience of the vendors	Less than 2 years	35 (6.8%)	8 (0.16%)
	2 to less than 5 years	66 (12.9%)	13 (2.8%)
	5 to less than 10 years	135 (26.3%)	79 (16.8%)
	More than 10 years	277 (54.0%)	371 (78.8%)
Positions of the vendors on the project	Project manager	449 (87.5%)	381 (80.9%)
	Team leader	50 (9.7%)	50 (10.6%)
	Others	14 (2.8%)	40 (8.5%)
Size of the client firm	1-100	77 (15%)	118 (25.1%)
	101-500	66 (12.9%)	74 (15.7%)
	501-1,000	83 (16.2%)	50 (10.6%)
	1,001-5,000	111 (21.6%)	79(16.8%)
	More than 5,000	176 (34.3%)	150 (31.8%)
Project budget (USD)	Less than $100,000	105 (20.5%)	104 (22.1%)
	$100,000 to less than $1M	188 (36.6%)	231 (49%)
	$1M to less than $10M	161 (31.4%)	91 (19.3%)
	More than $10M	59 (11.5%)	45 (9.6)
Duration in months	Less than 6 months	90 (20.5%)	97 (20.6%)
	6 months to less than 12 months	197 (36.6%)	194 (41.2%)
	12 months to less than 24 months	150 (31.4%)	88 (18.7%)
	More than 24 months	76 (11.5%)	92 (19.6)
Number of project members on each team	2-5	42 (8.2%)	70 (14.9%)
	6-100	407 (79.3%)	343 (72.8%)
	101-500	47 (9.2%)	43 (9.1%)
	Above 500	17 (3.4%)	15 (3.1%)
Industry of the clients	Finance/insurance	138 (26.9%)	105 (22.3%)
	Manufacturing	77 (15.0%)	32 (6.8%)
	Marketing/retail	18 (3.5%)	49 (10.4%)
	Health	49 (9.6%)	58 (12.3%)
	Software	55 (10.7%)	127 (27.0%)
	Transportation	21 (4.1%)	21 (4.5%)
	Others	155 (30.3%)	79 (16.8%)

Table II. Demographics of the respondents-group comparisons

more than 500 employees, as well as approximately 80 per cent being larger projects indicated by larger budgets mostly ranging between one and ten million dollars and lasting for more than six months.

3.4 Common methods bias and non-response bias

While a popular approach to overcoming self-reporting problems is to match data from two respondents, this approach was not possible in this study. Collecting data from separate respondents for the same variables would have made maintaining anonymity difficult and detrimentally affected the sample size required for fitting SEM models. Collecting data from two respondents can also introduce errors when linking up data together for predictor and criterion variables requires significantly more respondent time, effort and cost (Podsakoff *et al.*, 2003). Consistent with Podsakoff *et al.* (2003), item scales were improved by avoiding vague concepts and providing examples where necessary; keeping questions simple, specific and concise; avoiding double barrelled questions; reducing questions relating to more than one possibility into more simple questions; and avoiding complicated syntax. Common method bias was further assessed using Harman's one factor test (Podsakoff *et al.*, 2003). In a third test, a confirmatory factor analysis was conducted and included all the

97 measurement items, the 19 latent constructs, and a 20th latent construct linked to all 97 measurement items. Both statistical tests indicated that common method variance was not a problem in this study. For non-response bias, all mean differences for each of the constructs did not reveal any significant difference between the early and late questionnaires (two-tailed *t*-tests, $p < 0.05$). This test suggests the absence of non-response bias in this study.

3.5 SEM data analysis

Data were analysed using SEM. SEM is a multivariate technique in which parameters are estimated by minimizing the discrepancy between the model-implied covariance matrix and the observed covariance matrix (Joreskog and Sorbom, 1989). SEM was selected in this study because it tests an overall model rather than individual coefficients, is a confirmatory approach that provides explicit test statistics for establishing convergent and discriminant validity important to MIS and management research, allows for error terms, reduces measurement error through the use of multiple indicators and SEM also allows multi-group comparisons (Straub, 1989; Hair *et al.*, 2009). The robustness of SEM using maximum likelihood estimation (MLE) has been demonstrated in prior management/project management/MIS research (e.g. Kearns and Sabherwal, 2007; Jung *et al.*, 2008).

The approach chosen was separate analysis of the measurement and structural models in a two-step process recommended by Anderson and Gerbing (1988). This is consistent with prior project management research that has used SEM and allowed refinement of measures before testing of the structural model.

4. SEM results

All survey responses were anchored between strongly disagree (1) and strongly agree (7), so a response of 4.0 would indicate neutrality and responses of or greater than 4.5 would indicate agreement. Mean responses for the survey items were in the range of 3.00-6.15 (please refer to the supplementary file Table AI – mean differences for group comparisons). Mean differences for all of the constructs revealed no significant differences between the agile methodology projects data set and traditional plan-based methodology data set (two-tailed *t*-tests, $p < 0.05$). However, some few measurement items of, e.g., UE, P&C and TC were slightly different across the two groups ($p < 0.05$). This was somehow expected since the literature review suggests that traditional plan-based and agile methodologies are argued to differ across these dimensions.

4.1 Measurement model

In the first phase, a measurement model was used to measure the fit between the theorized model and observed variables, and to establish reliability and validity. Cronbach's α coefficients and composite scale reliabilities for all 19 constructs exceeded the recommended minimum of 0.70 (Fornell and Larcker, 1981) for both methodology data sets. The average variance extracted (AVE) was greater than 0.5 (Fornell and Larcker, 1981; Anderson and Gerbing, 1988). Table III shows the results of the reliability and validity tests.

For convergent validity, the factor loadings for each measurement item on each latent constructs were all high ($\lambda \geqslant 0.6$) with high levels of significance ($p < 0.001$) for all the indicator variables (Straub, 1989; Bagozzi and Yi, 1988; Anderson and Gerbing, 1988).

Table AII (in the online supplementary file) shows standardized regression weights from the measurement models for group comparisons.

The correlations between the constructs were strong and highly significant (Straub, 1989; Bagozzi and Yi, 1988) and for all 171 construct pairs, the AVE of each construct

Construct	Traditional plan-based methodology projects, $n = 513$ Composite scale reliability (CSR)	Cronbach's α coefficient	Average variance extracted (AVE)	Agile methodology projects, $n = 471$ Composite scale reliability (CSR)	Cronbach's α coefficient	Average variance extracted (AVE)
Top management support	0.87	0.89	0.58	0.88	0.89	0.62
Organizational culture	0.94	0.95	0.77	0.96	0.97	0.80
Planning and controlling	0.85	0.83	0.60	0.86	0.87	0.59
Leadership characteristics	0.88	0.89	0.64	0.90	0.93	0.71
Change management	0.84	0.85	0.58	0.85	0.85	0.56
Vision and mission	0.89	0.92	0.67	0.88	0.90	0.67
Int. project communication	0.84	0.85	0.55	0.87	0.88	0.63
Project team commitment	0.88	0.89	0.66	0.88	0.89	0.65
Dev. team's expertise	0.91	0.92	0.72	0.89	0.90	0.64
Team composition	0.86	0.85	0.55	0.86	0.88	0.62
User participation	0.84	0.85	0.55	0.90	0.90	0.70
User support	0.86	0.87	0.61	0.93	0.95	0.63
User experience	0.90	0.90	0.70	0.93	0.94	0.72
Technical complexity	0.87	0.89	0.70	0.88	0.87	0.75
Technological uncertainty	0.92	0.94	0.75	0.93	0.94	0.76
Relative project size	0.94	0.96	0.75	0.93	0.94	0.73
Specification changes	0.85	0.86	0.62	0.87	0.88	0.53
Process project success	0.87	0.88	0.67	0.86	0.87	0.63
Product project success	0.95	0.96	0.69	0.94	0.95	0.68

Notes: AMOS does not compute CSR and AVE. The following formulae were used to calculate AVE and CSR: composite scale reliability = (SUM (A))^2/(SUM (A))^2+SUM (B), where (SUM (A))^2 = sum of standardized factor loadings squared and SUM (B) = sum of indicator measurement errors (sum of the variance due to random measurement error for each loading = 1 − the square of each loading) (Fornell and Larcker, 1981). AVE = (SUM (A^2))/(SUM (A^2))+SUM (B), where (SUM (A^2)) = sum of squared standardized loadings and SUM (B) = sum of indicator measurement error) (Fornell and Larcker, 1981). Cronbach's α was computed using SPSS V.19 as AMOS does not compute it

Table III. Reliability and validity

exceeded the square of the construct correlations and discriminant validity was established. Tables AIII and AIV (in the online supplementary file) show correlations for both traditional plan-based and agile data sets.

4.2 Structural model

In the second phase, results of the measurement model were used to create a structural model in order to measure the strength of the theorized relationships. Table IV summarizes measures of goodness-of-fit for both the measurement and structural models.

Both research models are successful in accounting for a substantial portion of the variability in the dependent variables. The traditional plan-based methodology projects model explained more than 46 and 68 per cent of the total variance in both process and product success, respectively, whereas more than 53 and 69 per cent, respectively, was accounted for by the agile methodology projects model.

Statistical research findings on minimum sample size indicate very little statistical research in the fields of MIS and behavioural sciences to shed light on the issue of establishing a minimum desirable level of sample size (MacCallum *et al.*, 1999). Generally, numerous authors have recommended dissimilar minimum sample size in factor analysis based on two categories. One category suggests that the absolute number of cases (n) is important, while another argues that the subject-to-variable ratio (p) is important.

Table IV.
Measures of goodness-of-fit – group comparisons

Goodness-of-fit index	Traditional plan-based methodology projects data set, $n = 513$ Measurement model	Structural model	Agile methodology projects data set, $n = 471$ Measurement model	Structural model	Heuristics/ cut-off criteria
χ^2	6,683.41	6,675.86	5,984.33	6,626.92	$p > 0.5$
χ^2/df	2.12	2.23	2.02	2.06	< 3.00
TLI	0.91	0.89	0.92	0.88	> 0.95
CFI	0.89	0.88	0.89	0.89	> 0.95
RMSEA	0.04	0.04	0.04	0.04	< 0.06
NFI	0.88	0.90	0.89	0.91	$\geqslant 0.95$
PRATIO	0.90	0.91	0.90	0.93	> 0.90
GFI	0.93	0.89	0.94	0.90	> 0.90
AGFI	0.90	0.89	0.91	0.89	> 0.90

The underlying dilemma is that the general rule of thumb of the minimum sample size is only indicative of how big the sample should tend to be, but not absolute. The decision as to whether a set of data fits the model is a matter for the informed judgment of the researcher, based on theory and the nature of the measurements used to test the sample, rather than a matter of some contrary cut-off rule applied to a column of numbers on a computer printout.

According to Preacher and MacCallum (2002, p. 160), as long as factor loadings are high and the SEM model fit indices (for sample size) are within the acceptable range, researchers and reviewers should not be overly concerned about reasonable sample sizes. Hu and Bentler (1999, p. 214) acknowledge that it is difficult to designate a specific cut-off criteria because it does not work well equally with various conditions. Since the sample sizes were all in a close range of 500 for each data set (please refer to Rule of 500 by Comrey and Lee, 1992), and moreover, the SEM model fit indices that depend on sample sizes, e.g. AGFI, PGFI, PRATIO and PNFI were very satisfactory, there is a very high likelihood that the sample sizes were adequate. Thus, although, the SEM model appears intricate, it was appropriately analysed with no statistical concerns hence no method concerns are likely to biasing the results.

4.2.1 Support for hypotheses. Using 513 observations for traditional plan-based methodology projects and 471 observations for agile projects provided by a global survey of experienced and senior software development practitioners, survey data supported most initially hypothesized relationships. However, some paths coefficients did not demonstrate the expected signs. Similarly, some structural path coefficients were insignificant not as hypothesized. Table V summarizes the hypothesis testing results.

5. Discussion

This paper empirically examines the differences between CSFs in the current context of PMM from vendors' perspective. The results largely support the proposed hypotheses. This evidence of associations supports and extends previous research. A detailed discussion of the results for each hypothesis is provided in the online supplementary file, and summarized: discussion is provided here.

As predicted, the study reveals that for instance, flexible OC, UP, US and UE are not significant predictors of project success for traditional plan-based projects. Similarly, as initially hypothesized P&C, CM and LC, etc., significantly and positively influence project success, whereas, TU, TC, size and SC, etc., negatively and significantly influences project success for traditional plan-based projects.

On the other hand, for agile projects, the initial hypotheses for non-significant associations between P&C, V&M with project success were strongly supported.

