Journal of International Technology and Information Management

Volume 25 | Issue 1 Article 3

2016

Choice of Agile Methodologies in Software Development: A Vendor Perspective

Sriram Rajagopalan
Indian Institute of Technology Madras

Saji K. Mathew
Indian Institute of Technology Madras

Choice of Agile Methodologies in Software Development: A Vendor Perspective

Sriram Rajagopalan
Saji K Mathew
Department of Management Studies,
Indian Institute of Technology Madras
INDIA

ABSTRACT

The purpose of this research was to develop understanding about how vendor firms make choice about agile methodologies in software projects and their fit. Two analytical frameworks were developed from extant literature and the findings were compared with real world decisions. Framework 1 showed that the choice of XP for one project was not supported by the guidelines given by the framework. The choices of SCRUM for other two projects, were partially supported. Analysis using the framework 2 showed that except one XP project, all others had sufficient project management support, limited scope for adaptability and had prominence for rules.

Keywords: Agile, software development, vendor perspective

INTRODUCTION

Software development methodology has been understood as a model used to plan, design, test and control the processes for developing an information system, furnished with one or more techniques. In this context, model refers to a logical description of development processes, which can be sequential or iterative (Matković & Tumbas, 2010). A development methodology has a direct relationship with managing project complexity. Use of methodology has also implications for usability, maintainability, adaptability, reliability and portability of the software being developed. Further, adopting an incorrect methodology could result in slippages, lack of communication and administrative overheads, leading to customer dissatisfaction. A recent study has reported a significant relationship between supplier (vendor) satisfaction and the choice of methodology (Wright, 2013).

In the past two decades, fast paced evolution of software development methodologies has effected significant improvements in software quality (Huo et al., 2004). During late 1990's, agile methodologies became prevalent to address some shortcomings of traditional methodologies like heavy documentation, lack of productivity, reliability and simplicity (Cho, 2009).Agile alliance, in response to more process driven traditional methodologies, stresses upon people, communication and customer priorities (Beck et al., 2001). Also, different agile methodologies has exhibited flexibility to working within constraints, without demanding major upfront investments, while being adaptable to changing market conditions(Mohammad et al., 2013).

Inspite of their widespread use in the last 15 years, agile methodologies have also shown many limitations. Some notable limitations include dependence on run-time tacit knowledge rather than

  ISSN: 1543-5962-Printed Copy ISSN: 1941-6679-On-line Copy

more documented information, lack of traceable and proved implementation guidelines for mission critical projects, lack of adequate support for repetitive and large scale projects and team requirement of highly talented, self-motivated individuals with a high degree of implementation freedom (Cho, 2009).The outcomes of projects developed using agile methodologies are dependent mainly on organisational factors like customer commitment, decision time, team location and composition, corporate culture and people factors like competency and self-motivation (Vinekar et al., 2006). In order to address various concerns, several methodologies within the agile category evolved; Abrahamsson et al. (2010) identifies ten of them as truly agile.

While there is considerable interest in the IT industry to adopt newer methods fostering agility, guidelines in choosing the right methodologies, based on relevant factors, remain scanty (Wright, 2013; Coram & Bohner, 2005).What are the key determinants in the choice of agile methodologies in order to meet project goals? Are these factors identified in extant literature followed in practice? How are decisions on the choice of agile methodologies taken in practice? Prior research has provided some frameworks for analysing characteristics of agile methods. For example, Qumer and Henderson-Sellers (2008) identifies four dimensions for analysing agility of various agile methods. Drawing on these studies, our research seeks to develop a normative understanding about the choice of agile methodologies and compare the findings with the actual choices and their rationale. Following a case study method, we analyse five software development projects executed by three vendor organisations. The insights from this study would serve to inform practising managers on the choice of agile methodology and would also contribute to the body of knowledge in agile methods, where empirical studies have been found to be limited (Dyba & Dingsoyr, 2008).

The remaining part of this paper is organised as follows: Section 2 gives a review of literature on agile methods and their choices, Section 3 describes the methodology followed in our study, Section 4 gives the detailed analyses of our case data and Section 5 ends with conclusions.