Table V. Hypothesis testing results – group comparisons

Hypothesis and the structural path				Traditional plan-based methodology (T), $n = 513$			Agile data set methodology (A), $n = 471$			Comparison hypothesis, TA (statistical difference test)	
				Hypothesized relationship	Coefficient/ significance	Support	Hypothesized relationship	Coefficient/ significance	Support	Hypothesized relationship	Support
H1(i)	Top mgt. support	→	PS (process)	+	−0.04	No	+	−0.16**	No[a]	$A+ > T$	No[a]
H1(ii)	Top mgt. support	→	PS (product)	+	−0.03	No	+	−0.05	No	$A+ > T$	No
H2(i)	Organizational culture	→	PS(process)	#	0.02	Yes	+	0.06*	Yes	$A+ > T$	No
H2(ii)	Organizational culture	→	PS (product)	#	0.01	Yes	+	0.07*	Yes	$A+ > T$	No
H3(i)	Planning and controlling	→	PS (process)	+	0.28***	Yes	#	0.02	Yes	$T+ > A$	Yes
H3(ii)	Planning and controlling	→	PS (product)	+	0.05	No	#	0.04	Yes	$T+ >$ A	No
H4(i)	Vision and mission	→	PS (process)	+	0.07*	Yes	#	−0.02	Yes	$T+ > A$	No
H4(ii)	Vision and mission	→	PS (product)	+	0.02	No	#	0.01	Yes	$T+ > A$	No
H5(i)	Change management	→	PS (process)	+	0.06*	Yes	+	0.08*	Yes	$A+ > T$	No
H5(ii)	Change management	→	PS (product)	+	0.07*	Yes	+	0.21***	Yes	$A+ > T$	Yes
H6(i)	Leadership characteristics	→	PS (process)	+	0.06*	Yes	+	0.22***	Yes	$A+ > T$	Yes
H6(ii)	Leadership characteristics	→	PS (product)	+	0.05	No	+	−0.04	No	$A+ > T$	No
H7(i)	Int. proj. communication	→	PS (process)	+	0.08*	Yes	+	0.09*	Yes	$A+ > T$	No
H7(ii)	Int. proj. communication	→	PS (product)	+	0.06*	Yes	+	0.07*	Yes	$A+ > T$	No
H8(i)	Proj. team commitment	→	PS (process)	+	0.08*	Yes	+	0.09*	Yes	$A+ > T$	No
H8(ii)	Proj. team commitment	→	PS (product)	+	0.07*	Yes	+	0.08*	Yes	$A+ > T$	No
H9(i)	Devt. team expertise	→	PS (process)	+	0.07*	Yes	+	0.09*	Yes	$A+ > T$	No
H9(ii)	Devt. team expertise	→	PS (product	+	0.02	No	+	0.06*	Yes	$A+ > T$	No
H10(i)	Proj. team composition	→	PS (process)	+	0.07*	Yes	+	0.25***	Yes	$A+ > T$	Yes
H10(ii)	Proj. team composition	→	PS (product)	+	0.08*	Yes	+	0.26***	Yes	$A+ > T$	Yes
H11(i)	User participation	→	PS (process)	#	0.02	Yes	+	0.28***	Yes	$A+ > T$	Yes
H11(ii)	User participation	→	PS (product)	#	0.05	Yes	+	−0.14***	No[a]	$A+ > T$	No[a]
H12(i)	User support	→	PS (process)	#	0.03	Yes	+	0.06*	Yes	$A+ > T$	No
H12(ii)	User support	→	PS (product)	#	0.02	Yes	+	0.18***	Yes	$A+ > T$	Yes
H13(i)	User experience	→	PS (process)	#	0.03	Yes	+	0.07*	Yes	$A+ > T$	No
H13(ii)	User experience	→	PS (product)	#	0.02	Yes	+	0.06*	Yes	$A+ > T$	No
H14(i)	Technological uncertainty	→	PS (process)	−	−0.24***	Yes	−	−0.09*	Yes	$T- > A$	Yes
H14(ii)	Technological uncertainty	→	PS (product)	−	−0.26***	Yes	−	−0.06*	Yes	$T- > A$	Yes

(continued)

Table V.

Hypothesis and the structural path				Traditional plan-based methodology (T), $n = 513$			Agile data set methodology (A), $n = 471$			Comparison hypothesis, TA (statistical difference test)	
				Hypothesized relationship	Coefficient/ significance	Support	Hypothesized relationship	Coefficient/ significance	Support	Hypothesized relationship	Support
H15(i)	Technical complexity	→	PS (process)	–	–0.02	No	–	–0.17**	Yes	$A- > T$	No[a]
H15(ii)	Technical complexity	→	PS (product)	–	–0.14**	Yes	–	0.05	No	$A- > T$	No[a]
H16(i)	Relative project size	→	PS (process)	–	–0.07*	Yes	–	–0.25***	Yes	$A- > T$	Yes
H16(ii)	Relative project size	→	PS (product)	–	–0.05	No	–	0.04	No	$A- > T$	No
H17(i)	Specification changes	→	PS (process)	–	–0.14*	Yes	–	–0.08*	Yes	$T- > A$	No
H17(ii)	Specification changes	→	PS (product)	–	0.22***	No[a]	–	0.06*	No[a]	$T- > A$	No[a]
H18(i)	Project criticality	→	PS (process)	–	x	x	–	x	x	$A- > T$	x
H18(ii)	Project criticality	→	PS (product)	–	x	x	–	x	x	$A- > T$	x

Notes: PS, project success (process and product). +, Positive hypothesis; –, negative hypothesis; $<$, less effect; $>$, greater effect; #, insignificant path; x, construct dropped and excluded from SEM analysis due to poor psychometric measurement properties. [a]Significant but not as hypothesized. *,**,***Significant at 0.05, $p < 0.01$, $p < 0.001$, respectively

Similarly, the hypotheses for the significant positive relationships between PTC, UP, and US, etc., with project success were strongly supported. There was negative and significant influence between TU, TC, size and SC, etc., and project success.

Some paths did not demonstrate the expected signs. For instance, with the agile data set, the structural path between top level management support and process success was expected to be positive but was negative and significant. One possible explanation is that although TMS is generally viewed as a positive factor, some researchers argue that what begins as support can easily turn into interference, which is considered counterproductive and detrimental especially in highly innovative environments (Young and Jordan, 2008). This could imply that in circumstances where agility is coupled with uncertainty and changes, new software development project success depends more on other aspects such as the presence of deep technological expertise. It is also highly likely that top management directives and controls often accompany TMS (Swink, 2000). Such directives and controls are potentially counterproductive to agile practices.

Similarly, some structural paths were insignificant and not as expected. For example, in the traditional plan-driven data set, the path between TMS and process and product success was not supported, the path between V&M and product success was found to not differ significantly from zero among others. This is possibly because previous studies focussed on in-house development projects where developers and users are members of the same organization, rather than the vendor's perspective employed in this study. Nonetheless, TMS was found to indirectly and positively influence process success through P&C. This means that TMS only contributes towards setting project plans and controls which if adhered to can improve project success.

The structural path between project P&C and product project success for traditional plan-driven projects was insignificant. This is consistent with Jun *et al.* (2011) who did not find a statistically significant relationship between P&C and product success. This finding indicates that better quality of project plans and controls does not necessarily lead to better quality of outsourced software development project products. This might be because better long-range project plans may be followed by poor planning at the project or team level and mistakes in terms of the assignment of personnel, requirement specification, less customer participation or feedback, etc. Thus, a better software project master plan should not be considered as an end-goal in itself; instead, it should be complemented with top managers' participation in IT resource allocation, customer interaction, proper assignment of personnel and requirement specification. These findings are consistent with recent Kearns and Sabherwal's (2007) empirical survey of 274 CIO's indicating that most outsourced software development projects practice intensive planning, controlling and alignment but continue to experience failed software quality products and IT investments.

The findings suggest that if the top level management of an organization supports an adaptive behaviour, flexible leadership styles and agile culture, they are not likely to support a rigid traditional approach that requires up-front detailed planning and formal specification, but rather agile methodologies should be adopted. These findings mirror Sheffield and Lemétayer's (2013) study. If there is no commitment of the user to participate in requirements definition and provide full time support for the project team, traditional plan-based approaches should be used and vice versa. This is because unlike agile approaches, traditional approaches do not necessarily require full time customer involvement right from the initial specification phase to the end. This finding is consistent with Chow and Cao (2008), Misra *et al.* (2009) and Sheffield and Lemétayer (2013). If the software customers are highly educated, trained or experienced in IT and are willing to participate in the development process and are supportive of the new software product, agile methodologies should be used rather than traditional approaches and vice versa.

Simply because, for agile projects, the customer must be able to articulate his needs clearly, otherwise this threatens the whole project.

The findings indicate that traditional methodologies should be used when the future can be easily predicted, while agile (adaptive and innovative) methods should be adopted under conditions of uncertainty. The traditional approach attempts to minimize change, while the agile approach embraces it. This is consistent with Vinekar *et al.* (2006). If there is no or little change expected during the project, the plan does not need to be modified and traditional approaches that plan for every task in advance should be used. Future features should be prepared in the design and all the pieces designed to fit well together. However, since responding to change is one of the key principles of the agile manifesto and change usually occurs faster than plan can be updated, agile methodologies should be used when P&C are not possible since they handle such situation very well. The results essentially appear to suggest that traditional plan-based methodologies should be used in organizations that are characterized with mechanistic and bureaucratic structures that emphasize more P&C procedures, while agile approaches should be used in organizations with organic and flexible structures that possibly support more informal communication and empowerment of the project teams.

6. Conclusions, limitations and future research

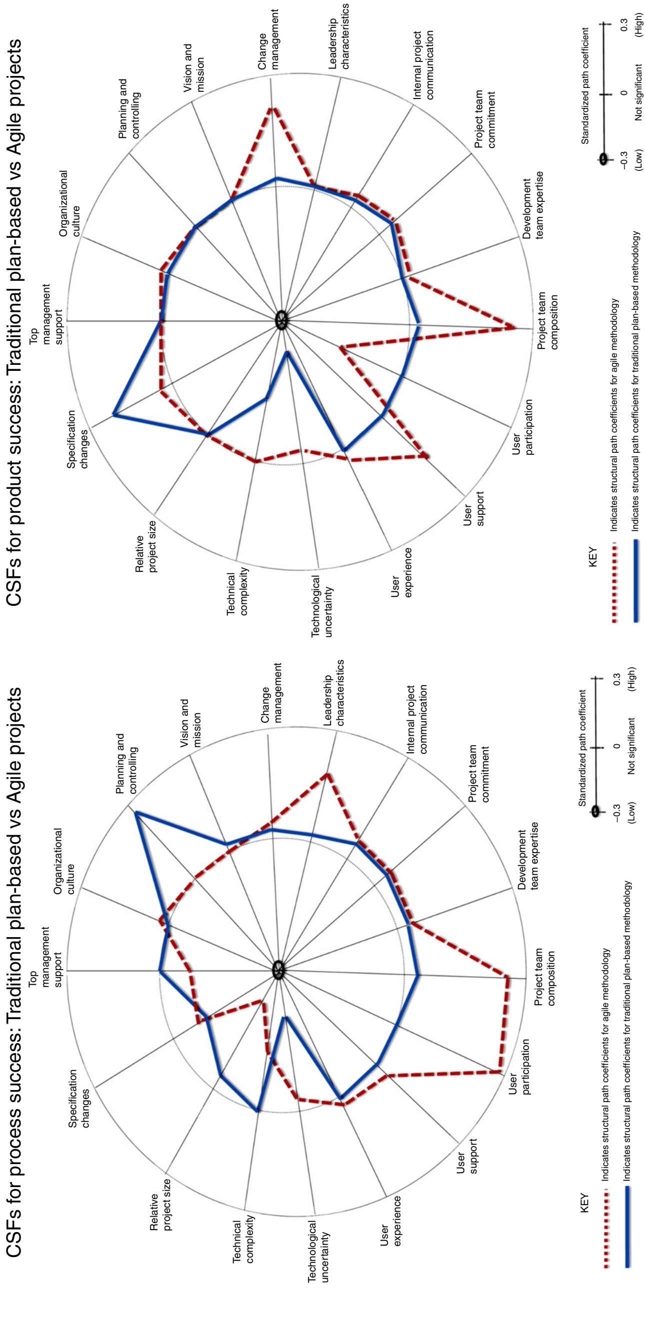
This section summarizes the overall contributions of this research to project management theory and practice, the major limitations and some future directions associated with this research.

6.1 Contributions to project contingency theory

Previous studies of software development projects (e.g. Misra *et al.*, 2009; Highsmith, 2010; Wan and Wang, 2010; Sheffield and Lemétayer, 2013) have addressed only one methodology per study, typically focussing on the benefits of agile methodologies. They have also tended to view CSFs as an inverted form of reasons of failures amidst the wide range of project success criteria, whereas some CSFs may have little or no effect depending on the methodology chosen. To address this, Ahimbisibwe *et al.* (2015) and others have recommended empirical studies comparing traditional plan-based and agile methodologies. The study presented here is the first to explicitly contrast traditional and agile methodologies from a vendor perspective, finding that the relationships of CSFs can differ significantly across the two methodologies as well as with respect to the two main project success measures (see Figure 2).

The detailed literature review entailed that two competing hypotheses and one hypothesis for empirical comparisons were formulated for each CSF, explicating these different and sometimes conflicting results of the impact of each CSF for various methodologies. These contingent hypotheses (a total of over 60), empirically assessed using a global survey which generated a large sample of 1,880 (984 usable) responses from senior project managers around the globe, permitted the aggregate sample to be split into two subgroups (traditional plan-based and agile projects) which enabled comparative analyses across the two project types. The robustness of covariance-based SEM using MLE enabled the simultaneous analysis of all these hypotheses, with the evidence of associations supporting and extending contingency theory with respect to software project management.

This study builds from the base of Boehm and Turner's (2003) five CSFs for software development project methodology selection by examining a total of 17 CSFs developed from the project management literature where key features of agile and traditional plan-based approaches are contrasted (Boehm and Turner, 2003, 2004; Vinekar *et al.*, 2006; Sheffield and Lemétayer, 2013; Ahimbisibwe *et al.*, 2015). These, collectively, determine the profile of a given software development project and their relationships to key success outcomes,



Note: Project criticality not empirically tested in the model due to poor psychometric properties

Figure 2.
Comparison of CSFs for project success for traditional plan-based and agile methodologies (standardized path coefficients)

which can help with the selection of an appropriate methodology. A conceptual model was developed to augment how an appropriate choice of a project methodology associated with these 17 CSFs is best evaluated by first ascertaining how well the software development project fits with either the traditional plan-based or agile approach.

In forming a significant initial step via its large-sample quantitative data collection, the study here is able to improve clarity in current project management debates. In using a narrower set of CSFs, constructs and outcomes, the previous project management studies suggested that in some areas there was limited coherency on the best fit between CSFs and PMMs, as well as on how to assess the impact of this fit on project success. In contrast, by borrowing a theorization of fit from strategic management studies and MIS (e.g. Thompson, 1967; Miles and Snow, 1978; Venkatraman, 1989; Sauser *et al.*, 2009), this research developed a more extensive and comprehensive measurement instrument for software development projects success in the context of alternative methodologies that hopefully will be useful for researchers in future studies to examine the concept of fit further.

The current research also contributes "best practice" frameworks of methodological success, with each accounting for a substantial portion of the variability in the dependent variables. The traditional plan-based methodology projects model explained more than 46 and 68 per cent of the total variance in both process and product success, respectively, whereas more than 53 and 69 per cent, respectively, was accounted for by the agile methodology projects model. The SEM models also provided strong evidence of convergent and discriminant validity, and very good internal consistency. The high reliability achieved for the few novel scales for measuring constructs may further assist researchers and practitioners in reflecting on PMM selection, and also to initiate changes that can help achieve higher rates of project success.

6.2 Contributions to project management practice

The most valuable contribution for practitioners from this research is the more comprehensive evidence that different project types in various project contexts require different management approaches and methodologies. This suggests that software practitioners should carefully consider the project types, project contexts (CSFs) and project methodologies when managing software development projects. The findings demonstrate that project managers and, more generally, top management and organizations should adopt a more project-specific approach to project management and software development based on CSFs. This is likely to improve the current project success rates.

Second, many software development practitioners appear to be affiliated, committed and loyal to different methodologically aligned communities. However, to obtain better desired end results, software development practitioners have to ensure that their allegiance to a particular community of PMM practice does not influence them to select an inappropriate SDM.

For example, for agile methodology projects, top management may want to draw a note of caution regarding their close involvement in high technologically innovative development efforts. Since TMS is usually in the form of providing sufficient resources to the team in ensuring project success and communicating with project team in the case of authority and responsibility, TMS is therefore, particularly important in the times of crisis or when unexpected situations arise.

6.3 Limitations of the research

Despite the large sample obtained, self-administration of the survey made it difficult to ensure that the project reported on was the last undertaken for an external client. Being based on the literature, equivalent factors internal to the vendor organization highlighted as CSFs associated with the client's project were not included and so their impact on project success could not be examined. The survey also required the vendors to assess the culture, degree of commitment, etc. in the external client, with which they likely had less familiarity. Nonetheless, more than 80 per cent of all the projects in this study were predominantly large and occurred over multiple months, so it is likely that vendors were able to gauge these CSFs in some depth.

The study of CSFs for project success was limited to factors that are currently documented in the software project management literature. Equally, for modelling purposes and parsimony reasons not all CSFs could be incorporated in one model and we drew on previous research to guide which CSFs were combined or represented through their constituent factors (e.g. Ahimbisibwe *et al.*, 2015). Without this, the model fitting process would have been difficult since several typical fit indices (e.g. AGFI, PGFI, PRATIO, PNFI) punish complexity in preference of parsimony.

The construct of project criticality did not fit the single congeneric model (preliminary assessment of the measurement model) as indicated by its single model fit indices which were all below the acceptable criteria despite several modifications. Therefore, project criticality had to be completely dropped from further data analysis due to poor psychometric properties and its impact on project success could not be assessed. However, project criticality is still a potentially important contingency but one for which enhanced measurement approaches and design are needed.

6.4 Future research directions

The study raises a number of possibilities for future studies. First, the extent to which the modelling of the CSFs fit well and were correlated in the SEM suggests that rather than representing discrete influential factors, the CSFs may be interconnected. This raises the possibility that lack of alignment with one CSF may lead to similar misalignment with others, potentially compounding any mismatch between CSFs and the chosen PMM. Thus, where vendor organizations or project managers favour particular methodologies, the multi-faceted lack of alignment likely contributes to the high reported rates of project failure. Second, while the vendor perspective is relatively under-researched in project management, only Jun *et al.* (2011) assess how it aligns with the client perspective and then only for a relatively small sample. Thus, while our study has examined the extent to which CSFs differ in their relationship to project success for the two main PMMs, if client and vendor perspectives on project success do not match each other closely, then the identified relationships for CSFs may differ for clients. Establishing the degree of concurrence would be a useful next step. It would also be interesting to understand the extent to which CSFs and PMMs match for in-house projects within vendor organizations. This would give insights into PMM choice since the vendor should be able to judge CSFs for in-house projects more quickly and accurately and thus is in a position to more readily match the methodology to the CSFs to drive project success. Finally, in aggregating insights from the wealth of software project management research, this research may help to enhance our understanding of the contingent nature of project management success in other areas as well.

References

Agile Alliance (2001), "Manifesto for agile software development", available at: www.agilealliance.org/agile101/the-agile-manifesto/ (accessed March 2017).

Ahimbisibwe, A. and Nangoli, S. (2012), "Project communication, individual commitment, social networks and perceived project performance", *Journal of African Business*, Vol. 13 No. 2, pp. 101-114.

Ahimbisibwe, A., Cavana, R.Y. and Daellenbach, U. (2015), "A contingency fit model of critical success factors for software development projects: a comparison of agile and traditional plan based methodologies", *Journal of Enterprise Information Management*, Vol. 28 No. 1, pp. 7-33.

Aladwani, A.M. (2002), "An integrated performance model of information systems projects", *Journal of Management Information Systems*, Vol. 19 No. 1, pp. 185-210.

Anderson, J.C. and Gerbing, D.W. (1988), "Structural equation modeling in practice: a review and recommended two-step approach", *Psychological Bulletin*, Vol. 103 No. 3, pp. 411-423.

Bagozzi, R.P. and Yi, Y. (1988), "On the evaluation of structural equation models", *Journal of the Academy of Marketing Science*, Vol. 16 No. 1, pp. 74-94.

Barki, H., Rivard, S. and Talbot, J. (2001), "An integrative contingency model of software project risk management", *Journal of Management Information Systems*, Vol. 17 No. 4, pp. 37-69.

Boehm, B. and Turner, R. (2003), "Using risk to balance agile and plan-driven methods", *IEEE Computer Society*, Vol. 36 No. 6, pp. 57-66.

Boehm, B. and Turner, R. (2004), *Balancing Agility and Discipline: A Guide for the Perplexed*, Addison-Wesley, Boston, MA.

Charvat, J. (2003), *Project Management Methodologies: Selecting, Implementing and Supporting Methodologies and Processes for Projects*, John Wiley & Sons Inc., New York, NY.

Chow, T. and Cao, D. (2008), "A survey of critical success factors in agile software projects", *Journal of Systems and Software*, Vol. 81 No. 6, pp. 961-971.

Cockburn, A. (2007), *Agile Software Development: The Cooperative Game*, Addison-Wesley, Upper Saddle River, NJ.

Cockburn, A. and Highsmith, J. (2001), "Agile software development: the people factor", *Computer*, Vol. 34 No. 11, pp. 131-133.

Comrey, A.L. and Lee, H.B. (1992), *A First Course in Factor Analysis*, Erlbaum, Hillsdale, NJ.

Dalcher, D. and Brodie, L. (2007), *Successful IT Projects*, Thomson Learning, London.

Dillman, D.A. (1991), "The design and administration of mail surveys", *Annual Review of Sociology*, Vol. 17 No. 1, pp. 225-248.

Fornell, C. and Larcker, D.F. (1981), "Evaluating structural equation models with unobservable variables and measurement error", *Journal of Marketing Research*, Vol. 18 No. 1, pp. 39-50.

Hair, J.F., Black, W.C., Babin, B.J. and Anderson, R.E. (2009), *Multivariate Data Analysis*, 7th ed., Prentice Hall, Englewood Cliffs, NJ.

Hass, K.B. (2008), "Introducing the new project complexity model", *Management Concepts*, pp. 22-31, available at: www.projecttimes.com/articles/introducing-the-new-project-complexity-model-part-i.html

Highsmith, J. (2010), *Agile Project Management: Creating Innovative Products*, 2nd ed., Addison-Wesley, Upper Saddle River, NJ.

Howell, D., Windahl, C. and Seidel, R. (2010), "A project contingency framework based on uncertainty and its consequences", *International Journal of Project Management*, Vol. 28 No. 3, pp. 256-264.

Hu, L. and Bentler, P.M. (1999), "Cut off criteria for fit indices in covariance structure analysis: conventional criteria versus new alternatives", *Structural Equation Modelling*, Vol. 6 No. 1, pp. 1-55.

Iivari, J. and Huisman, M. (2007), "The relationship between organizational culture and the deployment of systems development methodologies", *MIS Quarterly*, Vol. 31 No. 1, pp. 35-58.

Imreh, R. and Raisinghani, M. (2011), "Impact of agile software development on quality within information technology organisations", *Journal of Emerging Trends in Computing and Information Science*, Vol. 10 No. 10, pp. 460-475.

Jiang, J. and Klein, G. (2000), "Software development risks to project effectiveness", *Journal of Systems and Software*, Vol. 52 No. 1, pp. 3-10.

Joreskog, K.G. and Sorbom, D. (1989), *LISREL 7: A Guide to the Program and Applications*, 2nd ed., SPSS, Chicago, IL.

Jun, L., Qiuzhen, W. and Qingguo, M. (2011), "The effects of project uncertainty and risk management on IS development project performance: a vendor perspective", *International Journal of Project Management*, Vol. 29 No. 7, pp. 923-933.

Jung, Y.J., Wang, J.W. and Wu, S. (2008), "Competitive strategy, TQM practice, and continuous improvement of international project management: a contingency study", *International Journal of Quality and Reliability Management*, Vol. 26 No. 2, pp. 161-183.

Kandelousi, N., Ooi, J. and Abdollahi, A. (2011), "Key success factors for managing projects", *World Academy of Science, Engineering and Technology*, Vol. 5 No. 11, pp. 1291-1296.

Kearns, G.S. and Sabherwal, R. (2007), "Antecedents and consequences of information systems planning integration", *IEEE Transactions on Engineering Management*, Vol. 54 No. 4, pp. 628-643.

KPMG (2013), "Project Management Survey Report 2013", KPMG, Wellington, available at: www.kpmg.com/NZ/en/IssuesAndInsights/ArticlesPublications/Documents/KPMG-Project-Management-Survey-2013.pdf

Lacity, M. and Willcocks, L. (2000), "Relationships in IT outsourcing: a stakeholder perspective", *Cutter Consortium*, Vol. 2 No. 3.

Lindvall, M., Basili, V., Boehm, B., Costa, P., *et al.* (2002), "Empirical findings in agile methods", *Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods – XP/Agile Universe*, *London*, pp. 81-92.

Little, T. (2005), "Context-adaptive agility: managing complexity and uncertainty", *IEEE Software*, Vol. 22 No. 3, pp. 28-35.

Livermore, J.A. (2008), "Factors that significantly impact the implementation of an agile software development methodology", *Journal of Software*, Vol. 3 No. 4, pp. 31-36.

MacCallum, R.C., Widaman, K.F., Zhang, S. and Hong, S. (1999), "Sample size in factor analysis", *Psychological Methods*, Vol. 4 No. 1, pp. 84-99.

Meyer, J.P. and Allen, N.J. (1997), *Commitment in the Workplace: Theory, Research, and Application*, Sage, Thousand Oaks, CA.

Miles, R.E. and Snow, C.C. (1978), *Organizational Strategy, Structure, and Process*, McGraw-Hill, San Francisco, CA.

Misra, S.C., Kumar, V. and Kumar, U. (2009), "Identifying some important success factors in adopting agile software development practices", *Journal of Systems and Software*, Vol. 82 No. 11, pp. 1869-1890.

Nah, F.F. and Delgado, S. (2006), "Critical success factors for enterprise resource planning implementation and upgrade", *The Journal of Computer Information Systems*, Vol. 46 No. 5, pp. 99-113.

Nasir, M.H. and Sahibuddin, S. (2011), "Critical success factors for software projects: a comparative study", *Scientific Research and Essays*, Vol. 6 No. 10, pp. 2174-2186.

Nidumolu, S.R. (1995), "The effect of coordination and uncertainty on software project performance: residual performance risk as an intervening variable", *Information System Research*, Vol. 6 No. 3, pp. 191-219.

Nidumolu, S.R. (1996), "A comparison of the structural contingency and risk-based perspectives on coordination in software-development projects", *Journal of Management Information System*, Vol. 13 No. 2, pp. 77-113.

Office of Government Commerce (OGC) (2009), *Managing Successful Projects with PRINCE2*, TSO, London.

Petit, Y. (2012), "Project portfolios in dynamic environments: organising for uncertainty", *International Journal of Project Management*, Vol. 30 No. 5, pp. 539-553.

Pinto, J. and Prescott, J. (1988), "Variations in critical success factors over the stages in the project life cycle", *Journal of Management*, Vol. 14 No. 1, pp. 5-18.

PMI (2013a), *Pulse of the Profession Report 2013*, Project Management Institute Inc., Newtown Square, PA, available at: www.pmi.org/Knowledge-Center/Pulse/~/media/PDF/Business-Solutions/PMI-Pulse%20Report-2013Mar4.ashx

PMI (2013b), *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 5th ed., Project Management Institute, Newtown Square, PA.