LITERATURE REVIEW

Foundations

The fundamentals of agile methodology- incremental and iterative development, go six decades back, when researchers worked on the principle of separating design, implementation and testing (Larman & Basili, 2003). The implementation phase was characterised by generations of systems of codes and functional sub specifications, so that there were intermediate check points for validation and verification against the final expected product. Two decades later, evolutionary project management evolved as one of the key incremental and iterative development practice. Scholars thus approached complex systems development by reductionism, breaking down the system into small units, each one having a small well defined goal or prototype, totalling to larger goal and every prototype having sufficient provision for retreat (Nerur & Balijepally, 2007; Dingsoyr et al., 2007; Wright, 2013).

During the development and evolution of agile methodology and the community surrounding it, the movement has benefitted from conceptual foundations in other disciplines such as architecture, Socio technical systems, soft systems methodology, support congruence and transitional organization (Nerur et al., 2010).The key characteristics of agile development methodologies-greater autonomy, decision-making discretion and adaptive understanding, have a theoretical

  ISSN: 1543-5962-Printed Copy ISSN: 1941-6679-On-line Copy

background, which is consistent with problem-framing, problem-solving approaches in architecture and strategic management (Nerur & Balijepally, 2007).

Comparison of agile methodologies

Though there are many agile methodologies which have evolved by tailoring various principles and processes, there are only few which are commonly used. Despite the differences, all agile methods are focused mainly on business problems and their solutions in the shortest time-frame, with very frequent releases to business user community amidst their dynamically changing priorities. Here, the process is a low key affair and communication is high key, relying on smaller, very closely knitted highly motivated teams (Strode, 2006).

The differences among the agile methods are in their purpose they solve, based on the demand or customer needs. Some of them focuses more on practises and others on management aspects. Also, there are significant differences among the agile methods in the extent of coverage for phases of the software development life cycle. They also differ in the team composition they recommend, to bring efficiency in the respective methodology-usage for the given purpose. Abrahamsson et al. (2010) compare and contrast 10 agile methodologies using six dimensions. Table 1 gives a summary of this analysis with phases of development life cycle they support and constraints of its usage.

Qumer and Henderson-Sellers (2008) developed a similar framework for comparing agile methodologies based on their agility characteristics. Using a four dimensional framework of scope, features, agile values and processes, they analysed six agile methods. The authors finally arrive at a degree of agility index that could guide choice of an agile method for a given project. Geambasu et al. (2011) also developed an extant view of agile methods by using factors derived from literature.

ADOPTING AGILE METHODOLOGIES

Role of the customer

The customer plays a very important role in agile projects with key responsibilities to drive the project, interact constantly with business users and provide requirements and participate in retrospection to test the intermediate deliverable and its compliance (Martin et al., 2009). However, customer may fail to keep these practises on sustainable pace due to the dynamic nature of development projects. Hence, the customer's role is essentially not played by a single person but there are pseudo roles assumed by different people to drive the project effectively, known by role labels such as the technical liaison, negotiator, customer coach, skill specialists for designer, tester and quality facilitator.

Agile methodology	**Characteristics**	**Phases of development life cycle supported**	**Usage constraints**

ISSN: 1543-5962-Printed Copy ISSN: 1941-6679-On-line Copy

ASD	Promotes adaptive paradigm, derives principles from radical software development	Requirements, design, code, unit test, integration test, system test, acceptance test	It is more about concepts and culture rather than in-practice
AM	Agile modelling – agility and rapid development in producing sufficiently advanced models to support acute design.	None	It does not work independently but work within other methods as supplement
Crystal	Family of methods to be chosen suitable for the business needs with rules of thumb for tailoring. Flexible and configurable process	Design, coding, unit test, integration test, system test	Lack of support for mission-critical systems, distributed teams, scalability, insistence on only collocation.
DSDM	Provides control framework for rapid application development. Keeps time and resources as constant and adjust the functionality to be developed	Project inception, requirements, design, code, unit test, integration test, system test, acceptance test, system in use	Availability issue to wider audience
XP	Customer focused and close customer participation, short iterations and short release, continuous re-factoring.	Requirements, design, code, unit test, integration test, system test	Lack of attention on management practises
FDD	More emphasis on quality, frequent and tangible deliveries and accurate monitoring of project progress. Very short iterations	Requirements, design, code, unit test, integration test, system test	Focused mainly on design and implementation
ASP	The Agile Software Process model focuses on accelerated development with flexibility to include volatile requirements	Requirements, design, code, unit test, integration test, system test, acceptance test	Unable to predict effort upfront and threat of loss of focus due to changing requirements.
PP	Pragmatic programming – more of pragmatic perspective with a set of best practises	None	Does not define any concrete methodology to develop a system

ISSN: 1543-5962-Printed Copy ISSN: 1941-6679-On-line Copy

SCRUM	Focuses on flexibility, adaptability, productivity, through small, self-motivated teams. Integrated project management process to overcome deficiencies in the development process	Requirements specification, integration test	Coding and all testing process not defined completely.