Podsakoff, P.M., Mackenzie, S.B., Lee, J. and Podsakoff, N.P. (2003), "Common methods biases in behavioural research: a critical review of the literature and recommended remedies", *Journal of Applied Psychology*, Vol. 88 No. 5, pp. 879-903.

Preacher, K.J. and MacCallum, R.C. (2002), "Exploratory factor analysis in behavior genetics research: factor recovery with small sample sizes", *Behavior Genetics*, Vol. 32 No. 2, pp. 153-161.

Rai, A. and Hindi, A.H. (2000), "The effects of development process modeling and task uncertainty on development quality performance", *Information & Management*, Vol. 37 No. 6, pp. 335-346.

Ramesh, B., Mohan, K. and Cao, L. (2012), "Ambidexiterity in agile distributed development: an empirical investigation", *Information System Research*, Vol. 23 No. 2, pp. 323-339.

Ratbe, D., King, W.R. and Kim, Y.G. (2000), "The fit between project characteristics and application development methodologies: a contingency approach", *Journal of Computer Information Systems*, Vol. 40 No. 2, pp. 26-33.

Sabherwal, R. (2003), "The evolution of coordination in outsourced software development projects: a comparison of client and vendor perspectives", *Information and Organization*, Vol. 13 No. 3, pp. 153-202.

Sauer, C. and Cuthbertson, C. (2003), "The state of IT project management in UK 2002-2003", *Computer Weekly*, 15 April.

Sauser, B.J., Reilly, R.R. and Shenhar, A.J. (2009), "Why projects fail? How contingency theory can provide new insights – a comparative analysis of NASA's mars climate orbiter loss", *International Journal of Project Management*, Vol. 27 No. 7, pp. 665-679.

Schmidt, J.B. and Calantone, R.J . (1998), "Are really new product development projects harder to shut down?", *Journal of Product Innovation Management*, Vol. 15 No. 2, pp. 111-123.

Sheffield, J. and Lemétayer, J. (2013), "Factors associated with the software development agility of successful projects", *International Journal of Project Management*, Vol. 31 No. 3, pp. 459-472.

Shenhar, A. (2008), "Unleashing the power of project management", *Industrial Management*, Vol. 50 No. 1, pp. 14-18.

Shenhar, A.J. (2001), "One size does not fit all projects: exploring classical contingency domains", *Management Science*, Vol. 47 No. 3, pp. 394-414.

Shenhar, A. and Dvir, D. (2007), *Reinventing Project Management: The Diamond Approach to Successful Growth and Innovation*, Harvard Business School Press, Boston, MA.

Standish Group International Inc. (2012), *CHAOS 2012 Report Agile Projects Successful 3X More Than Non-agile Projects*, Standish Group International Inc., Boston, MA, available at: www.mountaingoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall (accessed March 2017).

Straub, D.W. (1989), "Validating Instruments in MIS research", *MIS Quarterly*, Vol. 13 No. 2, pp. 147-169.

Strode, D.E., Huff, S.H. and Tretiakov, A. (2009), "The impact of organizational culture on agile method use", *42nd Hawaii International Conference on System Sciences, HICSS, HI*, pp. 1-9.

Sudhakar, P. (2012), "A model of critical success factors for software development", *Journal of Enterprise Information Management*, Vol. 25 No. 6, pp. 537-558.

Swink, M. (2000), "Technological innovativeness as a moderator of new product design integration and top management support", *Journal of Product Innovation Management*, Vol. 17 No. 1, pp. 2008-2220.

Taylor, H. (2007), "Outsourced IT projects from the vendor perspective: different goals, different risks", *Journal of Global Information Management*, Vol. 15 No. 2, pp. 1-27.

Thompson, J.D. (1967), *Organizations in Action*, McGraw-Hill, New York, NY.

Tiwana, A. and Keil, M. (2004), "The one-minute risk assessment tool", *Communications of the ACM*, Vol. 47 No. 11, pp. 73-77.

Turner, R. and Zoli, R. (2012), "Forecasting success on large projects: developing reliable scales to predict multiple perspectives by multiple stakeholders over multiple time frames", *Project Management Journal*, Vol. 33.

Venkatraman, N. (1989), "The concept of fit in strategy research: toward a verbal and statistical correspondence", *Academy of Management Review*, Vol. 14 No. 3, pp. 423-444.

Vinekar, V., Slinkman, C.W. and Nerur, S. (2006), "Can agile and traditional systems development approaches coexist? An ambidextrous view", *Information Systems Management*, Vol. 23 No. 3, pp. 31-42.

Wallace, L., Keil, M. and Rai, A. (2004), "How software project risk affects project performance: an investigation of the dimensions of risk and exploratory model", *Decision Sciences*, Vol. 35 No. 2, pp. 289-321.

Wan, J. and Wang, R. (2010), "Empirical research on critical success factors of agile software process improvement", *Software Engineering and Applications*, Vol. 3 No. 12, pp. 1131-1140.

White, D. and Fortune, J. (2002), "Current practice in project management – an empirical study", *International Journal of Project Management*, Vol. 20 No. 1, pp. 1-11.

Wysocki, R.K. (2009), *Effective Project Management: Traditional, Agile, Extreme*, Wiley, Indianapolis, IN.

Wysocki, R.K. (2011), *Executives Guide to Project Management*, Wiley, Hoboken, NJ.

Yetton, P., Martin, A., Sharma, R. and Johnston, K. (2000), "A model of information systems development project performance", *Information Systems Journal*, Vol. 10 No. 4, pp. 263-289.

Young, R. and Jordan, E. (2008), "Top management support: mantra or necessity?", *International Journal of Project Management*, Vol. 26 No. 7, pp. 713-725.

Zwikael, O. and Globerson, S. (2006), "From critical success factors to critical success processes", *International Journal of Production Research*, Vol. 44 No. 17, pp. 3433-3449.

Zwikael, O. and Smyrk, J. (2012), "A general framework for gauging the performance of initiatives to enhance organizational value", *British Journal of Management*, Vol. 23 No. 1, pp. S6-S22.

Zwikael, O., Pathak, R.D., Singh, G. and Ahmed, S. (2014), "The moderating effect of risk on the relationship between planning and success", *International Journal of Project Management*, Vol. 32 No. 3, pp. 435-441.

Further reading

Arrindell, W.A. and van der Ende, J. (1985), "An empirical test of the utility of the observations-to-variables ratio in factor and components analysis", *Applied Psychological Measurement*, Vol. 9 No. 2, pp. 165-178.

Cockburn, A. (2000), "Selecting a project's methodology", *IEEE Software*, Vol. 17 No. 4, pp. 64-71.

Hair, J.F.J., Anderson, R.E., Tatham, R.L. and Black, W.C. (1995), *Multivariate Data Analysis*, 4th ed., Prentice Hall, Upper Saddle River, NJ.

Velicer, W.F. and Fava, J.L. (1998), "Effects of variable and subject sampling on factor pattern recovery", *Psychological Methods*, Vol. 3, pp. 231-251.

(The Appendix follows overleaf.)

Appendix

JEIM Supplementary Online appendix to "Empirical Comparison of Traditional Plan-based and Agile Methodologies: Critical Success Factors for Outsourced Software Development Projects from Vendors' perspective".

1. Copy of Qualtrics questionnaire

Qualtrics Survey Software Page 2 of 11

Section A: Background information (Please tick in the box of your choice). Please note that all these questions relate to your last completed software development project irrespective of whether it was successful or not.

How long have you been involved in software development projects?

- Less than 1 year
- 1 to less than 2 years
- 2 to less than 5 years
- 5 to less than 10 years
- More than 10 years

What was the size of the client organization in which your last software development project was conducted?

- 1-10
- 11-50
- 51-100
- 101-500
- 501-1000
- 1001-5000
- More than 5000

In what type of sector was your recent software development project conducted?

- Public sector
- Private sector
- Other- please specify

In what industry was your last software development project conducted?

- Finance/Insurance
- Manufacturing
- Marketing/retail
- Health
- Consulting
- Software
- Transportation
- Utility

https://vuw.qualtrics.com/ControlPanel/Ajax.php?action=GetSurveyPrint... 9/02/2014

- Aerospace
- Education
- Other – please specify

What was your position on that software development project?

- Project Manager
- Team Leader
- Developer
- Tester
- Other – please specify

What project management methodology was used?

- PRINCE2
- PMI (PMBOK)
- Agile project management
- Other – please specify

Who made the decision of choosing the methodology that was used?

- Top management
- Project team members
- It is an organizational policy to use this methodology
- Do not know
- Other-please specify

On which development life cycle was your last software development project based?

- Linear or Waterfall
- Iterative
- Incremental
- Adaptive
- Other-please specify

Qualtrics Survey Software

Page 4 of 11

What was the estimated total number of people who worked on this project from both the vendor and client organizations?

- 2-5
- 6-100
- 101-500
- 501-1000
- 1001-5000
- Above 5000

What was the estimated project budget in USA Dollars?

- Less than $100,000
- $100,000 to less than $1M
- $1M to less than $10M
- $10M to less than $100M
- More than $100M

What was the estimated duration of your last software development project?

- Less than 6 months
- 6months to less than 12 months
- 12 months to less than 24 months
- 24 months to less than 36 months
- More than 36 months

In which country was this software development project conducted? (Please indicate the major location for international software development projects)

Section B: Kindly indicate your responses for the following factors based on your last completed software development project irrespective of whether it was successful or unsuccessful. Please, indicate the extent to which you agree or disagree with the following statements by ticking the box of your choice.The word 'organization' refers to the client or customer organization for which the software was developed.

Top level management commitment. A 'project champion' is not a project manager or team leader but a person who provides moral, psychological and physical support to the team and provides them with needed resources and advocates for the project's benefits and advantages to its stakeholders.

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
There was approval and support from top level management.	○	○	○	○	○	○	○
Top level management publicly and explicitly identified the project as a top priority.	○	○	○	○	○	○	○
The project had a project champion.	○	○	○	○	○	○	○
The project champion was very committed to the realization of project benefits	○	○	○	○	○	○	○
There was continued project participation from top level management.	○	○	○	○	○	○	○

Internal project communication. A 'project team' means a 'team' whose members usually are assigned to activities for the same project. In this questionnaire 'team' is used to refer to 'project team'.

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
The project team met frequently.	○	○	○	○	○	○	○
Project team members were kept informed about the major decisions concerning the project.	○	○	○	○	○	○	○
Every effort was made to keep project team turnover at a minimum.	○	○	○	○	○	○	○
Project team members actively participated in the definition of project goals and schedules.	○	○	○	○	○	○	○
The amount of information disseminated by project team leaders was satisfactory.	○	○	○	○	○	○	○

Organizational culture

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
The organization culture was flexible and participative, valued teamwork and encouraged social interaction.	○	○	○	○	○	○	○
The management style was that of leadership and collaboration.	○	○	○	○	○	○	○
The organization was based on loyalty and mutual trust and commitment.	○	○	○	○	○	○	○
The organization valued feedback and learning.	○	○	○	○	○	○	○
The organization enabled empowerment of people and teams.	○	○	○	○	○	○	○

Qualtrics Survey Software Page 6 of 11

User participation (a user refers to the client or beneficiary of the software development project including the general public).

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
Users actively participated in requirements definition.							
The project team kept users informed concerning project progress and problems.							
Users formally evaluated the work done by the project team.							
Users formally approved the work done by the project team.							

Project team commitment

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
No other activities could match the benefits that the software development project activities presented to me.							
It would have been very hard for me to abandon the software development project activities even if I wanted to.							
Team members were emotionally attached to the software development project.							
Team members felt a strong sense of belonging to the software development project.							
Team members had a sense of obligation to the client of software development project.							
Team members owed a great deal to the software development project.							

Technical complexity

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
Project involved the use of new technology.							
Project had high level of technical complexity.							
Project involved the use of technology that had not been used in prior projects.							
Project involved computers, databases and networks that were highly complex.							
The software e.g operating system, procedures, algorithms and programming language, was technically complex.							

Development team skill

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
The development team lacked experience with the development platform/ environment used in this project.	○	○	○	○	○	○	○
The development team was very unfamiliar with this type of application.	○	○	○	○	○	○	○
The development team lacked knowledge of application domain involved in this project.	○	○	○	○	○	○	○
The development team generally lacked technically competent and experienced people who had developed similar software in the past.	○	○	○	○	○	○	○

Client experience level

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
The client was not familiar with this type of application.	○	○	○	○	○	○	○
The client didn't know what they wanted.	○	○	○	○	○	○	○
The client didn't have a good understanding of the problems they want solved.	○	○	○	○	○	○	○
The client was not familiar with data processing as a work tool.	○	○	○	○	○	○	○
The client was not aware of the importance of their roles in successfully completing the project.	○	○	○	○	○	○	○

Project criticality (potential damage that the project could cause to the client if there was a defect or failure)

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
A defect only could have a very minor impact leading to loss of comfort.	○	○	○	○	○	○	○
A defect could be fixed pretty easily because of good backup procedures leading to loss of discretionary monies.	○	○	○	○	○	○	○
A defect in the system could make the company go bankrupt leading to loss of essential monies.	○	○	○	○	○	○	○
A defect could be catastrophic, leading to loss of lives such as failure in nuclear power stations or flight control system.	○	○	○	○	○	○	○

Project planning and controlling

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
Special attention was paid to project planning.							
Project milestones were clearly defined.							
Project progress was monitored closely e.g. using PERT or CPM tools.							
Periodic formal status reports were used instead of the project plan.							
There were reviews at each milestone.							

Change management

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
There was recognition for the need for change.							
There was organizational wide change culture and structure.							
There was commitment to change.							
There was business process re-engineering to match the new software.							
Analysis of user feedback was continuously done.							

Leadership characteristics

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
The client's organizational leaders were truthful and did what they said.							
The client's organizational leaders enjoyed taking risks.							
The client's organizational leaders had technical expertise in their businesses.							
The client's organizational leaders were perceptive and conceptually skilled.							
The client's organizational leaders were approachable with good interpersonal skills and were liked in their positions.							

Vision and mission

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
The organization had a strategic business plan and a vision.							
The organizational had a mission with clearly defined goals.							
There was a business case approval for justification for investment in software.							
The project realization benefits were in line with the organization's vision.							
The project was part of the strategy to achieve the organizational goals.							

Project team composition

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
There were best people available on the team.							
The team was cross functional and balanced in its composition.							
Team members were full time on the software development project.							
The team and incentives supported partnerships, trust, risk-sharing.							
The teams were empowered decision makers.							
The team was comprised of the necessary business and technical knowledge of team members and consultants.							

User support

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
Users had a positive opinion about the system meeting their needs.							
Users were enthusiastic about the project.							
Users were an integral part of the development team.							
Users were available to answer the questions.							
Users were ready to accept the changes the system entailed.							

Technological uncertainty

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
There was lack of a clearly known way to develop software that would meet requirements specifications							
No available knowledge was of great help in developing software that would meet requirements specifications.							
No established procedures and practices could be relied upon to develop software that would meet requirements specifications							
There was lack of an understandable sequence of steps that could be followed for developing software to meet requirements							

Relative project size (i.e. compared to other software projects developed in the cleint's organization)

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
The scheduled number of person-days for completing this project was much higher.							
The scheduled number of months for completing this project was much higher.							
The dollar budget allocated to this project was much higher.							
The number of people on team was much larger.							
The overall project size was much larger.							

Specification changes

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
Requirements fluctuated quite a bit in earlier phases.							
Requirements fluctuated quite a bit in later phases.							
Requirements identified at beginning of the project were quite different from those existing at end.							
Requirements may fluctuate quite a bit in the future after the project is completed.							

Qualtrics Survey Software

Project success

	Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Agree	Strongly Agree
The project was completed within budget.							
The project was completed within schedule.							
The project scope was met.							
The software developed is reliable.							
The application developed is easy to use.							
Flexibility of the system is good.							
The system meets users' intended functional requirements.							
Users were satisfied with the system delivered.							
The project team was satisfied.							
Top level management of the client's organization was satisfied.							
The overall quality of the delivered application is high.							

Please kindly write down any general comments that you may have about this questionnaire or the whole study in general.

2. Supplementary tables

Constructs and measurement items	Traditional methodology, $n=513$ Mean	SD	Agile methodology, $n=471$ Mean	SD
Top level management support				
1 – There was approval and support from top level management	5.95	1.03	5.85	1.14
2 – Top level management publicly and explicitly identified the project as a top priority	5.55	1.26	5.69	1.23
3 – The project had a project champion	5.19	1.50	5.29	1.47
4 – The project champion was very committed to the realization of project benefits	5.24	1.49	5.32	1.46
5 – There was continued project participation from top level management	5.19	1.50	5.26	1.39
Internal project communication				
1 – The project team met frequently	6.11	.94	6.15	1.04
2 – Members were kept informed about the major decisions concerning the project	5.99	1.02	5.98	1.08
3 – Every effort was made to keep project team turnover at a minimum	5.53	1.25	5.42	1.35
4 – Members actively participated in the definition of project goals and schedules	5.37	1.25	5.42	1.35
5 – The amount of information disseminated by project team leaders was satisfactory	5.63	1.03	5.52	1.24
Organizational culture				
1 – Flexible and participative, valued teamwork and encouraged social interaction	5.14	1.36	5.29	1.35
2 – The management style was that of leadership and collaboration	5.29	1.34	5.24	1.44
3 – The organization was based on loyalty and mutual trust and commitment	5.24	1.34	5.18	1.44
4 – The organization valued feedback and learning	5.09	1.37	5.09	1.50
5 – The organization enabled empowerment of people and teams	5.03	1.37	5.09	1.49
User participation				
1 – Users actively participated in requirements definition	5.57	1.21	5.64	1.31
2 – The project team kept users informed concerning project progress and problems	5.67	1.13	5.68	1.32
3 – Users formally evaluated the work done by the project team	5.35	1.34	5.39	1.39
4 – Users formally approved the work done by the project team	5.58	1.21	5.58	1.45
Project team commitment				
1 – No other activities could match the benefits that the project activities presented to me	4.89	1.17	4.88	1.26
2 – Hard for me to abandon the software development project activities	5.14	1.25	5.02	1.37
3 – Team members were emotionally attached to the software development project	4.98	1.27	5.14	1.21
4 – Team members felt a strong sense of belonging to the software development project	5.25	1.24	5.34	1.18
5 – Team members had a sense of obligation to the client of software development project	5.40	1.09	5.34	1.19
6 – Team members owed a great deal to the software development project	4.99	1.27	5.00	1.24
Technical complexity				
1 – Project involved the use of new technology	5.35	1.47	5.54	1.27
2 – Project had high level of technical complexity	5.60	1.27	5.65	1.16
3 – Project involved the use of technology that had not been used in prior projects	4.78	1.68	5.28	1.45

Table AI.
Mean differences for group comparisons

(*continued*)

Constructs and measurement items	Traditional methodology, $n = 513$ Mean	SD	Agile methodology, $n = 471$ Mean	SD
4 – Project involved computers, databases and networks that were highly complex	5.32	1.46	5.41	1.29
5 – The operating system, procedures, algorithms and programming was complex	5.17	1.38	5.37	1.25
Development team skills				
1 – Team lacked experience with the development platform/environment used	3.45	1.60	3.58	1.57
2 – The development team was very unfamiliar with this type of application	3.18	1.49	3.25	1.49
3 – The development team lacked knowledge of application domain involved in this project	3.17	1.53	3.36	1.52
4 – Team generally lacked technically competent and experienced people	3.06	1.56	3.00	1.48
User experience				
1 – The client was not familiar with this type of application	4.04	1.62	3.84	1.57
2 – The client did not know what they wanted	3.83	1.66	3.84	1.58
3 – The client did not have a good understanding of the problems they want solved	3.73	1.60	3.72	1.57
4 – The client was not familiar with data processing as a work tool	3.63	1.58	3.56	1.57
5 – The client was not aware of the importance of their roles in the project	3.69	1.72	3.69	1.64
Planning and controlling				
1 – Special attention was paid to project planning	5.65	1.24	5.19	1.31
2 – Project milestones were clearly defined	5.77	1.11	5.21	1.33
3 – Project progress was monitored closely, e.g. using PERT or CPM tools	5.25	1.46	4.34	1.62
4 – Periodic formal status reports were used instead of the project plan	5.31	1.40	4.85	1.47
5 – There were reviews at each milestone	5.51	1.31	5.19	1.39
Change management				
1 – There was recognition for the need for change	5.44	1.17	5.48	1.13
2 – There was organizational wide change culture and structure	4.60	1.52	4.64	1.40
3 – There was commitment to change	4.86	1.40	4.92	1.30
4 – There was business process re-engineering to match the new software	4.66	1.52	4.62	1.42
5 – Analysis of user feedback was continuously done	4.96	1.34	4.78	1.39
Leadership characteristics				
1 – The client's organizational leaders were truthful and did what they said	5.08	1.32	5.07	1.29
2 – The client's organizational leaders enjoyed taking risks	4.00	1.44	4.14	1.40
3 – The client's organizational leaders had technical expertise in their businesses	4.97	1.36	4.87	1.36
4 – The client's organizational leaders were perceptive and conceptually skilled	5.04	1.24	4.92	1.29
5 – Leaders were approachable with good interpersonal skills and liked in their positions	5.10	1.27	5.04	1.36
Vision and mission				
1 – The organization had a strategic business plan and a vision	5.58	1.18	5.34	1.21
2 – The organization had a mission with clearly defined goals	5.56	1.17	5.32	1.19
3 – There was a business case approval for justification for investment in software	5.50	1.26	5.34	1.23
4 – The project realization benefits were in line with the organization's vision	5.60	1.06	5.50	1.08
5 – The project was part of the strategy to achieve the organizational goals	5.73	1.07	5.66	1.08
Team composition				
1 – There were best people available on the team	5.28	1.16	5.15	1.23
2 – The team was cross-functional and balanced in its composition	5.36	1.17	5.23	1.19

(*continued*)

Table AI.

Constructs and measurement items	Traditional methodology, $n = 513$ Mean	SD	Agile methodology, $n = 471$ Mean	SD
3 – Team members were full time on the software development project	4.84	1.59	5.14	1.38
4 – The team and incentives supported partnerships, trust and risk-sharing	4.81	1.35	4.75	1.32
5 – The teams were empowered decision makers	4.83	1.31	4.90	1.38
6 – Team comprised of the necessary business and technical knowledge of consultants	5.48	1.09	5.26	1.19
User support				
1 – Users had a positive opinion about the system meeting their needs	5.26	1.10	5.29	1.12
2 – Users were enthusiastic about the project	5.12	1.18	5.22	1.17
3 – Users were an integral part of the development team	4.69	1.40	4.58	1.39
4 – Users were available to answer the questions	5.29	1.16	5.00	1.33
5 – Users were ready to accept the changes the system entailed	4.97	1.26	4.99	1.22
Technological uncertainty				
1 – Lack of a clearly known way to develop software …	3.48	1.52	3.61	1.50
2 – No available knowledge was of great help in developing software …	3.12	1.37	3.24	1.36
3 – No established procedures and practices could be relied upon to develop software …	3.07	1.42	3.20	1.41
4 – Lack of an understandable sequence of steps that could be followed for developing …	3.02	1.46	3.14	1.40
Relative project size (compared to previous projects)				
1 – The scheduled number of person-days for completing this project was much higher	4.17	1.48	4.04	1.42
2 – The scheduled number of months for completing this project was much higher	4.07	1.51	3.94	1.40
3 – The dollar budget allocated to this project was much higher	3.87	1.52	3.81	1.42
4 – The number of people on team was much larger	3.63	1.50	3.35	1.36
6 – The overall project size was much larger	3.97	1.58	3.84	1.45
Specification changes				
1 – Requirements fluctuated quite a bit in earlier phases	5.04	1.35	5.05	1.27
2 – Requirements fluctuated quite a bit in later phases	4.34	1.53	4.54	1.38
3 – Requirements identified at beginning of the project were quite different from …	3.93	1.59	4.41	1.39
4 – Requirements may fluctuate quite a bit in the future after the project is completed	4.20	1.51	4.54	1.33
Process project success				
1 – The project was completed within budget	4.78	1.57	4.70	1.47
2 – The project was completed within schedule	4.44	1.65	4.54	1.56
3 – The project scope was met	5.49	1.23	5.45	1.17
Product project success				
1 – The software developed is reliable	5.69	1.01	5.64	1.04
2 – The application developed is easy to use	5.54	1.06	5.54	1.02
3 – Flexibility of the system is good	5.41	1.11	5.46	1.09
4 – The system meets users' intended functional requirements	5.71	1.14	5.75	1.87
5 – Users were satisfied with the system delivered	5.43	1.13	5.49	1.05
6 – The project team was satisfied	5.45	1.15	5.43	1.14
7 – Top level management of the client's organization was satisfied	5.51	1.13	5.48	1.12
8 – The overall quality of the delivered application is high	5.53	1.08	5.49	1.12

Table AI.