Table 1. Comparison of Agile Methodologies.

Martin et al. (2009) summarise customer effectiveness in agile projects based on three practices:(i) real customer involvement – extent of direct involvement of all stakeholders, (ii) whole team – skills and perspectives required for the team to create a sense of team-ness and (iii) energised work – working effectively and productively. In line with above, some empirical studies have suggested new customer-focused practices which include customer's apprentice, programmer on-site, use of contextual enquiry, programmer holiday, road show, customer pairing, customer boot camp, big picture up-front and re-calibration (Beyer et al., 2004; Takats & Brewer, 2005).

Organisational culture and deployment of agile methods

Siakas and Siakas (2007) studied the close relationship between organisational culture and agile method usage and showed how agile approach forms distinct cultures of its own. They also propose four different types of organisational culture as clan, democratic, hierarchical and disciplined. Agile methodologies are more suited for organisations following democratic culture. Iivari and Huisman (2007) proposed a competing value model, a two dimensional model, focussing on values as core constituents of organisation culture. The model help analyse culture based on two contrasting aspects– change (flexibility and spontaneity) vs. stability (control, continuity and order) and internal focus (integration and maintenance of Socio technical system) vs. external focus (competition and interaction with organisation environment). Four types of culture evolved – group culture concerned with human relationship and flexibility, developmental culture, which is more future oriented, rational culture, more achievement oriented and hierarchical culture, focused on order, routine and regulations.

Iivari and Iivari (2010) used the competing value model in the context of agile methods and posited that agile methods and hierarchical culture are incompatible. Further agile method implementation in hierarchical organisation is still possible through combination of complementary features of different types of methodologies. However, this might result in heavier implementation leading to loss of agility.

DECISION FACTORS IN CHOOSING AGILE METHODOLOGIES

Some studies have focused on agile methodologies and their impact on project management, especially project success. Chow et al. (2008) identified factors that can serve to guide in the selection of agile methodologies for projects in organisations with specific characteristics. Further, the authors summarised 6 key dimensions of agile methodologies with specific attributes to guide selection of a specific methodology. These dimension covers delivery strategies (short delivery cycles), software engineering techniques (simple design, rigorous re-factoring techniques, strong testing strategy), team capability (highly competent teams, adaptive management style, strong

ISSN: 1543-5962-Printed Copy ISSN: 1941-6679-On-line Copy

training mechanism), project management process (strong requirement management, configuration management, communication management processes), team environment (team location, self-organised teams) and customer involvement (strong relationship with customers, customer having full authority).

Wan et al. (2010) highlight aspects of organisational agility namely innovation, rapid response to change, initiative and learning that are key determinants while selecting agile development methodologies. Similarly, Lindvall et al. (2002) suggested project size, highly competent personnel, criticality, reliability and safety issues as key aspects in selecting projects suitable for agile methodologies.

In summary, our survey of literature shows that agile methodologies have been studied in relation with project management, organisational culture and specific project characteristics. Although several studies have been conducted in these three streams of research, empirical evidence to guide choice of methodologies are still scanty, as also reported by a research review in this area (Dyba & Dingsoyr, 2008).Some scholars have suggested frameworks that are useful in comparing agile methodologies based on their characteristics; these frameworks complement and inform methodological studies in agile development (Geambasu et al., 2011; Abrahamsson et al., 2010; Qumer & Henderson-Sellers, 2008). However, studies that map project characteristics to agile methodology characteristics to provide an analytical framework for decisions on methodologies still need more attention. Further, empirical evidence to compare pragmatic choices with normative choices have not been found in prior studies to the best of our knowledge. Our study addresses this gap by developing an analytical framework from extant literature and using it to develop a normative view of agile methodology choice and then further compare it with actual decisions.