Measurement items to constructs	Traditional plan-based methodology, $n = 513$ Standardized loadings	Agile methodology projects data set, $n = 471$ Standardized loadings
Top level management support		
1 – Continued project participation from top level mgt	0.84***	0.85***
2 – Top mgt. publicly and explicitly identified the project	0.78***	0.82***
3 – The project had a project champion	0.77***	0.75***
4 – Approval and support from top level management	0.76***	0.81***
Internal project communication		
1 – The project team met frequently	0.77***	0.73***
2 – Members were kept informed about the major decisions	0.79***	0.85***
3 – Members actively participated in the definition of project	0.72***	0.73***
4 – Information disseminated was satisfactory	0.74***	0.83***
Organizational culture		
1 – Flexible and participative and valued teamwork	0.86***	0.87***
2 – The mgt. style was that of leadership and collaboration	0.89***	0.90***
3 – Organ. based on loyalty and mutual trust and commitment	0.89***	0.88***
4 – The organization valued feedback and learning	0.89***	0.90***
5 – The organization enabled empowerment of people and teams	0.88***	0.90***
User participation		
1 – Users formally approved the work done by the project team	0.76***	0.82***
2 – Users formally evaluated the work done by the project team	0.75***	0.85***
3 – Users informed concerning project progress and problems	0.75***	0.81***
4 – Users actively participated in requirements definition	0.76***	0.81***
Project team commitment		
1 – Team members were emotionally attached to the project	0.80***	0.83***
2 – Team members felt a strong sense of belonging	0.89***	0.90***
3 – Team members had a sense of obligation to the client	0.80***	0.71***
4 – Team members owed a great deal to the project	0.76***	0.67***
Technical complexity		
1 – Operating system, procedures and programming was complex	0.89***	0.87***
2 – Computers, databases and networks that were highly complex	0.84***	0.85***
3 – Project had high level of technical complexity	0.78***	0.76***
Development team skills		
1 – Team lacked experience with the development platform	0.86***	0.82***
2 – Team was very unfamiliar with this type of application	0.83***	0.84***
3 – Lacked knowledge of application domain involved	0.88***	0.80***
4 – Team generally lacked technically competent people	0.86***	0.84***
User experience		
1 – Client was not aware of the importance of their roles	0.82***	0.83***
2 – Client was not familiar with data processing as a work tool	0.79***	0.80***
3 – Client did not have a good understanding of the problems	0.90***	0.90***
4 – The client did not know what they wanted	0.84***	0.84***
Planning and controlling		
1 – Special attention was paid to project planning	0.86***	0.81***
2 – Project milestones were clearly defined	0.85***	0.83***
3 – Progress was monitored closely, e.g. using PERT or CPM tools	0.78***	0.76***
4 – There were reviews at each milestone	0.65***	0.64***

(*continued*)

Table AII. Standardized regression weights from the measurement models – group comparisons

Measurement items to constructs	Traditional plan-based methodology, $n = 513$ Standardized loadings	Agile methodology projects data set, $n = 471$ Standardized loadings
Change management		
1 – Business process re-engineering to match the new software	0.74***	0.68***
2 – There was commitment to change	0.90***	0.83***
3 – There was organizational wide change culture and structure	0.83***	0.73***
4 – There was recognition for the need for change	0.76***	0.69***
Leadership characteristics		
1 – Leaders were approachable with good interpersonal skills	0.79***	0.84***
2 – Leaders were perceptive and conceptually skilled	0.86***	0.90***
3 – Leaders had technical expertise in their businesses	0.77***	0.78***
4 – Leaders were truthful and did what they said	0.76***	0.82***
Vision and mission		
1 – The project was part of the strategy	0.73***	0.69***
2 – Realization benefits were in line with the vision	0.76***	0.73***
3 – The organization had a mission with clearly defined goals	0.91***	0.89***
4 – The organization had a strategic business plan and a vision	0.90***	0.88***
Team composition		
1 – Team comprised of necessary technical knowledge and consultants	0.81***	0.80***
2 – The teams were empowered decision makers	0.72***	0.75***
3 – The team was cross-functional and balanced	0.78***	0.80***
4 – There were best people available on the team	0.70***	0.75***
User support		
1 – Users were ready to accept the changes the system entailed	0.78***	0.79***
2 – Users were available to answer the questions	0.72***	0.74***
3 – Users were an integral part of the development team	0.66***	0.69***
4 – Users were enthusiastic about the project	0.78***	0.80***
5 – Users had a positive opinion about the system	0.81***	0.82***
Specification changes		
1 – Requirements may fluctuate quite a bit in the future	0.77***	0.66***
2 – Requirements identified at beginning were quite different	0.86***	0.78***
3 – Requirements fluctuated quite a bit in later phases	0.81***	0.73***
Technological uncertainty		
1 – Lack of a clearly known way to develop software	0.82***	0.79***
2 – No available knowledge was of great help in developing software	0.84***	0.86***
3 – No established procedures and practices could be relied upon	0.89***	0.92***
4 – Lack of an understandable sequence of steps that could be followed	0.90***	0.86***
Relative project size		
1 – The overall project size was much larger	0.87***	0.78***
2 – The number of people on team was much larger	0.86***	0.76***
3 – The dollar budget allocated to this project was much higher	0.91***	0.84***
4 – Months for completing this project were much higher	0.87***	0.89***
5 – Person-days for completing this project were much higher	0.87***	0.89***

Table AII. (*continued*)

Measurement items to constructs	Traditional plan-based methodology, $n = 513$ Standardized loadings	Agile methodology projects data set, $n = 471$ Standardized loadings
Process project success		
1 – The project was completed within budget	0.86***	0.81***
2 – The project was completed within schedule	0.85***	0.82***
3 – The project scope was met	0.73***	0.68***
Product project success		
1 – The software developed is reliable	0.78***	0.82***
2 – The application developed is easy to use	0.78***	0.77***
3 – Flexibility of the system is good	0.75***	0.75***
4 – The system meets users' intended functional requirements	0.87***	0.86***
5 – Users were satisfied with the system delivered	0.87***	0.88***
6 – The project team was satisfied	0.85***	0.86***
7 – Top level management was satisfied	0.83***	0.84***
8 – The overall quality of the delivered application is high	0.88***	0.89***

Notes: All regression factor loadings are standardized and greater than 0.6 for convergent validity (Straub, 1989; Bagozzi and Yi, 1988). ***Significant at $p < 0.001$

Table AII.

Table AIII.
Correlations for traditional plan-based methodology projects data set

Construct	TMS	OC	PC	LC	CM	VM	IPC	PTC	TE	TC	UP	US	UE	TCO	TU	RPS	SC	PRC	PRD
Top mgt. support (TMS)	*0.58*	0.26	0.24	0.19	0.17	0.24	0.24	0.18	0.01	0.19	0.23	0.14	0.05	0.05	0.03	0.01	0.01	0.08	0.12
Organ-culture (OC)	0.54***	*0.77*	0.25	0.23	0.31	0.25	0.32	0.19	0.03	0.36	0.25	0.27	0.09	0.05	0.08	0.0	0.05	0.20	0.21
Planning and controlling (PC)	0.49***	0.51***	*0.60*	0.11	0.23	0.23	0.45	0.20	0.02	0.30	0.29	0.28	0.09	0.08	0.13	0.00	0.11	0.23	0.23
Leadership characteristics (LC)	0.46***	0.49***	0.34***	*0.64*	0.22	0.24	0.11	0.08	0.01	0.22	0.25	0.36	0.22	0.02	0.04	0.00	0.12	0.22	0.22
Change management (CM)	0.44***	0.57***	0.47***	0.48***	*0.58*	0.21	0.15	0.11	.00	0.26	0.15	0.24	0.06	0.06	0.03	0.00	0.03	0.17	0.21
Vision and mission (VM)	0.49***	0.51***	0.47***	0.49***	0.46***	*0.67*	0.12	0.09	0.04	0.22	0.17	0.19	0.13	0.05	0.06	0.00	0.04	0.12	0.15
Int. proj. communication (IPC)	0.49***	0.57***	0.68***	0.33***	0.39***	0.35***	*0.56*	0.23	0.02	0.30	0.34	0.26	0.05	0.09	0.08	0.00	0.09	0.20	0.25
Proj. team commitment (PTC)	0.45***	0.45***	0.45***	0.28***	0.33***	0.31***	0.48***	*0.66*	0.00	0.35	0.15	0.21	0.00	0.11	0.01	0.00	0.00	0.12	0.16
Dev. Team expertise (TE)	−0.08	−0.15***	−0.13**	−0.13*	−0.02	−0.20***	−0.15**	−0.01	*0.72*	0.08	0.02	0.02	0.17	0.01	0.32	0.1	0.16	0.03	0.05
Team composition (TC)	0.45***	0.59***	0.54***	0.46***	0.51***	0.49***	0.55***	0.60***	−0.28***	*0.55*	0.32	0.35	0.07	0.10	0.07	0.00	0.03	0.21	0.28
User participation (UP)	0.50***	0.49***	0.54***	0.49***	0.39***	0.41***	0.58***	0.40***	−0.16**	0.56***	*0.55*	0.34	0.10	0.04	0.07	0.00	0.06	0.19	0.24
User support (US)	0.40***	0.53***	0.52***	0.59***	0.487***	0.44***	0.51***	0.45***	−0.15	0.60***	0.58***	*0.61*	0.19	0.02	0.07	0.00	0.15	0.31	0.28
User experience (UE)	0.25***	−0.32***	−0.31***	−0.48***	−0.25***	−0.36***	0.21***	0.05	0.42***	−0.27***	−0.31***	−0.43**	*0.70*	0.04	0.23	0.4	0.35	0.09	0.09
Technical complexity (TCO)	0.26***	0.23***	0.29***	0.14**	0.24***	0.21***	0.30***	0.35***	0.11*	0.31***	0.21***	0.13**	−0.01	*0.70*	0.01	0.04	0.01	0.01	0.06
Technology uncertainty (TU)	−0.19***	−0.28***	−0.60***	−0.23***	−0.17***	−0.25***	−0.29***	−0.08	0.56***	−0.26***	−0.26***	−0.3***	0.48***	0.08	*0.75*	0.10	0.29	0.07	0.12
Relative project size (RPS)	0.10*	−0.01	0.07	−0.03	0.04	0.04	−0.03	0.057	0.26***	0.03	−0.04	−0.05**	0.20***	0.21***	0.32***	*0.75*	0.13	0.05	0.03
Specification changes (SC)	−0.12*	−0.24	−0.34***	−0.36***	−0.17***	−0.19***	−0.30***	−0.04	0.39***	−0.18***	0.24***	−0.4***	0.59***	0.10***	0.54***	0.36***	*0.62*	0.14	0.09
Process project success (PRC)	0.12*	0.43*	0.49***	0.46***	0.42***	0.35***	0.46***	0.34***	−0.17***	0.47***	0.44***	0.54***	−0.31***	0.11*	−0.28***	−0.21***	−0.4***	*0.67*	0.57
Product project success (PRD)	0.37*	0.44*	0.49***	0.46***	0.47***	0.38***	0.49***	0.39***	−0.22***	0.54***	0.48***	0.52***	−0.29***	0.23***	−0.34***	−0.21***	−0.2***	0.79***	*0.69*

Notes: $n = 513$. Zero order (Pearson) correlations appear below the diagonal. Squared correlations are placed above the diagonal. Average variances extracted (AVE) values are indicated on the diagonal in italics. *,**,***Significant at 0.05, 0.01 and 0.001 levels, respectively

Table AIV.
Correlations for agile methodology projects data sets

Factor	TMS	OC	PC	LC	CM	VM	IPC	PTC	TE	TC	UP	US	UE	TCO	TU	RPS	SC	PRC	PRD
Top mgt. support (TMS)	*0.62*	0.33	0.17	0.21	0.27	0.22	0.26	0.11	0.06	0.19	0.16	0.16	0.05	0.09	0.03	0.00	0.01	0.05	0.09
Organ-culture (OC)	0.56***	*0.80*	0.2	0.38	0.33	0.29	0.45	0.31	0.11	0.44	0.26	0.26	0.04	0.03	0.03	0.00	0.02	0.26	0.31
Planning and controlling (PC)	0.43***	0.46***	*0.59*	0.19	0.32	0.25	0.30	0.12	0.02	0.28	0.26	0.17	0.06	0.05	0.02	0.00	0.07	0.18	0.18
Leadership characteristics (LC)	0.47***	0.61***	0.45***	*0.71*	0.35	0.40	0.24	0.31	0.02	0.31	0.28	0.39	0.13	0.03	0.02	0.00	0.03	0.22	0.21
Change management (CM)	0.51***	0.56***	0.55***	0.59***	*0.56*	0.27	0.28	0.26	0.01	0.31	0.22	0.23	0.02	0.11	0.00	0.01	0.00	0.12	0.15
Vision and mission (VM)	0.48***	0.55***	0.49***	0.63***	0.51***	*0.67*	0.15	0.19	0.01	0.27	0.20	0.24	0.04	0.10	0.03	0.00	0.01	0.10	0.17
Int. proj. communication (IPC)	0.50***	0.66***	0.54***	0.50***	0.53***	0.40***	*0.63*	0.35	0.08	0.43	0.30	0.28	0.02	0.06	0.04	0.01	0.02	0.26	0.32
Proj. team commitment (PTC)	0.32***	0.55***	0.36***	0.55***	0.52***	0.42***	0.60***	*0.65*	0.02	0.39	0.26	0.23	0.01	0.10	0.01	0.00	0.00	0.17	0.22
Dev. Team expertise (TE)	0.26***	−0.34***	−0.16***	−0.17**	−0.13*	−0.13***	−0.30***	−0.14**	*0.64*	0.11	0.00	0.03	0.23	0.00	0.29	0.05	0.09	0.07	0.09
Team composition (TC)	0.43***	0.66***	0.53***	0.57***	0.56***	0.53***	0.66***	0.61***	−0.35***	*0.62*	0.29	0.39	0.02	0.13	0.04	0.00	0.00	0.29	0.42
User participation (UP)	0.39***	0.50***	0.51***	0.53***	0.48***	0.45***	0.55***	0.50***	−0.04	0.53***	*0.70*	0.47	0.01	0.06	0.00	0.00	0.01	0.27	0.21
User support (US)	0.39***	0.51***	0.43***	0.61***	0.49***	0.50***	0.54***	0.49***	−.15**	0.62***	0.68	*0.63*	0.05	0.05	0.03	0.00	0.00	0.23	0.31
User experience (UE)	0.22***	−0.21***	−0.24***	−0.36***	−0.13*	−0.21***	−0.17**	−0.08	0.47***	−0.16*	−0.12*	−0.22***	*0.72*	0.00	0.22	0.06	0.34	0.04	0.02
Technical complexity (TCO)	0.28***	0.16***	0.22*	0.18***	0.35***	0.33***	0.25***	0.31***	−0.03	0.36***	0.24***	0.23***	0.03	*0.75*	0.00	0.05	0.03	0.00	0.02
Technological uncertainty (TU)	−0.17***	−0.17***	−0.16***	−0.14*	−0.07	−0.17*	−0.19***	−0.09	0.55***	−0.21***	−0.06***	−0.17	0.47***	0.05	*0.76*	0.10	0.16	0.01	0.07
Relative project size (RPS)	0.04	−0.07	−0.03	0.07	0.13	0.08	−0.09	−0.02	0.23***	0.03	−0.02	0.06	0.25***	0.21***	0.29	*0.73*	0.08	0.07	0.02
Specification changes (SC)	−0.07	−0.14***	−0.25***	−0.17*	0.02	−0.06***	−0.13*	0.04	0.31***	0.03	−0.09	−0.06	0.58***	0.18***	0.39**	0.30***	*0.53*	0.02	0.00
Process project success (PRC)	−0.12***	0.51***	0.42***	0.47***	0.34***	0.32***	0.50***	0.43***	−0.27***	0.53***	0.52***	0.48***	−0.19***	−0.01	−0.11*	−0.25***	−0.16*	*0.63*	0.54
Product project success (PRD)	−0.19***	0.56***	0.43***	0.45***	0.39***	0.43***	0.55***	0.45***	−0.29***	0.66***	0.46***	0.56***	−0.15*	0.12***	−0.25***	−0.13*	−0.07	0.75***	*0.68*

Notes: $n = 471$. Zero order (Pearson) correlations appear below the diagonal. Squared correlations are placed above the diagonal. Average variances extracted (AVE) values are indicated on the diagonal in italics. *,**,***Significant at 0.05, 0.01 and 0.001 levels, respectively

3. S5. Detailed discussion of the hypothesis tests

This paper empirically examines the differences between CSFs in the current context of PMM from vendors' perspective. The results largely support the proposed hypotheses. This evidence of associations supports and extends previous research.