RESEARCH METHODOLOGY

The purpose of our study is to develop understanding about decisions made regarding the choice of specific agile methodology. We chose multi case study method as most appropriate for our research as theoretical studies have been very limited in this area (Edmondson and McManus 2007). We follow a deductive case study approach, as we could identify useful frameworks that could be synthesised for analysing qualitative data (Yin, 2013).

Data sources and sampling

Following the principles of deductive multi-site case study approach (Yin, 2013), we chose three organisations to collect data for our study. A brief profile of the organisations with the projects pertaining to each organisation is given in Table 2 below. Our sample consisted of two large firms with over 100,000 employees and one small firm with about 100 employees. One large organisation we studied was headquartered in the United States and the other two were based in India. All the three firms had business focus on IT and IT consulting. This profile of the firms provided sufficient diversity as well as similarity at the organisational level in line with the recommendations for multiple case studies (Yin, 2013).

Since our unit of analysis was software development projects, we applied criteria following the guidelines of replication logic in the choice of projects. We chose to study five projects from three organisations including large projects (largest: 1600 KLOC / 120960 person hrs) and relatively small projects (smallest: 300 KLOC / 17136 person hrs). We also ensured that the agile methods

chosen for the projects were also sufficiently diverse: two projects involved SCRUM with pair programming, another two with SCRUM and one with XP. Respondents were chosen based on fulfilling three criteria: (a) minimum 10 years of experience in software development projects (b) experience in working with a minimum of five agile projects (c) involvement in the methodological decisions of project management.

Sl No	Company name*	No of employees	Turn over (₹)	Business focus	No of projects	Projects
1	ABC Limited	100,000 +	40 million	IT, BPO, Consulting	3	p1, p3, p5
2	DEF Technology Services	100,000 +	600 billion	IT, Consulting	1	p2
3	GHI Technologies	100		IT, Consulting	1	p4

*real names camouflaged

Table 2: Description of case sites.

We predominantly used interview method for data collection. A structured questionnaire was used for conducting the interview, the questionnaire was developed based on the framework developed from extant literature discussed in section 3.2. The questionnaire was emailed to our respondents a few days in advance of the interview dates. All the interviews were conducted face to face and notes were taken in a pre-prepared spreadsheet.

Data analysis framework

We used two frameworks for analysing our data. The frameworks were developed using extant literature dealing with agile methods in software development. The first framework was formed by combining relevant variables from the study of Qumer and Henderson-Sellers (2008) and Geambasu et al. (2011). According to Qumer and Henderson-Sellers (2008, p. 282), variables under the four dimensions of project scope, features, agile values and development process would be useful in determining the degree of agility of agile methods. We also extracted relevant variables from the work of Geambasu et al. (2011), who have identified influence factors for the choice of a software development methodology. This analysis framework 1 thus consists of 12 variables, which together is useful in explaining the choice of the agile method used in each project.

In order to develop further understanding about the agile methods adopted in the five projects, we used relevant dimensions of the analytical framework developed by Abrahamsson et al. (2010, p. 37). According to this framework, agile methods could be analysed based on the six perspectives of project management support, software development life cycle, availability of concrete guidance for application, adaptability in actual use, research objective and empirical evidence. For the purpose of our study, we will use the first four perspectives, which also enable us to compare projects as against methodologies which were the objective of the study.

According to Abrahamsson et al. (2010), DSDM and ISD supports the full life cycle and provide complete project management support. Scrum and XP are comparable in life cycle support from requirements specification to system testing, However XP does provide concrete guidance and processes, though it does not provide project management support whereas scrum does. Similarly, pair programming offers no project management support or processes, but it does provide concrete guidance. Scrum with pair programming has thus the potential to complement each other.

RESULTS AND DISCUSSION

Table 3 uses the framework developed by Qumer and Henderson-Sellers (2008, p. 282) to analyse the projects. Based on this analytical framework, we notice that, project 4 was distinctly different from other four projects. The high requirement volatility and agile familiarity may explain the choice of scrum as the predominant choice for project 1, 2, 3 and 5. However, project 4 was least complex as compared to all other projects and was very critical but similar development was done earlier and requirements were fairly stable.

The cost and time estimation accuracy was high for project 4 compared to other projects and so was the client experience in agile development due to their past experience in executing similar projects. However, team familiarity with agile development was found to be medium to low among all the 5 projects. The team size of project 4 was 45, working across 3 locations. This project defies the logic of the choice of agile methods, which is not meant for repetitive projects. Further, the recommended team size for XP and SCRUM is <10 (Qumer and Henderson-Sellers, 2008), whereas project 4 had 45 team members. The condition for team size to be less than 10 was not satisfied strictly in any of the projects.