H1: top management support (TMS) and project success

Overall, the results from testing for significant differences of coefficients across the two data sets indicate that the importance of TMS on project success is not greater for agile than for traditional plan-based methodologies, but is significantly different. These results were not as expected. When examining the traditional plan-based projects data set, there is an insignificant association between top level management support and process success ($\beta = -0.04$, $p > 0.05$) and product success ($\beta = -0.03$, $p > 0.05$), respectively, not supporting both *H1(T)*(i) and *H1(T)*(ii), respectively. On the other hand, for the agile projects data set, the association between top level management support and process success ($\beta = -0.16$, $p < 0.05$) had a significant but negative path coefficient, but is insignificantly associated with product success ($\beta = -0.05$, $p > 0.05$), not supporting both *H1(A)*(i) and *H1(A)*(ii), respectively.

TMS has often been reported in empirical studies as a CSF for software development projects. For instance, in a case-based study, Kandelousi *et al.* (2011) found that TMS impacted on traditional plan-based projects and in a survey conducted by White and Fortune (2002) to determine the current practice in project management, support of TMS ranked third out of 23 CSFs listed. The research carried out by Young and Jordan (2008) also concluded that TMS had a positive impact on traditional project success. Jung *et al.* (2008) findings provided support for the hypothesis that TMS was a significant predictor of project performance. However, these aforementioned studies focussed on in-house development projects where developers and users are members of the same organization, rather than the vendor's perspective employed in this study. Nonetheless, TMS was found to indirectly and positively influence process success through planning and controlling (P&C). This means that the TMS only contributes towards setting project plans and controls which if adhered to can improve project success.

For the agile data set, *H1(A)*(i) and *H1(A)*(ii) which theorized a significant positive relationships between TMS and both process and product project success, respectively, were also not supported. Instead, the results suggest that for agile projects, there is a significant negative association between TMS and process project success, but an insignificant association between TMS and product project success. Although these findings seem surprising, they are not new. Chow and Cao's (2008) study of 906 agile software projects did not find TMS as a significant CSF that contributes to the success of agile software development projects. On the other hand, however, these findings are inconsistent with other studies (e.g. Wan and Wang, 2010; Sheffield and Lemétayer, 2013; Imreh and Raisinghani, 2011). However, none of these studies focussed on the narrow vendor's perspective used in this study.

Nonetheless, the results in both models indicate positive and significant associations of TMS with internal project communication (IPC), project team commitment (PTC) and organizational culture (OC). This seems to raise the possibility that TMS has indirect influence on project success.

Further, although TMS is generally viewed as a positive factor, some researchers argue that what begins as support can easily turn into interference, which is considered counterproductive and detrimental especially in highly innovative environments (Young and Jordan, 2008). This is also consistent with Swink (2000) who found that the presence of TMS may not matter in really new product innovations. This could imply that in circumstances where agility is coupled with uncertainty and changes, new software development project success depends more on other aspects such as the presence of deep technological expertise. It is also highly likely that top management directives and controls often accompany TMS (Swink, 2000). Such directives and controls are potentially counterproductive to agile practices.

Additionally, a previous empirical study by Schmidt and Calantone (1998) demonstrated that some managers are prone to greater levels of optimism, psychological commitment and resource support for really new innovative product development projects even when performance forecasts are poor. Perhaps, this finding that TMS is ineffective in traditional plan-based and seemingly detrimental in agile software development projects efforts reflects over optimism. If top level managers find it difficult to do something that prevents an activity from continuing, especially to stop giving money on innovative software projects, then ceteris paribus they are likely to support a greater percentage of project failures (Schmidt and Calantone, 1998). This would also account for the finding that TMS is insignificant with both process and product project success in what are typically lower innovative contexts (traditional plan-based projects, but negatively and significantly associated with process project success for highly technological innovations (agile projects)).

H2: OC and project success

Overall the contribution of OC is not statistically greater for agile than for traditional plan-based methodologies, failing to support *H2 (TA)*. For traditional plan-based projects data set there are insignificant path association between OC and process success ($\beta = 0.02, p > 0.05$) and product success ($\beta = 0.01$, $p > 0.05$), respectively, supporting both *H2(T)*(i) and *H2(T)*(ii), respectively. For the agile data set there are significant path coefficients association between OC and process success ($\beta = 0.06, p < 0.05$) and product success ($\beta = 0.07, p < 0.05$), respectively, supporting both *H2(A)*(i) and *H2(A)*(ii), respectively.

H3: P&C and project success

Overall, the model comparison results from testing for significant differences of coefficients across the two data sets indicate that the importance of P&C is statistically greater for plan-based than agile methodologies on only process success but not product success. As initially hypothesized, for traditional plan-based projects, there is a positive and significant association between P&C and process success ($\beta = 0.28$, $p < 0.001$) and an insignificant path coefficient with product success ($\beta = 0.05$, $p > 0.05$), respectively, supporting *H3(T)*(i) but rejecting *H3(T)*(ii), respectively. These findings are in line with the views expressed by research participants during the pilot study. For instance, research participant 14 who uses traditional plan-driven methodologies said: "It is essential that a detailed plan is in place, this highlights conflicts, resource availability, etc. Also gives the project team a structure to work within. If there is no detailed project plan, then the project is doomed to fail from the start". Similarly, research participant 10 (traditional plan-based project management) said: "Planning and controlling are very important for software project success. Successful projects I know spent more than 60 per cent of the time on planning and controlling. This helps to set up tolerance levels and meet goals". Thus, for outsourced traditional plan-driven projects, this strongly suggests that P&C are essential for setting milestones and taking corrective actions where the outsourced SD project is not going on well as expected.

These findings concur with previous research which has demonstrated a positive relationship between project P&C and process performance. For instance, Jun *et al.* (2011) found that project P&C are positively correlated with process performance at low levels of uncertainty. Similarly, Yetton *et al.* (2000) found that project P&C were negatively related to budget variances and that budget variances are a negative function of planning. This implies that effective project P&C for traditional plan-based methods is likely to be associated with efficiencies in development and, thus, prevent budget, scope and time variances. These results also mirror empirical results of Wallace *et al.* (2004), which confirmed a negative relationship between P&C risks and both process and product project performance.

However, the structural path between project P&C and product project success for traditional plan-driven projects was insignificant, not supporting *H3(T)*(ii). This is consistent with Jun *et al.* (2011) who did not find a statistically significant relationship between P&C and product success. This finding indicates that better quality of project plans and controls does not necessarily lead to better quality of outsourced software development project products. This might be because better long-range project plans may be followed by poor planning at the project or team level and mistakes in terms of the assignment of personnel, requirement specification, less customer participation or feedback, etc. Thus, a better software project master plan should not be considered as an end-goal in itself; instead, it should be complemented with top managers' participation in IT resource allocation, customer interaction, proper assignment of personnel and requirement specification. These findings are consistent with recent Kearns and Sabherwal's (2007) empirical survey of 274 CIO's indicating that most outsourced software development projects practice intensive planning, controlling and alignment but continue to experience failed software quality products and IT investments.

On the other hand, for the agile projects data set, the association between P&C and process success ($\beta = 0.02$, $p > 0.05$) and product success ($\beta = 0.04$, $p > 0.05$), respectively, had insignificant positive path coefficients, supporting both *H3(A)*(i) and *H3(A)*(ii), respectively. Partially supporting *H3 (TA)*. This is consistent with the pilot study, for example, a research participant 9 who uses agile methodologies said: "For agile projects, detailed planning up-front is useless and dangerously misleading. Making estimates based on really rough order of magnitude estimates and comparisons to prior experience are useful for securing funding and gaining go-ahead, but are likely to change repeatedly as requirements are clarified through iterative processes. There are far too many unknowns in agile projects to be able to create a well-defined plan upfront. The up-front planning should be at the level needed to make the next decision. Detailed planning should be reserved for the near-term work, where there is a chance of being realistic and accurate".

This clearly suggests that the management of agile projects does not need formalized P&C since formal plans and controls may become absolute sooner than later due to continuous changes initiated by high levels of environmental uncertainty. By nature, this demonstrates that agile approaches are more risk oriented and cope with uncertainty, unlike traditional plan-based approaches. These findings are consistent with Misra *et al.* (2009), Jun *et al.* (2011) and Barki *et al.* (2001) that when dealing with high project uncertainty and change, P&C are insignificant predictors of process and product success.

H4: V&M and project success

Overall, the contribution of V&M is not statistically greater for agile than for traditional plan-based methodologies, failing to support *H4(TA)*. The direct structural path between V&M and process success ($\beta = 0.07$, $p < 0.05$) was significant providing support for *H4(T)*(i), while the structural path between V&M and product success was insignificant ($\beta = 0.02$, $p > 0.05$), failing to provide support for *H4(T)*(ii) in the traditional plan-driven projects model. The agile projects model, had insignificant direct structural paths between V&M and process success ($\beta = -0.02$, $p > 0.05$) and product success ($\beta = 0.01$, $p > 0.05$), respectively, thus, providing support for *H4(A)*(i) and *H4(A)*(ii), respectively. The direct structural paths between V&M and process and product project success were not significant in the model for the agile projects, thus, supporting *H4(A)*(i) and *H4(A)*(ii), respectively.

H5: CM and project success

The test results for significant differences of coefficients across the two data sets indicate that the importance of CM is statistically greater for agile than for traditional methodologies on product but not process success. For traditional plan-based projects data set, there significant positive path associations between CM and process success ($\beta = 0.06$, $p < 0.05$) and product success ($\beta = 0.07$, $p < 0.05$), respectively, providing support for both *H5(T)*(i) and *H5(T)*(ii), respectively. These findings are consistent with Nah and Delgado (2006) and Jung *et al.* (2008) who found that CM is essential for traditional software project success. CM is needed most especially for traditional plan-based projects where there is likelihood of resistance to change. Even though training tends to be one of the areas to be cut in the case of budget overruns, it is critical to 257 the success of the implementation of outsourced project change as well as the quality of decisions that will be taken based on the system. On the hand, there were significant positive relationships between CM and process success ($\beta = 0.08$, $p < 0.05$) and product success ($\beta = 0.21$, $p > 0.001$), respectively, for agile projects, supporting both *H5(A)*(i) and *H5(A)*(ii), respectively, partially supporting *H5(TA)*. The results suggest that agile methodologies require CM skills to deal with changes and uncertainty.

H6: LC and project success

The model results from testing for significant differences of coefficients across the two data sets indicate that the importance of LC is statistically greater for agile than for plan-based methodologies on only process but not product success. The construct of LC had a direct positive and significant association with process success ($\beta = 0.06$, $p < 0.05$) and an insignificant association with product success ($\beta = 0.05$, $p > 0.05$), respectively, supporting *H6(T)*(i) but rejecting *H6(T)*(ii), respectively.

The significant results between LC and process success are consistent with Lindvall *et al.*'s (2002) empirical findings that it is not only the experience and competency of the team members that are important, but also their LC such as honesty, collaborative attitude, sense of responsibility, readiness to learn, and work with others that are considered equally important for software development project success. Therefore, having project members with effective leadership on big teams is critical to traditional project success.

For the agile projects, LC had a direct positive and significant association with process success ($\beta = 0.22$, $p < 0.001$) and an insignificant association with product success ($\beta = -0.04$, $p > 0.05$), respectively, supporting *H6(A)*(i) but not *H6(A)*(ii), respectively. Partially supporting *H6(TA)*. The significant results between LC and process success are consistent with Misra *et al.* (2009) and Strode *et al.* (2009). Misra *et al.* (2009) established that for the successful implementation of agile projects to be completed on time, budget and scope, the project leaders need to possess some LC. Surprisingly, these results seem to suggest that LC significantly contribute only towards process success but not product success, irrespective of project types, i.e. traditional plan-based or agile projects. One possible explanation is that regardless of the project type, a project leader is required to handle people's emotions, to motivate people when they get disappointed, to take the teams together during the difficult times, and finally to make sure that they have concentrated on the goals (schedule, budget and scope) of the project.

H7: IPC and project success

Overall, the results from testing for significant differences of coefficients across the two data sets indicate that the importance of IPC is not statistically greater for agile than for traditional plan-based methodologies. The structural paths between IPC and process success ($\beta = 0.08$, $p < 0.05$) and product success ($\beta = 0.06$, $p < 0.05$), respectively, had significant positive associations for the traditional plan-driven projects, providing support for both *H7(T)*(i) and *H7(T)*(ii), respectively. Likewise, the structural paths between IPC and process success ($\beta = 0.09$, $p < 0.05$) and product success ($\beta = 0.07$, $p < 0.05$), respectively, had significant positive associations for the agile data set, supporting both *H7(A)*(i) and *H7(A)*(ii), respectively.

H8: PTC and project success

Overall, the contribution of PTC is not statistically greater for agile than for traditional plan-based methodologies, failing to support *H8(TA)*.There are significant positive relationships between PTC and process success ($\beta = 0.08$, $p < 0.05$) and product success ($\beta = 0.07$, $p < 0.05$), respectively, for traditional plan-driven projects, providing support for both *H8(T)*(i) and *H8(T)*(ii), respectively. Similarly, there are significant positive relationships between PTC and process success ($\beta = 0.09$, $p < 0.05$) and product success ($\beta = 0.08$, $p < 0.05$), respectively, for agile projects, supporting both *H8(A)*(i) and *H8(A)*(ii), respectively.

H9: development team expertise and project success

The contribution of development team expertise is not statistically greater for agile than for traditional plan-based methodologies, failing to support *H9(TA)*. There is a significant positive relationships between development team expertise and process success ($\beta = 0.07$, $p < 0.05$) but insignificant association with product success ($\beta = 0.02$, $p > 0.05$), respectively, for traditional plan-driven projects, supporting *H9(T)*(i) but not *H9(T)*(ii), respectively. There are significant positive relationships between team expertise and process success ($\beta = 0.09$, $p < 0.05$) and project success ($\beta = 0.06$, $p < 0.05$), respectively, for agile projects, providing support for both *H9(A)*(i) and *H9(A)*(ii), respectively.

H10: PTC and project success

The importance of PTC is statistically greater for agile than for plan-based methodologies on both process and product success, supporting *H10(TA)*. The traditional plan-driven projects data set suggests that there are significant positive relationships between team composition and process success ($\beta = 0.07$, $p < 0.05$) and product success ($\beta = 0.08$, $p < 0.05$), respectively, supporting both

H10(T)(i) and *H10(T)*(ii), respectively. The findings support that project teams composed of best various people with both business and technical knowledge in traditional plan-based projects is essential for both process and product project success. These findings are in line with a previous study by Jung *et al.* (2008). Likewise, Nah and Delgado (2006) found that outsourced software development projects require balanced cross-functional teams. The results in this study are also consistent with Lacity and Willcocks (2000, p. 362) who found that strategic project stakeholders should be included on project teams for IT outsourcing. In case of multi-sourcing, Lacity and Willcocks (2000) found that even the number and required knowledge of these stakeholders increases.

The agile projects data set suggests that there are significant positive relationships between team composition and process success ($\beta = 0.25$, $p < 0.001$) and product success ($\beta = 0.26$, $p < 0.001$), respectively, supporting both *H10(A)*(i) and *H10(A)*(ii), respectively. Thus, providing support for *H10 (TA)*. The results suggest the agile project team should be well balanced, cross-functional, and have representatives from the internal organization as well as vendors. The top managers, IT and other departments and consultants should be included on the team as well. The best people in the organization should make part of the implementation team in order to foster innovation and creativity that are important for agile project success and the functional team members involved in the project should preferably be on a full time basis. These findings are consistent with previous agile success studies. For example, Chow and Cao (2008) found that well-composed teams contributed to successful agile software development projects.

H11: UP and project success

The importance of UP is statistically greater for agile than for traditional plan-based on process but not product success, thus, supporting *H11(TA)*. The traditional plan-driven projects data set indicates that there are no significant positive relationships between UP and process success ($\beta = 0.02$, $p > 0.05$) and product success ($\beta = 0.05$, $p > 0.05$), respectively, supporting both *H11(T)*(i) and *H11(T)*(ii), respectively. These findings suggest traditional plan-driven projects do not necessarily require substantial UP during software development. There is no or little change expected during the traditional project execution and therefore, the plan and specification requirements are not expected to change once made. There is no much need for continuous involvement of the user during project implementation.

In contrast, however, the agile projects data set suggests that there are significant positive relationships between UP and process success ($\beta = 0.28$, $p < 0.001$) and a negative association with product success ($\beta = -0.14$, $p < 0.001$), respectively, supporting *H11(A)*(i) but not *H11(A)*(ii), respectively. This is consistent with empirical studies by Chow and Cao (2008), Misra *et al.* (2009) and Sheffield and Lemétayer (2013) that have provided data to support significant and positive relationship between UP and agile software development project success. Yetton *et al.* (2000) also found that UP decreases budget variance by managing expectations and quickly resolving potential problems. Similarly, Jun *et al.* (2011) found that resolving potential conflicts early arising from greater UP plays a vital role in the perceived system satisfaction of software developers and users. Therefore, UP is an effective way to know and fulfil the needs of the agile users. It may also create user commitment to the agile projects. The findings also imply that a process of continuous and dynamic evaluation and debate between informed users and developers provides the best chance for information system optimization. It can reduce friction and help to align the different viewpoints of the parties involved. Furthermore, UP through intensive communication can improve the knowledge of management on IT projects-issues and the knowledge of IT-people on the specific management issues as agile project development and implementation progresses.

However, the negative effect between UP and product success indicates that UP negatively affect product success. The results possibly suggest that although UP is necessary for process success, and participation in the requirements analysis stage can decrease the risk of there being insufficient requirements, too much UP may have a negative effect on product success. Due to the novelty of the technology involved in outsourced software development projects and the corresponding uncertainty about requirements, clients/users can continually change their requirements, which can lead to confusion and conflict and the poor product being delivered. This concurs with Nidumolu (1995) that increased interaction between users and IS staff does not necessarily lead to a software project that converges well (i.e. improved product performance). Therefore, managers need to be aware of the potential trade-offs between too much, and extremely limited UP in requirements analysis.

H12: US and project success

The importance of US on project success is statistically greater for agile than plan-based methodologies on product success but not process success, supporting *H12(TA)*. The traditional plan-driven projects model indicates that there are no significant positive relationships between US and process success ($\beta = 0.03$, $p > 0.05$) and product success ($\beta = 0.02$, $p > 0.05$), respectively, supporting both *H12(T)*(i) and *H12(T)*(ii), respectively. These findings also support that traditional plan-driven projects do not necessarily require much US during software development. There is no need for continuous support of the user during project implementation. Changes are not expected and all is planned for every task in advance. The results possibly suggest that for traditional projects, US is enforced through controlled culture using prescribed rules and regulations rather than seeking the discretionary support from users. Thus, there is compulsory compliance expected from all users.

In contrast, however, the agile projects data set suggests there is a significant positive relationship between US and process success ($\beta = 0.06$, $p > 0.05$) and product success ($\beta = 0.18$, $p < 0.001$), respectively, providing support for both *H12(A)*(i) and *H12(A)*(ii), respectively. These findings support that user attitudes need to be supportive and positive for agile projects to achieve success. Users and developers need to pay attention to each other using cautious interpersonal approaches to ensure project success. These findings are consistent with other previous scholars; Yetton *et al.*'s (2003) study found that US increases project success. Chow and Cao (2008) also found that a strong US contributed to the success of agile software development projects. Likewise, Misra *et al.* (2009) found a statistically significant and positive relationship between US and software development project success.

H13: UE and project success

The contribution of UE is not statistically greater for agile than for traditional plan-based methodologies, failing to support *H13(TA)*. There are no significant relationship between UE and process success ($\beta = 0.03$, $p > 0.05$) and product success ($\beta = 0.02$, $p > 0.05$), respectively for traditional plan-driven projects, providing support for both *H13(T)*(i) and *H13(T)*(ii), respectively. For agile projects, there are significant positive relationships between UE and process success ($\beta = 0.07$, $p < 0.05$) and product success ($\beta = 0.06$, $p < 0.05$), respectively, supporting both *H13(A)*(i) and *H13(A)*(ii), respectively.

H14: TU and project success

TU has a less negative effect for agile than for plan-based methodologies for both process and product success, supporting *H14(TA)*. TU had a negative significant direct association with process success ($\beta = -0.24$, $p < 0.001$) and product success ($\beta = -0.26$, $p < 0.001$), respectively, for traditional plan-driven projects, supporting both *H14(T)*(i) and *H14(T)*(ii), respectively. This suggests that the use of unfamiliar technologies can lead to software problems that reduce process and product success of the software development project. Similarly, for the agile projects model, TU had a direct negative and significant association with process success ($\beta = -0.09$, $p < 0.05$) and product success ($\beta = -0.06$, $p < 0.05$), respectively, supporting *H14(A)*(i) and *H14(A)*(ii), respectively. This is consistent with Jun *et al.* (2011) who found that the use of unfamiliar technologies leads to software problems that reduce the performance of the software product.

H15: TC and project success

TC has a greater negative effect on agile than on plan-based methodologies on process but not product success, partially supporting *H15(TA)*. TC had an insignificant direct association with process success ($\beta = -0.02$, $p > 0.05$) and a negative significant association with product success ($\beta = -0.14$, $p < 0.01$), respectively, for traditional plan-driven projects, not supporting *H15(T)*(i) but providing support for *H15(T)*(i), respectively. This is in line with Jun *et al.* (2011) who found that TC adversely and negatively affected software project performance in terms of both process and product performance.

For the agile projects model, TC had a direct negative and significant association with process success ($\beta = -0.17, p < 0.01$) and an insignificant association with product success ($\beta = 0.05, p > 0.05$), respectively, supporting *H15(A)*(i) but rejecting *H15(T)*(ii), respectively. This clearly indicates that, whereas TC has no direct influence on product success, it is negatively associated with process success. The findings in this study suggest that technically complex agile projects involve extensive tasks and activities related to requirements' specifications, physical design and code implementation. These responsibilities increase the difficulty of coordination efforts, which in turn cause the implementation process to deteriorate leading to a project being completed behind the schedule and over budget.

H16: RPS and project success

The negative influence of project size is statistically greater for agile than plan-based methodologies on process but not product success, supporting *H16(TA)*. The traditional plan-driven projects data set suggests that there is a significant negative relationship between RPS and process success ($\beta = -0.07$, $p < 0.05$) but insignificant association with product success ($\beta = -0.05$, $p > 0.05$), respectively, providing support for *H16(T)*(i) but not *H16(T)*(ii), respectively. These findings resonate with Yetton *et al.*'s (2003) study that larger projects are likely to be characterized by high complexity and high levels of task interdependence which increase project failure. The findings are also in line with other empirical evidence that large project size increases project risk (e.g. Sauer and Cuthbertson, 2003; Jun *et al.*, 2011; Turner and Zoli, 2012). However, the β coefficient between RPS and product success was statistically insignificant, rejecting *H16(T)*(ii).

Equally, the agile projects data set suggests that there is a strong and significant negative relationships between RPS and process success ($\beta = -0.25$, $p < 0.001$) but an insignificant association with product success ($\beta = 0.04$, $p > 0.05$), respectively, supporting *H16(A)*(i) but not *H16(A)*(ii), respectively. According to these findings, as the size of projects increases, the chances of project success tend to diminish with agile methodologies. The results suggest that agile methodologies work best on small projects as measured in terms of the number people (fewer than ten) and the budget. These results are in line with Sauer and Cuthbertson's (2003) empirical evidence that project size in terms of effort and duration can negatively affect project performance (p. 81). This seems to suggest that the increase in project size of agile projects complicates coordination and requires P&C which is counterproductive to agile practices thereby negatively affecting agile process success. In large projects, the size of communication paths becomes too many making coordination impossible for agile methodologies. The process of sharing information gets constrained as the number of people on a project increases.

H17: SC and project success

The negative effect of SC is not statistically greater for traditional plan-based than for agile methodologies for both product and process success, partially supporting *H17(TA)*. The traditional plan-driven projects model suggests there is a negative significant association between SC and process success ($\beta = -0.14$, $p < 0.05$) and a significant positive association with product success ($\beta = 0.22$, $p < 0.001$), respectively, supporting *H17(T)*(i) but not *H17(T)*(ii), respectively. These findings are in line with Ratbe *et al.* (2000) who found that changes in specifications negatively affect project success. Shenhar (2001) also found that SC had a negative effect on project performance. This is possibly because for traditional projects any changes in requirement specifications are likely be followed by increases in the budget and schedules. In contrast, the results indicate that SC improve product success of traditional plan-based projects. The β coefficient between SC and product success was significant but positive, not supporting *H17(T)*(ii). The results suggest that specification analysis of user requirement is necessary for project success and can decrease the risk of there being insufficient requirements. However, too much specification may have a negative effect on product success. When clients/users continually change their requirements, it can lead to conflicts and conflicts and a poor product being delivered late and over budget. Therefore, managers need to be aware of the potential trade-offs between too much, and extremely limited SC.

For the agile projects model, SC had a negative significant association with process success ($\beta = -0.08$, $p < 0.05$), but a positive significant association with product success ($\beta = 0.06$, $p > 0.05$), respectively, providing support for *H17(A)*(ii) but not *H17(A)*(ii), respectively. This is consistent with Boehm and Turner (2003) that SC negatively affect project performance. Equally, Nidumolu (1995) found that requirements instability had a negative effect on project performance. Too much user SC may have a negative effect on project success and in particular, variations in delivery time, scope and budget (Nidumolu, 1995).

Corresponding author
Arthur Ahimbisibwe can be contacted at: aahimbisibwe@mubs.ac.ug