Determinants	**Project 1 (p1)**	**Project 2 (p2)**	**Project 3 (p3)**	**Project 4 (p4)**	**Project 5 (p5)**
Project Size	300 KLOC / 50400 Person hrs (medium)	48000 person hrs (large)	1600 KLOC / 120960 person hrs (large)	2000 man days (medium)	300 KLOC / 17136 person hrs (small)
Development style	Iterative, distributed agile	3 amigo methodology, acceptance testing driven development, distributed agile	Poker methodology, test driven development, distributed agile	Product development in fixed price model, test driven development in distributed agile	Iterative, distributed agile
Requirements uncertainty	Undisciplined, so caused challenges in accommodating changes even during sprints (high	Improperly thought-out requirements impacted development cycles (medium)	Observed impact on quality of the output (medium)	Product requirement document was available which owned by customer (low)	Assumption was agile project can accommodate any number of requirement changes (high)

Cost and time estimation accuracy	Experienced schedule delay due to requirement complexities and very high requirements volatility (low)	Sufficient buffer time was not planned for retrospection. Requirements were met on schedule and cost benefit was achieved due to distributed agile. Investments were factored for training the team (medium)	The project was estimated to complete in 12 months and the high attrition was not factored which impacted the schedule (medium)	Past experience in developing similar product helped in effort estimation accuracy (high)	There were flaws in estimation as some of the roles were not factored full-time for onsite particularly and sufficient time was not factored for sprint meetings and planning meetings (medium)
Software criticality	Critical for business development (high)	The product was planned for internal use mostly (medium)	Customer was involved in every stage of projects and every iteration (high)	Product development highly critical for the banking business (high)	Time to market was highly desirable (medium)
Complexity	24 interfaces (high)	4 major interfaces (medium)	IP based latest technology, 8 interfaces, multiple location of the scrum teams (high)	3 external interfaces and overall 7 interfaces (low)	6 interfaces (medium)
Technology environment	RTC (Rational team Concert), Maven, RAD were used. Maven builds the code and errors thrown are reworked and rebuild happens till the build is successful	Quality centre - defect tracking tool, RAD – Rapid Application Development tools used for building graphical user interfaces	RTC - tracking tools - velocity will give the health of the project and whether the capacity is optimally used	Bugzilla - defect tracking tool to identify and fix key defects during iteration and releases, VSS – version control of source code	Rally - Used for reviews of the progress of the project and decisions are made if required for re-planning etc.
Business culture	Collaborative and cooperative	Customer ecosystem was conducive to implement agile though there was lack of infrastructure support for some time period.	Cooperative but faced communication challenge and cultural differences	Collaborative and cooperative	Collaborative and cooperative
Physical environment	Distributed agile - Onsite - IPM, Business Leads, Offshore - rest of the	Distributed agile - Onsite - One scrum master, some scrum developers,	Distributed agile - Onsite - scrum team, One scrum master, One product owner,	Distributed agile - Sydney (onsite)- One technical lead, One Agile PM,	Distributed agile - Onsite – project manager, business analysts, technical lead, tester,

	technical leads, business analysts	Offshore - rest of the scrum masters	Offshore - 4 scrum teams, 4 product owners, scrum of scrum masters	4 developers Singapore (offshore) – product owner, one technical lead, 8 developers, India (OFF) - rest of PM, leads and developers	Offshore – project manager, rest of the technical leads, developers, testers
Client familiarity with agile	Client has sufficient working experience in using agile (medium)	Customer has medium level of familiarity using agile (medium)	Customer had less experience in executing agile projects (low)	Customer had extensive experience using agile for 5 years (high)	Customer had extensive experience using agile for 3 years (medium)
Team size	25 people working across 2 locations (medium)	30 people working across 2 locations (medium)	60 people working across 5 locations (high)	45 people working across 3 locations (high)	17 people working across 4 locations (low)
Team familiarity with agile	ZERO agile experience and all were only trained before the start of the projects (low)	No prior agile experience for the team (low)	Only scrum masters and product owners had agile experience and team had no experience (low)	Team had an average of 1.5 years of prior experience working in agile projects (medium)	Team members had average of 1 year experience working in agile projects (medium)

Project 1, 3: Scrum and pair programming; Project 2, 5: Scrum; Project 4: XP

Table 3: Determinants of agile methodology for development.

Perspective	Project 1 (p1)	Project 2 (p2)	Project 3 (p3)	Project 4 (p4)	Project 5 (p5)
Project management support	Iteration manager, sprint planning daily scrum meeting reflect the project management process followed	Scrum Master, iteration planning and daily scrum meeting reflect the project management process followed	Scrum Master, Scrum of Scrum Masters, pre-sprint planning meeting, scrum of scrum and techno meetings reflect the project management process followed	Does not offer a clear project management view, though the planning game and agile PM reflect the project management process followed in parts	Sprint planning meeting and daily scrum meeting reflects the project management process followed extensively

Software development life cycle	Covers life cycle process for requirements, design and system testing phases. Implementation phase does not follow any particular life cycle process and focusses only on releasing usable product at the end of phased releases	Iteration 1 phase was utilised for finalising architecture and design. Development was based on test cases developed in previous iterations. After certain cycles in one intermediate phase, release test was done. User acceptance testing, code freeze and migration to production for release done	Requirement finalisation process not followed, initial iteration phase was used for dry run, poker methodology followed for development sprints. Unit testing and weekly integration testing happens in same sprint and functional testing for previous sprint in subsequent sprint	Though there is no explicit inception phase, at every phase, there is customer validation, acceptance testing and release of usable product. The project followed all process of requirement, design, implementation and system testing phases	There are 2 design and planning sprints. On the last day of each sprint, build and demo done and moved to staging environment on acceptance. At the end of every 4th sprint, release planning done and moved into production and a workable product released
Availability of concrete guidance for application	Implementation phase of design, coding and unit test rely on abstract principles and the specific methodology offers guidance for the requirements and integration testing phase	Specific organisation / project defines and develops their practises for the design, coding and unit testing phase but no concrete guidance on how this should be done and it is subjectively driven	Employed the best practises back into the actual practice of software development for the requirements gathering and integration testing phase. The rest of the life cycle phase is free flow driven with no specific settings and completely tailored based on the context	The agile methodology used in the project have been directly derived from practical settings and relies on concrete guidance	Key emphasis is on abstract principles over concrete guidance for the implementation phase to enable flexibility in the development process
Adaptability in actual use	Allows adaptability in actual use, but no guidance provided for adaptation rules	The business requirements were included during run time, which were adopted at the right time into sprints. But high adaptability wears the team soon in the	Though there is a flexibility to allow situation specific modifications, the degree of volatility is indirectly proportional to the quality of the output, although SCRUM works	XP principle of "Just rules" were followed and principles and procedures were changed, but with a certain degree of uniformity with acceptance from the development groups involved	Followed situation applicability based adaptation. Does not follow any pre-prescribed practises

		cycle due to unexpected changes	on principle of high adaptability		

Project 1, 3: Scrum and pair programming; Project 2, 5: Scrum; Project 4: XP

Table 4: Analysis of agile methods adopted.

In reference to Table 4, our analysis using the framework of Abrahamsson et al. (2010) shows that all projects with the exception of p4 received adequate project management support from the agile methodology adopted. None of the projects were supported for all the life cycle processes completely; however, development processes were managed by incorporating them within the available phases. All projects except p5 had subjectivity and practical considerations guiding the processes and p5 drew from abstract principles for guidance. Projects p2 and p3 had adaptability, which had somewhat a negative influence on team performance. Project p4 provided less space for adaptability and rules were given more prominence.

Table 5 provides a comprehensive view of the choice of methodology in each project. The fit of the specific methodology to the given project has been either low (p4) or medium (p1, p2, p3, p5). This inference has been drawn based on the qualitative scoring and analyses given in Tables 3 and 4. We found that XP was not an appropriate choice for p4 according to the frameworks given by Qumer and Henderson-Sellers (2008), Geambasu et al. (2011) and Abrahamsson et al. (2010). The project was repetitive, requirements were stable, and estimation was near accurate, however XP was still chosen as the methodology. The rationale used by the customer seems to be the team's familiarity and customer's own experience in agile development. In particular, the need to see an interim output seemed to have strong influence. In projects other than p4, the fit is medium and the methodology was the choice of the customer. In p1 and p4 there was some involvement of the vendor in the choice of methodology. A representative of project 1 team commented as follows:

"Developers, testers and business analysts together frame the requirements for each iterations and develop the story cards. Need for business to work hand in hand with IT services and intermediate validations helps in shaping the final product motivated to use agile"

Projects	**Agile method**	**Fit**	**Decision maker(s)**	**Supporting quotes**
p1	SCRUM Pair Programming	medium	customer-vendor	*"Customer mandated that the team should not be more than 24 members. Team had only prior training but no prior working experience in Agile"*
p2	SCRUM	medium	customer	*"Followed scrum of scrums and each scrum team was mandated not to have more than 8 members"*

p3	SCRUM Pair Programming	medium	customer	*"Customer insisted on agile methodology and since the vendor organization had expertise, this was planned to be executed in scrum. Customer wanted uniformity of following scrum for all their projects and rejected the TDD and Pair programming and requested vendor to drive a separate competency plan"*
p4	XP	Low	customer	*"Customer had earlier projects running on scrum and XP and based on experience, the customer wanted to go with XP which suited their organization. Alternate methodology was taking 10 months but they wanted a release within 6 months. Customer was unable to see any tangible output of the product till first 6 months in waterfall model"*
p5	SCRUM	medium	customer, vendor consultants, (review)	*"Customer decision to adopt SCRUM; agile consultants from vendor reviewed and suggested not more than 6 member for each scrum. Before the start of this project, customer had experience on agile and scrum and customer had no thought of alternate methodology"*

Table 5: Decision fit and rationale used: extant vs practical view.

CONCLUSIONS

The purpose of this study was to develop understanding about the choice of agile development methodologies in software development projects. We used conceptual frameworks available in prior literature to develop an extant view of what the choice of methodology ought to be and further contrasted it with the rationale for the actual choice. This approach of comparing the normative and descriptive aspects of the phenomenon provided a useful approach for the evaluation of choices involved.

Our findings show that the fit was poor for one of the projects and the decision was made by the client predominantly based on their prior experience. Other projects considered in the study had a medium fit with the methodologies chosen however, the decision was again influenced by the client; in two cases pertaining to this category vendor had no involvement in the decision concerning the choice. This influential role of the client in the decision to choose methodology was not related to the size of the vendor firm. We found that the two projects were done by vendor firms with employee strength more than 100,000 and another by a small vendor firm with an employee strength of 100. However in all the three cases client chose the methodology. Even when

the extant view provided only medium motivation for the choice of the chosen methodology, the actual choice was guided by more subjective considerations. These subjective reasons are not fully known as we did not have access to the client firms due to confidentiality requirements of the vendors involved in the study. The absence of evidence from the client firms is a limitation of this study although, to some extent this is overcome by the vendor representatives' knowledge of the clients.

Our study provides a theoretical basis to conduct future research that could be more generalisable by following quantitative approach. The variables identified in this study for understanding choice of agile methodology and their proposed relationships with the choice could be tested using statistical techniques. Such findings will provide substantial and valuable guidelines for project management related to software development.

REFERENCES

Abrahamsson, P., Oza, N., & Siponen, M. T. (2010). Agile Software Development Methods: A Comparative Review1. In *Agile Software Development* (pp. 31-59). Springer Berlin Heidelberg.

Beck, K., Beedle, M., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D. & van Bennekum, A. (2001). *The agile manifesto*, Available from: http://agilemanifesto.org/ (12, 2014).

Beyer, H., Holtzblatt, K., & Baker, L. (2004). An agile customer-centered method: rapid contextual design. In *Extreme Programming and Agile Methods-XP/Agile Universe 2004* (pp. 50-59). Springer Berlin Heidelberg.

Cho, J. (2009). A hybrid software development method for large-scale projects: rational unified process with scrum. *Issues in Information Systems*, *10*(2). 340-348.

Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, *81*(6), 961-971.

Coram, M., & Bohner, S. (2005, April). The impact of agile methods on software project management. In *Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the* (pp. 363-370). IEEE.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, *50*(9), 833-859.

Edmondson, A. C., & McManus, S. E. (2007). Methodological fit in management field research. *Academy of management review*, *32*(4), 1246-1264.

Geambaşu, C. V., Jianu, I., Jianu, I., & Gavrilă, A. (2011). Influence factors for the choice of a software development methodology. *Accounting and Management Information Systems*, *10*(4), 479-494.

Hajjdiab, H., & Taleb, A. S. (2011). Adopting agile software development: Issues and challenges. *International Journal of Managing Value and Supply Chains (IJMVSC)*, *2*(3), 1-10.

Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004, September). Software quality and agile methods. In *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International* (pp. 520-525). IEEE.

Iivari, J., & Huisman, M. (2007). The relationship between organizational culture and the deployment of systems development methodologies. *MIS Quarterly*, *31*(1), 35-58.

Iivari, J., & Iivari, N. (2011). The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology*, *53*(5), 509-520.

Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, *36*(6), 47-56.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L., & Zelkowitz, M. (2002). Empirical findings in agile methods. In *Extreme Programming and Agile Methods—XP/Agile Universe 2002*, 197-207. Springer Berlin Heidelberg.

Martin, A., Biddle, R., & Noble, J. (2009, August). XP customer practices: A grounded theory. In *Agile Conference, 2009. AGILE'09.* (pp. 33-40). IEEE.

Matković, P., & Tumbas, P. (2010). A Comparative Overview of the Evolution of Software Development Models. *International Journal of Industrial Engineering and Management (IJIEM)*, *1*, 163-172.

Mohammad, A. H., Alwada'n, T., & Ababneh, J. M. A. (2013). Agile Software Methodologies: Strength and Weakness. *International Journal of Engineering Science and Technology (IJEST)*, *5*(03), 455-459.

Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, *50*(3), 79-83.

Nerur, S., Cannon, A., Balijepally, V., & Bond, P. (2010). Towards an understanding of the conceptual underpinnings of agile development methodologies. In *Agile Software Development* (pp. 15-29). Springer Berlin Heidelberg.

Qumer, A., & Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and software technology*, *50*(4), 280-295.

  ISSN: 1543-5962-Printed Copy ISSN: 1941-6679-On-line Copy

Siakas, K. V., & Siakas, E. (2007). The agile professional culture: A source of agile quality. *Software Process: Improvement and Practice*, *12*(6), 597-610.

Strode, D. E. (2006). Agile methods: a comparative analysis. In *Proceedings of the 19th annual conference of the national advisory committee on computing qualifications, NACCQ* (Vol. 6), 257-264.

Takats, A., & Brewer, N. (2005, July). Improving communication between customers and developers. In *Agile Conference, 2005. Proceedings* (pp. 243-252). IEEE.

Vinekar, V., Slinkman, C. W., & Nerur, S. (2006). Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information systems management*, *23*(3), 31-42.

Wan, J., & Wang, R. (2010). Empirical research on critical success factors of agile software process improvement. *Journal of Software Engineering and Applications*, *3*(12), 1131-1140.

Wright, G. P. (2013). Success rates by software development methodology in information technology project management: A quantitative analysis, *UMI Number: 3590342*, UMI Dissertation Publishing, ProQuest LLC, Michigan.

Yin, R. K. (2013). *Case study research: Design and methods*. Sage publications.

Sriram Rajagopalan

Department of Management Studies, Indian Institute of Technology Madras,
Chennai – 600 036, Tamil Nadu, INDIA

Sriram Rajagopalan is currently pursuing his PhD from Department of Management Studies, Indian Institute of Technology Madras. He has 19 years of work experience in Indian Software industry and has played roles of Project Manager, Delivery Manager and Delivery head. He has high level of expertise in Remote Infrastructure Management, Software development, Project management, Program Management and Delivery Management and his core expertise is in managing end to end critical customer infrastructure and SLA governance. His areas of academic interest are Global sourcing, Performance of Global software development and his present work is on Choice of agile Methodologies in Software projects.

Saji K. Mathew

Department of Management Studies, Indian Institute of Technology Madras,
Chennai – 600 036, Tamil Nadu, INDIA

Dr. Saji K. Mathew is currently an Associate Professor at the Department of Management Studies, Indian Institute of Technology Madras. In research, he focuses on the role of Information Technology in Business and Management. As a Fulbright Scholar, he did his post-doctoral research on risk mitigation in offshore IT outsourcing at the Goizueta Business School of Emory University, Atlanta (USA). His present research interests cover strategies in offshore IT outsourcing, issues in IT infrastructure management services, information privacy and data mining. He teaches courses such as Management Information Systems, Information Systems Development and Research in IT and Organizations.